PROCEEDINGS OF THE
INTERNATIONAL MEETING
ON ADVANCES IN LEARNING

IMAL - 86

Les Arc, July 28th - Auqust 1st 1986

Unite Associee au CNRS UA 410, AL KHOCWARIZMI

Rapport de Recherche No 290

PROCEDINGS OF IMAL - 86

INTERNATONAL MEETING ON ADYANCES IN LEARNING

Les Arcs (France) | July 8th - August 1st, 1986

ACKNOWLEDGEMENTS

This meeting is organised by “Association Francaise pour l’Aprentissage Symbolique Automatique". Its initiators and organisers are Y Kodratoff and R. Michalski.

It has been sponsored by USARG, DIGITAL, and CGE.

Without funding of USARG and DIGITAL, its organization would have been impossible.

Let us thank their real interest in Artificial Intelligence and Machine learning.

Programme Committee

The programme committee is made of

the invited speakers:

Yves Kodratoff, Pat Langley, Michael Lebowitz. Ryszard Michalski, Bruce Porter, Tom Mitchell, Roger Shank.

the commentators:

I. Bratko, P. Brazdil, R. Holte, R. Stepp.

SUMMARY

The papers of Mitchell, Lebowitz, Michalski. Langley & Nordhausen, and Shank were received too late for inclusion in this volume. They will be distributed to participants at the beginning of the meeting.

CERCLE TECHNICAL REPORT NO. 6

CONCEPT LEARNING: ALTERNATIVE METHODS OF FOCUSSING

By

DAVIO J. GILMORE

Centre for Research on Computers and Learning

University of Lancaster
Lancaster, LA1 4YR
United Kingdom

CONCEPT LEARNING: ALTERNATIVE METHODS OF FOCUSSING

by

David J. Gilmore,

Centre for Research on Computers and Learning, University of Lancaster,

Lancaster, LA1 4YR. England.

ABSTRACT

This paper discusses the standard focussing algorithm and discusses its strengths and weaknesses. Two alternative versions of the algorithm (POSNEG and MULTI) are presented, which do not suffer from some of the weaknesses of the standard algorithm, and which can learn some disjunctive concepts. A closer analysis of the standard algorithm and POSNEG reveals that they are closely related to a common underlying algorithm, but that they make different assumptions about how to take advantage of the structure of the description space. These assumptions are termed the Containment Assumption and the Structured Negative Assumption.

It is suggested that the Structured Negative Assumption is preferable because it produces a more robust algorithm (POSNEG), which can be easily generalised to the learning of multiple concepts (MULTI). However, the main conclusion of the analysis is that the description space is the crucial factor in the success of concept learning, and that research should be aimed at ways of creating and adjusting the structure of the description space while learning. Such research could be of general use to machine learning, outside the particular domain of concept learning.

1. INTRODUCTION

Focussing is an important technique for learning rules and concepts (Bundy, Plummer and Silver, 1985), and although it is not a complex technique it has received very little attention in the literature (an exception is Wielemaker and Bundy, 1985). Our interest was stimulated by the fact that a very similar algorithm can be derived from work by Bruner, Goodnow and Austin (1956) as a psychological process of concept learning. Our current research (on intelligent tutoring systems) requires a machine learning technique which has psychological plausibility and therefore focussing was chosen as the relevant technique. The research reported in this paper was intended to develop the focussing algorithm of Bundy et al, into something more closely related to human learning. In fact, the result was a fuller understanding of the algorithm itself, along with some possibilities for psychological improvements. This paper concentrates on the discussion of the focussing algorithm itself.

To begin with, the standard focussing algorithm is described, along with a simple, but typical, example. This leads on to a consideration of the algorithm's main strengths and weaknesses, which give rise to an alternative algorithm (called POSNEG), which seems to be more powerful than the standard. The next section of the paper tries to understand the nature of this extra power, and it presents an alternative perspective on the nature of the description space and the algorithm. This reveals some assumptions underlying both versions of the algorithm, suggesting that both are specific versions of a common focussing algorithm. From these assumptions a third algorithm (MULTI) is suggested, which can handle multiple concepts at once, and which can handle some disjunctive concepts. However, at the end of these discussions it becomes clear that the important part of the success of focussing lies not in the algorithm, but in the description space.

For the sake of clarity one piece of terminology must be defined:- ***includes*** is used to refer to concepts including an example (ie. positive examples are included in the concept, and negative examples are excluded). In contrast, ***contains*** refers to a concept including all examples defined by some other concept (ie. the concept coloured objects contains the concept red objects).

