A VARIABLE DECISION SPACE APPROACH FOR IMPLEMENTING A CLASSIFICATION SYSTEM

by

Ryszard S. Michalski

Second International Joint Conference on Pattern Recognition, pp. 71-75,
Copenhagen, Denmark, August 13-15, 1974.

74CHO885-4C

IFIP

DIF

SECOND INTERNATIONAL JOINT CONFERENCE ON PATTERN RECOGNITION

August 13-15, 1974
Copenhagen, Denmark

2nd International Joint Conference on Pattern Recognition,
August 13-15, 1974, Lyngby-Copenhagen, Denmark.

A VARIABLE DECISION SPACE APPROACH FOR IMPLEMENTING A CLASSIFICATION SYSTEM

R. S. Michalski
Department of Computer Science
University of Illinois
Urbana, Illinois 61801 U.S.A.

Summary

This paper gives a brief exposition of a variable decision space classification approach based on the application of a recently developed concept of a variable valued logic system[3,4,6] (specifically system* VL_1). This approach produces descriptions of object classes which involve only the most appropriate or sufficient descriptors (features) characterizing each object class ('variable decision space'), as opposed to conventional classification approaches which use the same set of descriptors for describing each class ('constant decision space'). Two classification methods are described: DCD (Dependent Class Descriptions) and ICD (Independent Class Descriptions) and are illustrated by an example.

1. INTRODUCTION

In conventional decision (feature) space methods, descriptions of classes of objects (class descriptions) are usually in the form of probabilistic or algebraic expressions which depend on the same variables (features) for every class. Geometrically this means that the classes correspond to certain n-dimensional domains in the n-dimensional decision space. Such methods, which we will call constant decision space methods, do not have proper mechanisms for selecting, from the original set of variables, the subsets of them which are most suitable for describing each class. Also, to decide class membership, knowledge of the values of all variables describing an object is required, no matter to which class the object belongs.

The above requirements are significant restrictions of the conventional methods. It is easy to notice that, e.g., humans, in describing and recognizing objects use class descriptions involving variables (descriptors) which usually are most suitable for describing each individual class. Consequently, each class may be described by a different set of descriptors. Thus, as we will say, they use a variable decision space method. For example, in describing the letter 'H', one might say that it 'consists of 2 parallel vertical bars linked in the middle by a horizontal short bar'. But when describing the letter 'O', one might say that it is 'an ellipse elongated vertically'. The only common descriptor in both descriptions is the concept 'vertical'.

The need for developing variable decision space methods is becoming recognized. For example, methods oriented in this direction were recently described by Watanabe[1] and Hayes-Roth[2]. The first, to this author's knowledge, implemented adaptive recognition system which used a variable decision space method was developed by Karpinski and Michalski[5].

The present paper gives a brief** exposition of a variable decision space classification approach based on the application of a recently developed concept of a variable valued logic (VL) system[3,4,6] (specifically system VL_1). In the VL_1 system, any proposition and each of the variables in a proposition range over independent domains. These domains are determined according to the meaning of the proposition and variables, and to the requirements of a given problem. The VL_1 logic system appears to be well suited for classification problems. In such applications, formulas of the VL_1 system are used as descriptions of object classes, and the variables in these formulas are used as descriptors characterizing individual objects. Object descriptors can often be interpreted as variables with different number of values (e.g., color: yellow, red, blue, etc.; sex: male, female; temperature: low, medium, high, very high), and therefore the VL_1 system can be easily applied. The system also provides a sound theoretical basis for constructing 'most efficient' or 'simplest' (according to defined criteria) class descriptions which use only the most appropriate descriptors for characterizing each class of objects. It also has a number of other attractive features from the viewpoint of classification problems, such as high computational efficiency, applicability in intrinsically non-linear or multimodal problems, simplicity for human comprehension, the comparatively small number of learning samples required, etc.[7]

In this paper we distinguish two specific classification methods based on the system VL_1 ('VL_1-based methods'):

DCD - a Dependent Class Description method

ICD - an Independent Class Description method,

and illustrate each method by an example.