2. **FOCUSSING**

Focussing is a concept learning algorithm, which requires positive and negative examples of the concept to be learnt. These examples are expressed as a collection of values of pre-specified attributes. The algorithm is based on a description space, which represents all the possible concepts which may occur: each concept being described as a set of values, one for each attribute. The description space is a set of trees: each tree describing the structure of the values of a particular attribute. The algorithm works by maintaining "markers" on these trees, so as to represent a maximally general concept (***G***), and a maximally scientific concept (***S***). The maximally general concept is initially marked by the root nodes of each tree, representing any value on each attribute. This is because some examples are necessary before any concept can be ruled out. Likewise the maximally specific concept is initially marked as having no value on any tree, since nothing more specific can be claimed about the concept. Positive examples cause ***S*** to be altered, through generalisation, while negative examples affect ***G*** through a process of discrimination. The.generalisation process moves the "markers" for ***S*** up the tree towards ***G***, such that the concept described by the new "markers" will include the new positive example. This can always be achieved in only one way, and therefore ***S*** is always a single concept. However, the discrimination process moves the ***G*** markers" down the tree such that the negative example is excluded from the concept described by ***G***. This cannot always be done uniquely which means that ***G*** has to be described as a set of concepts (with possibly only one member). To avoid confusion ***S*** will also be described as a set (of one concept only). When the markers for ***S*** and ***G*** coincide it is assumed that learning is complete and the concept they describe is the target concept.

Using Holte's (1986) approach to describing such systems, Figure 1 represents focussing as a Learning System, a Performance System and a Performance System Declarative Aspect. The Learning System extracts information from the presented examples, whereas the Performance System classifies unlabelled examples. The Performance System Declarative Aspect is the representation used for passing information between the two parts of the system.

In standard focussing the Learning System performs both generalisation and discrimination, and the Declarative Aspect is two sets, one - ***S*** - a singleton and the other - ***G*** - with many members. This lack of duality between the treatment of positive and negative examples and within the Declarative Aspect is much emphasised by Bundy et al (1985), and is one of the complications within the standard algorithm. The task of the Performance System is to discover whether an unlabelled example is included in the concept in S (respond "Yes"), or excluded from the concepts in G (respond "No"), or neither of these (respond "Don't know"). This task is complicated by the fact that G may contain more than one concept, in which case the response is "No- only if the example is excluded from all the concepts in G.

2.1 **Example 1**

Figure 2 displays the description space for all the examples given in this paper. Although this tree is binary, none of the algorithms described depends upon this fact. Clearly it is a very limited space, but it is sufficient to demonstrate the important points. In the following example the concept to be learnt is any black object". initially the markers for G are at the top of the tree, and the markers for S are at the bottom. Thus,

G = { <anycolour anyshape> }; S = { <nocolour noshape> }

(1) Positive example: <black triangle>.

G = { <anycolour anyshape> }; S = { <black triangle> }.

Given this example, the maximally specific concept must be that example. There is no reason for G to be altered.

(2) Negative example: <red circle>.

G = { <monochrome anyshape> <anycolour pointed> }; S = { <black triangle> }.

The maximally general concept must now exclude red circles.

This can be done by either excluding red objects from the concept, or circular objects. But G's markers need only move towards S, since eventually S and G must coincide. Thus, G now contains two possible concepts, each of which would lead to a correct classification of the two examples seen so far.

(3) Positive example: <black oval>.

G = { <monochrome anyshape> }; S { <black anyshape> }.

This example is inconsistent with one of the concepts in G (pointed objects), and therefore this concept is removed from G. Also, S is generalised to include this example.

(4) Unlabelled example: <red triangle>.

This example is not included in the concept in S, nor is it included in the concepts in G and thus, it cannot be a member of the concept being learnt. When the system is then told that "No" was the correct answer, no changes are necessary in G or S.

G { monochrome anyshape> }; S = { <black anyshape> }.

(5) Negative example: <white square>.

G = { <black anyshape> }: S = { <black anyshape> }.

Since this example is included in G's concept, it is necessary for G to be altered by discrimination. This produces two possibilities, but only one of which is towards S.

(6) Unlabelled example: <black circle>.

At this point, when G and S are equal, learning is complete and unlabelled examples can be classified without doubt. Since black circles are clearly included in S's concept, the response should be "Yes". Also, it is clear that the concept black objects' has been successfully learnt.

2.2. **Strengths and Weaknesses of Focussing**

The above example illustrates most of the important features of focussing, and from it we can comment on some of the strengths and weaknesses of the algorithm.

2.2.1. **Strengths**

1) Learning is incremental. This is different from, for example, Quinlan's ID3 classification algorithm (Quinlan, 1983), which requires all examples to be presented simultaneously.

(2) Also unlike 1D3, focussing produces a compact representation of the concept being learnt. The algorithm is not distracted by the presence of irrelevant attributes in the description space.