2. VL_1-BASED CLASSIFICATION METHODS

2.1 Event Space as the Universe of Representations of Objects

By VL_1-based classification methods we mean methods of using the VL_1 system for classification purposes. These methods assume that objects to be classified are described by sequences of values of n variables, the domains of which are finite sets D_i, $i = 1,2,\ldots,n$. The set of all possible such sequences (serving as a universe of object representations) is called an event space and is denoted by $E(d_1, d_2, \ldots, d_n)$ or, briefly, E, where d_i, $i = 1,2,\ldots,n$, are the cardinalities of the D_i. Thus we have:

$$E(d_1,d_2,\ldots,d_n) = D_1 \times D_2 \times \ldots \times D_n \qquad (1)$$

Elements of an event space are called events. Variables (the values of which are components of events) are generally identified by names and no order among these variables is implied. To be specific, however, we will assume that these variables are $x_1, x_2, \ldots, x_n$, and their domains are $D_1, D_2, \ldots, D_n$, respectively. Variables x_i, $i = 1,2,\ldots,n$, correspond to certain descriptors (features, properties, relations, or domains of relations) which are used to characterize objects (e.g., they could be various measurements derived from objects). Values of variables could be numbers (e.g., results of measurements) or, in the most general case, names of any objects.

In the AQVAL/1 implementation[3] of the system VL_1 (to which we will refer in this paper) it is assumed

---

* For the lack of space, the definition of the VL_1 system has not been included. We will present here, however, examples of the VL_1 formulas and their interpretation. Therefore, a Reader not acquainted with papers[3,4,6] will be able to understand the contents of this paper to a satisfactory degree.

** For a more detailed description see paper[7].

that domains D_i are sets of non-negative integers:
$D_i = \{0,1,2,\ldots,\not{d}_i\}$ where $\not{d}_i = d_i - 1$. (2)

If the original domains of variables are not such sets of non-negative integers, then they should be mapped into such sets. This mapping can be accomplished in the following way.

If an original domain of a variable is a finite set, then the elements of the domain are ordered according to some desired semantic or syntactic property, and are assigned positions $0,1,2,\ldots,\not{d}_i$. These positions will then be taken as values of variables (variables with domains in the form (2) will from now on be called AQVAL/1-variables). By 'ordering according to a semantic property' we mean an ordering based on the meaning of values of variables. For example, consider a variable 'height of a person'. Its values can be measurements of a person's height expressed in centimeters, or in words, such as 'tall, short, average, very tall', depending on the degree of preciseness required in a given classification problem. If the latter case is assumed, then an ordering of the domain according to a semantic property could be 'short, average, tall, very tall', reflecting increasing height described by these adjectives. By an 'ordering according to a syntactic property' we mean, e.g., an alphabetic order. A syntactic ordering is used when a domain is a set of names of independent objects, e.g., a set of first names {Helen, Barbara, Hanna, Renata} (a variable with this domain is called a nominal or cartesian variable).

If a domain of a variable is an interval of a real line (i.e., when the variable is continuous), then this domain is quantized into discrete units. These units are assigned positions $0,1,2,\ldots,\not{d}$ which are taken as values of an AQVAL/1-variable (the problem of how to quantize continuous variables goes beyond the scope of this paper).

2.2 VL_1 System

A full definition of the variable-valued logic system VL_1 was given in papers[3,6]. Here we will restrict ourselves to describing only a few basic concepts and to illustrating them by examples.

The well-formed formulas in VL_1 (VL_1 formulas) are interpreted as expressions of a function (called a VL function):

$$f: \; E = D_1 \times D_2 \times \ldots \times D_n \rightarrow D \qquad (3)$$

where $D = \{0,1,2,\ldots,\not{d}\}$ is called an output variable domain.

The VL_1 system is complete, that is for every function (3), there exists at least one VL_1 formula which is an expression of this function. Following is a simple example of a VL_1 formula and its interpretation.

Example 1.

$$3[x_2 \geq 2][x_4 = 2{:}5,7][x_5 \neq 0] \vee 2[x_3 = 0,3] \vee 1[x_1 \leq 3][x_2 = 3] \qquad (4)$$

The forms in brackets are called selectors. They represent conditions which a value of the variable within a selector must fulfill in order to satisfy the selector. If a selector is satisfied, then it is assigned the highest value in the domain D, that is $\not{d}$, otherwise value 0. Constants outside the selectors are elements of domain D. Concatenations of selectors and a constant (from D) are interpreted as logical products (conjunctions) and are called terms. Logical products are evaluated as the minimum of values of their components (the value of a constant is the constant itself). Terms are linked by the symbol '$\vee$' denoting a logical sum (disjunction). The logical sum is evaluated as the maximum of all the values of the terms. A VL_1 formula in the form of a logical sum of terms (e.g., (4)) is called a disjunctive simple VL_1 formula or, simply, a DVL_1 formula.

Assuming that the domains of variables x_1, x_2, x_3, x_4 and x_5 are respectively:

$D_1 = \{0,1,2,3\}$ $\quad D_2 = \{0,1,2,3,4\}$ $\quad D_3 = \{0,1,2,3\}$
$D_4 = \{0,1,2,3,4,5,6,7\}$ $\quad D_5 = \{0,1,2\}$

and the domain of values of the formula is $D = \{0,1,2,3\}$, the formula (4) is interpreted as an expression of a function:

$$f: \; D_1 \times D_2 \times D_3 \times D_4 \times D_5 \rightarrow D \qquad (5)$$

The formula (4) is assigned value 3 (has value 3), if the value of variable x_2 is greater than or equal to 2, the value of variable x_4 is between 2 and 5, inclusively, or is 7, and the value of variable x_5 is not 0. (Note that the values of variable x_1 and x_3 have no affect on this outcome.) The formula has value 2, if the above compound condition does not hold (i.e., if one or more selectors is not satisfied), and the value of variable x_3 is 0 or 3. If both of the above compound conditions do not hold, and, if the value of variable x_1 is smaller than or equal to 3 and the value of variable x_2 is 3, then the formula has value 1.

If none of the above compound conditions hold, the formula has value 0.

Example 2.

$$2[x_2 + x_3 = 1,2] \setminus 1[x_1 = 0,4] \qquad (6)$$

This formula illustrates how a symmetry with regard to some or all variables can be expressed in the VL_1 system and, also, illustrates the so-called exception operation (denoted by $\setminus$). The formula is interpreted as follows: it has value 2 if the values of x_2 and x_3 arithmetically sum to 1 or 2, except when variable x_1 has value 0 or 4. In the latter case, the formula is assigned value 1. If none of the above conditions hold, the formula has value 0. (A selector which has variables connected by the operator +, is called a symmetric selector.)

It can be seen from the above examples, that the VL_1 system expresses VL functions by setting various conditions on the values of variables and by linking these conditions by logical connectives.

2.3 AQVAL/1 Program

A relation between a universe of object representations and a set of classes to which objects are assigned is called a classification relation. An expression of this relation in the form of a mathematical formula or an algorithm is called a classification rule[7]. Suppose that in a given classification problem, the universe of object representations is an event space E and that the set of classes is D (3). Thus, a VL function $f: E \rightarrow D$ can be interpreted as a classification relation, and a VL_1 expression of this function as a classification rule.

For a given VL function, there exist, in general, very many different VL_1 expressions of this function. These expressions may involve different numbers of variables and different numbers of operations and, consequently, different computational costs will be involved in handling, storing, modifying, and evaluating these expressions. Thus, the problem arises of how to determine, for a given classification relation (in the form of a VL function), a VL_1 expression of this relation which will be the most desired (optimal) according to a certain optimality functional.

A solution to the above problem has been achieved by developing a computer program AQVAL/1*, which, for a given VL function, produces a DVL_1 (disjunctive simple) expression of it, which is near-optimal, or optimal, according to an assumed optimality functional. The program also produces a measure Δ of the maximum possible difference, in the number of terms, between the formula obtained and a formula with the minimum number of terms. Thus, if $\Delta = 0$, the formula obtained has the absolutely minimal number of terms which is possible in a DVL_1 expression of the given VL function.

A VL function f for which an optimal expression is desired can be specified to the AQVAL/1 program by:

1. listing sets of events from E, such that each set includes all known events for which a VL function f takes the same value k, $k = 0,1,2,\ldots,\not{d}$. That is, by listing sets:

   $$F^k = \{e = (\dot{x}_1,\dot{x}_2,\ldots,\dot{x}_n) \mid f(e) = k\}, \qquad (7)$$

   where $k = 0,1,\ldots,\not{d}$, and $\dot{x}_i$ denotes a value of variable x_i.

2. a DVL_1 expression of the function f (in this case AQVAL/1 will try to optimize, if possible, the given expression according to the user-specified optimality functional).

The functional description of AQVAL/1, and other information pertinent to this program, are described by Michalski[3].

2.4 Dependent and Independent VL_1 Class Descriptions

We will describe here two methods of using AQVAL/1 to determine optimal classification rules according to an assumed optimality functional:

- method DCD, which produces the so-called Dependent Class Descriptions, and
- method ICD, which produces the so-called Independent Class Descriptions.