(3) Memory for all the training instances is not necessary for learning, since all relevant information is extracted from each example when it is presented.

(4) There is always a partially learnt concept which can be used to classify any unlabelled examples. It is possible to describe this concept as Just S, or Just G or as both (as the system above does), in which case the classification can include “Don't know".

2.2.2. **Weaknesses**

(1) The algorithm can only learn conjunctive concepts, since disjunctives lead to overgeneralisation and inconsistencies.

(2) The presence of far misses (ie. where discrimination does not lead to a single solution) can lead to a substantial increase in processing load, since they generate a lot of alternatives within G.

(3) There is a lack of duality in the processing of positive and negative examples.

(4) There is no capability for handling noisy data.

(5) The description space, which must be specified in advance of learning, may be inadequate.

2.3. **Psychological Issues and Improvements**

Of these strengths and weaknesses, it is apparent that most of the strengths are important for a psychological model of concept learning. For example, human learning is clearly incremental, and also it succeeds without storing all of the training examples. People can also make judgments about unknown examples, even when only a few examples have been observed.

Likewise the weaknesses are important. Most of them would be serious weaknesses in a psychological model of concept learning. The last two, however, have proved to be problematic for much of the machine learning research, not just focussing. Some attempts have bean made to deal with these problems, but many of them succeed only at the expense of some of the strengths. Here, we will discuss attempts to tackle these problems which do not compromise the strengths of focussing.

The problem of 'far misses' is handled by Bruner et al (1956), in their psychological model, by processing information from positive examples only. In doing this they have taken S as their current concept, and not used G at all. This means that they cannot tell when learning is complete, though it is not clear that this is necessary for a psychological model.

Winston (1975) overcame the problem of 'far misses' by only allowing 'near misses' to be presented as negative examples. Unfortunately, as van Someren (1986) points out, focussing based on 'near misses' alone is inadequate, because the nature of a 'near miss' is dependent upon the structure of the description space. Van Someren's solution is to allow domain knowledge to adapt the description space, in order that previous 'far misses' become 'near misses*. Similarly, Wielemaker and Bundy (1985) tackle the problem of the description space, by adjusting the structure of individual trees. The approach we take here is to reformulate the focussing algorithm itself, in an attempt to solve the problem of both 'far misses and disjunctive concepts. The discrimination process is, therefore, our major interest and it is this which we propose to adjust.

3. **AN ALTERNATIVE APPROACH (POSNEG)**

3.1. **The Algorithm**

Figure 3 displays POSNEG according to Holte's distinctions. The main changes are that the Learning System performs generalisation only, with the Performance System using discrimination. This means that the PSDA is simply two maximally specific concepts, one for the positive examples and one for the negative examples. The generalisation process is identical to that for standard focussing, which means that after each example there is a unique maximally specific concept (positive or negative, depending on the status of the example). The maximally specific concept for the positive examples is referred to as POS, and that for the negative examples is referred to as NEG. In this way there is no distinction between far misses and near misses, since they are all summarised as negative examples. Thus, two of the weaknesses of the standard algorithm are tackled together.

The discrimination process is very similar in style to standard focussing, with the difference that it occurs within the Performance System and, instead of discriminating a negative example from the set G. it discriminates POS from NEG. This means that the results of the discrimination process define the concept learnt so far, rather than the current set of maximally general concepts, which

have to be stored for comparison with later examples. Thus the task of the Performance System is simply to compare an unlabelled example with the result of discriminating POS and NEG.

3.1. **The discrimination process**

(a) If POS = NEG then no discrimination is possible. There is no description of the concept within the description space (assuming the examples were correctly classified).

(b) If POS does not contain NEG or overlaps with it then POS is the concept. Oveerlapping can occur when POS or NEG becomes overgeneralised. It is assumed that the negative examples are more likely to be the cause of overgeneralisation.

(c) If POS contains NEG then resolve (see d) the values of the first attribute and attach them to the remaining values in POS, and attach the first value in POS to the result of discriminating the remainder of POS and NEG. In this way at least two descriptions of the concept are generated. The Performance System will respond "Yes" if the example is included in any of them (ie. the concept can be disjunctive).

(d) To resolve two values of a particular attribute is to obtain all the nodes of the attribute tree, which are below the posi- tive value and not above the negative (this is like discrimination in the standard algorithm). If there is more than one such node, then they all define the concept for this attribute. This process allows recovery from overgeneralisation.

Having thus performed discrimination, al the performance system has to do is to evaluate whether the unlabelled example is included in any concept in the result. This gives rise to a simple "Yes"/"No" response.

3.2 **Example 1 Again**

The description space and the concept to be learnt are the same as for the previous example (ie. all black objects).