2.4.1 Method DCD. Assume that the a priori probability that an object (which is to be classified) is from class k, $k = 0,1,2,\ldots,\not{d}$, is $p(k)$. Assume further, that the probabilities $p(k)$, $k = 0,1,2,\ldots,\not{d}$, are ordered:

$$p(\not{d}) > p(\not{d}-1) > \ldots > p(1) > p(0) \qquad (8)$$

and remain constant in time.

Suppose now that $C(k)$, $k = 0,1,2,\ldots,\not{d}$, denote descriptions of classes k, respectively. These descriptions $C(k)$ are statements of certain conditions for the values of variables x_i, $i = 1,2,\ldots,n$, (where x_i represent descriptors characterizing objects). If the conditions in a description $C(\dot{k})$ (where $\dot{k}$ denotes a particular value of k), are satisfied, then the object is classified as belonging to class $\dot{k}$. An assumption is made here, that descriptions are evaluated one at a time, and also that if more than one of the descriptions is satisfied, then the object is classified as belonging to the class with the highest a priori probability. Under these assumptions, to minimize the average number of descriptions which are evaluated before a classification decision is made, the order of evaluation of the descriptions should correspond to the descending order of a priori class probabilities, i.e., first should be evaluated description $C(\not{d})$, then $C(\not{d}-1)$, and so on to $C(0)$.

The fact that descriptions are always evaluated in the same order can be used to simplify these descriptions (due to the 'sieve effect'). Namely, the conditions in a given description $C(\dot{k})$, whose only purpose is to distinguish class $\dot{k}$ from classes $\dot{k}+1$, $\dot{k}+2$, ..., $\not{d}$ can be removed from the description $C(\dot{k})$. To achieve a classification rule which takes advantage of the above assumption, the AQVAL/1 program should be used in the following way. Elements of the domain D should be assigned to classes k in such a way that $\not{d}$ represents the class with the highest a priori class probability, $\not{d}-1$, the next highest, etc., until 0, which represents the class with the lowest a priori class probability. Learning samples (in the form of events $e \in E$) for classes k, $k = 0,1,\ldots,\not{d}$, should be specified as sets F^k. AQVAL/1 will produce a DVL_1 expression for the function $f\colon E \to D$ specified by sets F^k (i.e., $F^k = \{e \mid f(e) = k\}$, $k = 0,1,\ldots,\not{d}$). The obtained expression will be quasi-optimal according to a user specified optimality functional[3,4] (by 'quasi-optimal' we mean optimal or approximately-optimal).

In the obtained formula, the description of a class $\dot{k}$ is represented by the logical sum of terms with constant $\dot{k}$. There is, obviously, no term with constant 0, thus the class 0 is not represented explicitly in the formula, but implicitly, i.e., those events are assigned to it which do not satisfy any term in the formula. The obtained class descriptions are interrelated in the sense that satisfying a term with constant $\dot{k}$ is a necessary, but not sufficient condition for classifying an object to class $\dot{k}$. The sufficient condition requires also that none of the terms with a constant $k > \dot{k}$ is satisfied. The above explains the name DCD (dependent class descriptions) given to the method described.

2.4.2 Method ICD. This method produces independent class descriptions. It can be used when:

- class descriptions are to be evaluated in parallel, or
- they are to be evaluated sequentially (as in method DCD), but a priori class probabilities vary for each time an object is to be classified.

The latter situation exists, e.g., when the classification system receives, together with the representation of the object to be classified, also a preliminary hypothesis about the class membership of the object.* In this situation, the classification system should first of all test the given hypothesis. Therefore, class descriptions should be independent of each other, which means that satisfying a description of class $\dot{k}$ should be a sufficient condition of classifying an object to class $\dot{k}$.

Such independent descriptions are created by describing each class by a single VL_1 formula which can take only two values 0 or 1, that is, which is an expression of a function $f\colon E \to \{0,1\}$. If the formula assumes value 1, then the class description associated with this formula is considered to be satisfied, if 0 -- not satisfied.

A set of independent class descriptions is obtained by running AQVAL/1 in the so-called 'IC' or 'DC' mode.[3] In the 'IC' ('Intersecting Covers') mode AQVAL/1 produces a set of VL_1 formulas, each being an

---

*The name AQVAL/1 was derived from 'Algorithm AQ Applied for the synthesis of Variable-Valued Logic formulas. The algorithm AQ is a general, very efficient algorithm[8] for solving covering problems (in particular, for minimization of logic functions with large number of variables) which was used in AQVAL/1.

*An example of such a hypothesis is an 'admitting diagnosis' given to a patient admitted to a hospital.

independent generalization of the learning samples of individual classes. Therefore, some formulas describing different classes may (potentially) intersect, i.e., may be satisfied by the same events (however, they will not be simultaneously satisfied by any events from the learning sets). In the 'DC' (Disjoint Covers) mode, all formulas obtained will be pairwise disjoint.

The penalty for having independent class descriptions is that they will usually be 'longer' than dependent class descriptions (i.e., will have more terms and/or more selectors). In the following section we will illustrate both methods, DCD and ICD, by an example.

3. DEVELOPING A CLASSIFICATION SYSTEM USING DCD AND ICD METHODS

Figure 1 presents 14 groups of 'animals*'. Assume that each group includes all characteristic examples of 'animals' of one species. Suppose now that the problem is to design a very efficient classification system which for any 'animal' from Fig. 1 (or 'sufficiently similar' to one in Fig. 1) correctly determines the species to which it belongs.

A. The first step in solving such a problem by the VL_1-based approach, is to design an event space which serves as the universe of representations of all objects (i.e., in this case 'animals'). This step consists of defining a basic set of descriptors and their domains, minimizing the size of the domains and mapping the domains into sets of non-negative integers (see Michalski[7]).

In the example under consideration the set of basic descriptors included 13 descriptors such as: number of black circles on the body (3), number of tails (2), number of crossmarks on tails (3), type of body texture (7), shape of the body (4), etc. (numbers in parentheses denote cardinalities of domains of descriptors).

B. The next step is to develop a classification rule with the help of the AQVAL/1 program.

All animals were described in terms of the assumed 13 descriptors. Descriptions of the members of one species were combined into individual learning sets F_k, $k = 0,1,\ldots,13$. A detailed report of the obtained results is given in paper[7]; here, for the lack of space, we will give only a few explanations.

Method DCD (Dependent Class Descriptions)

To use this method, an order of classes has to be assumed. In our example we have assumed the order according to numbers $0,1,\ldots,13$ in Fig. 1. A minimal (with regard to number of terms) VL_1 formula found by AQVAL/1, describing all 14 classes, had 20 terms and 54 selectors. For illustration, let us interpret obtained descriptions of classes 0 and 7:

$$C(0) = [x_6=1][x_8=0][x_{10}=0]$$

$$C(7) = [x_3=0][x_5=0,1][x_9=1] \vee [x_4=0][x_8=1]$$

According to C(0), an animal belongs to species 0 (Jexems), if it has 2 circles, no triangles and an irregular shape. According to C(7), an animal belongs to species 7, if it does not satisfy descriptions C(i), $i < 7$, and satisfies at least one of the conditions: 1) has one tail with no crossmarks and texture is blank or  , or 2) has no extremities and one or more triangles on the body.[7]

Method ICD (Independent Class Descriptions)

Here no order among classes is assumed. A minimal (with regard to number of terms) VL_1 formula found by AQVAL/1 had 30 terms and 89 selectors (thus, the price for having independent descriptions was 10 terms and 35 selectors). Let us, for comparison, list the obtained independent description C(7) of class 7 ($\hat{C}(0)$ is obviously the same as C(0)):

$$\hat{C}(7) = [x_1=1][x_5=0][x_9=1] \vee [x_4=0][x_6=0][x_8=1]$$

($[x_1=1]$ means 'one black circle', $[x_6=0]$ -- no empty circles).

4. CONCLUDING REMARKS

The approach described in this paper can be especially useful for solving classification problems with very large number of classes, problems which are intrinsically non-linear and/or when only a small number of learning samples is available. Is is currently being investigated for application in areas such as inferential medical decision making[3] geological classification problems, texture discrimination[3] and plant pathology.

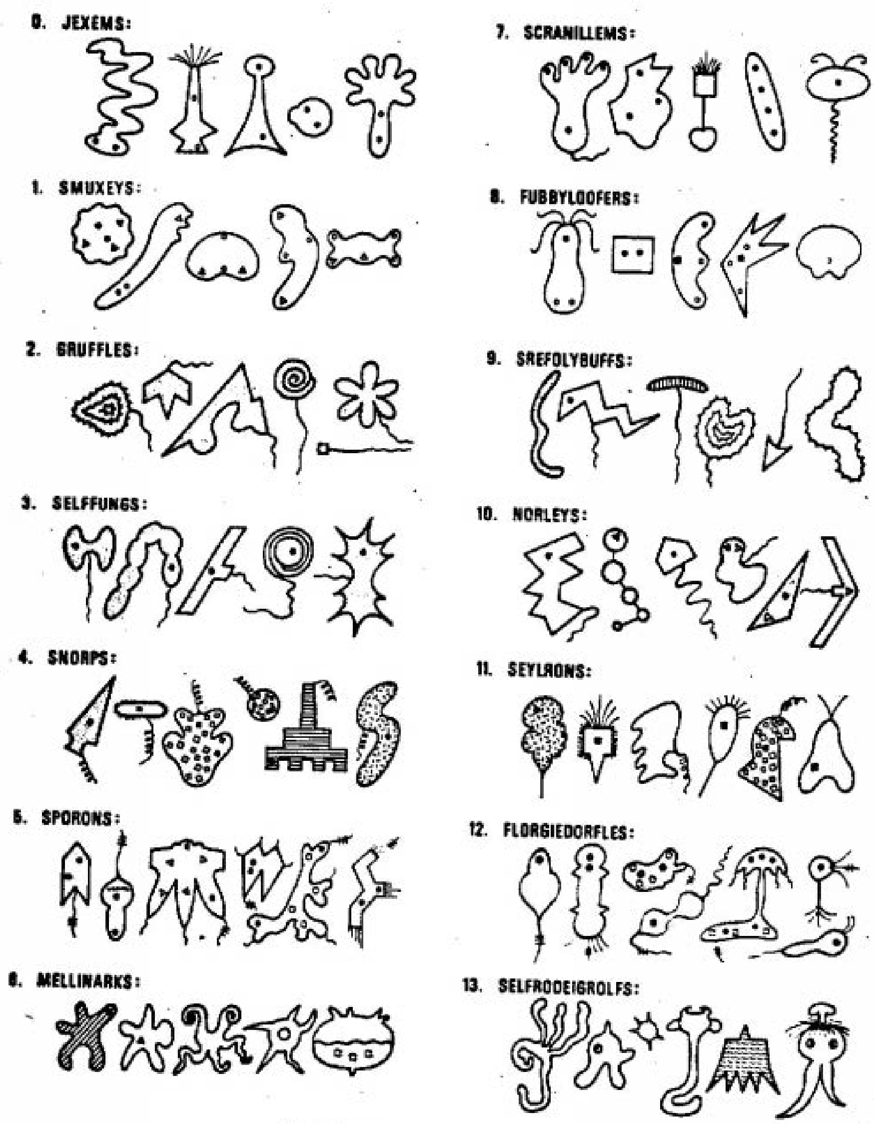
Acknowledgment

The author expresses sincere thanks to Ms. Renata Chimiak and Mr. Shu Kan Yuen for their help in completing the experiment on classification of 'animals'. The author also thanks Mr. Gary Pace for the help in editing the paper, and Mrs. Connie Slovak for her excellent typing of this paper.

References

1. Watanabe, S., Pakvasa, N., Subspace method in pattern recognition, Proceedings of the First International Joint Conference on Pattern Recognition, Washington, D.C., October 30-November 1, 1973.
2. Hayes-Roth, F., A structural approach to pattern learning and the recognition of classificatory power, ibid.
3. Michalski, R. S., AQVAL/1--Computer implementation of a variable-valued logic system VL_1 and examples of its application to pattern recognition, ibid.
4. Michalski, R. S., A variable-valued logic system as applied to picture description and recognition, GRAPHIC LANGUAGES, Proceedings of the IFIP Working Conference on Graphic Languages, pp. 20-47, Vancouver, Canada, May 22-26, 1972.
5. Karpiński, J., Michalski, R. S., A digital system for recognizing handwritten alphanumeric characters (in Polish), Prace Instytutu Automatyki PAN, Warszawa, Zeszyt 35, 1966. (An abbreviated English translation of the paper is available from the author.)
6. Michalski, R. S., VARIABLE-VALUED LOGIC: System VL_1, Proceedings of the 1974 International Symposium on Multiple-Valued Logic, West Virginia University, Morgantown, W.V., May 29-31, 1974.
7. Michalski, R. S., A logistic approach to optimal classification into large number of classes, submitted for publication at PATTERN RECOGNITION.
8. Michalski, R. S., On the quasi-minimal solution of the general covering problem, Proceedings of the V International Symposium on Information Processing (FCIP 69), Vol. A3 (Switching Circuits), Yugoslavia, Bled, October 8-11, 1969.

*The 'animals' were selected (after slight modification) from problem cards in the book 'Altitude Games and Problems', by Educational Development Center, Inc. published by the Webster Division of McGraw-Hill Book Co., 1968.



14 SPECIES OF 'ANIMALS'

Figure 1