NEG = <nocolour noshape>; POS = <nocolour noshape>.

(1) Positive example: <black triangle>.

NEG = <nocolour noshape>; POS = <black triangle>.

(2) Negative example: <red circle>.

NEG = <red circle>: POS = <black triangle>.

POS is not altered following a negative example, so this example simply updates NEG.

(3) Positive example: <black oval>.

NEG 2 <red circle>: POS = <black anyshape>.

(4) Unlabelled example: <red triangle>.

At this point the Performance System is required and discrimination occurs. The result of the discrimination process is the concept <black anyshape>, since POS does not contain NEG and therefore the result of discrimination is POS. Following discrimination the Performance System checks whether the example is included in the result. In this instance it is not and, thus the response is "No". When the system is told that this is indeed the correct answer then NEG will be updated by the now labelled example. Notice that this is the first time that discrimination has been performed, and that the result is not stored.

NEG = <red anyshape>; POS = <black anyshape>.

(5) Negative example: <white square>

NEG = <anycolour anyshape>; POS = <black anyshape>.

At this point learning is complete as far as negative examples go. Generalisation of the negative examples has the produced the most general concept, and no more information can be obtained from negative examples. Learning will continue when further positive examples are seen.

(6) Unlabelled example: <black oval>.

The result of the discrimination process is <black anyshape>, which includes this example and so the response for this example is "Yes". Since <black anyshape> is the concept to be learnt, learning is also complete for the positive examples. But POSNEG, unlike the standard algorithm, has no way of knowing this.

3.3 **Main Differences with Standard Focussing**

I have already described the main differences within the algorithm. In this section I shall outline four implications of these differences:-

(1) Because POSNEG is revolutionary, rather than evolutionary, there is a reduction in the potential for high demands on memory. In standard focussing the results of the discrimination process must be stored, because they are part of the Learning System, but in POSNEG, the results of discrimination are used by the Performance System and then forgotten.

(2) The Performance System compares unlabelled examples with the concepts resulting from the discrimination process. This yields a simple "Yes"/"No" response, whereas standard focussing can respond "Don't know". However, the Performance System could be easily changed to offer all three responses, simply by checking the example against both POS and NEG. Such changes would not affect the learning system.

(3) POSNEG is symmetrical, since positive and negative examples are treated in exactly the same way. POSNEG's performance would not be affected if the negative examples were labelled as positive and the positive negative, but standard focussing would almost certainly fail.

(4) POSNEG does not detect inconsistencies. Since either POS or NEG can become overgeneralised (due to the structure of the description space), legitimate examples may appear to be inconsistent. If there is a genuine inconsistency (eg. due to noisy data) then the discrimination process will fail to work. However, this failure to detect inconsistency at the time of presentation enables POSNEG to be able to learn a greater range of concepts than standard focussing, as in the following example.

3.4 **Example 2**

Again the description space is as in Figure 2. The example will be given in both POSNEG and the standard algorithm.

G = { <anycolour anyshape> }; S { <nocolour noshape> }

NEG = <nocolour noshape>; POS = <nocolour noshape>.

Positive example: <red circle>.

G { anycolour anyshape> } S = { <red circle> }.

NEG <nocolour noshape>: POS = <red circle>.

(2) Positive example: <green oval>.

G = { <anycolour anyshape> }; S = { <coloured round> }.

NEG = <nocolour noshape>: POS = <coloured round>.

(3) Negative example: <black triangle>.

G { <anycolour round> <coloured anyshape> }: S { <coloured round> }.

NEG <black triangle>: POS = <coloured round>.

(4) Negative example: <white square>.

G { <anycolour round> <coloured anyshape> }: S = { <coloured round> }.

NEG = <monochrome pointed>; POS = <coloured round>.

(5) Positive example: <red triangle>.

G = { <coloured anyshape> }: S { <coloured anyshape> }.

NEG = <monochrome pointed>: POS : <coloured anyshape>.

At this point the standard algorithm will terminate because G and S are equivalent, the concept being <coloured anyshape>. The result of POSNEG's discrimination process is also <coloured anyshape>, since this excludes NEG.

(6) Positive example: <white oval>.

The standard algorithm fails at this point because the example is not contained within S, but POSNEG simply incorporates the information into POS.
NEG = <monochrome pointed>: POS : <anycolour anyshape>.

The discrimination process finds that POS contains NEG and thefirst two values are 'resolved', giving the result "coloured". One result of the discrimination process is thus <coloured anyshape>, whilst the other depends upon the resolution of "pointed" and “anyshape”. The result of this ‘resolve’ leads to the other result of the discrimination process, which is <anycolour round>. Thus, the concept being learnt is { <coloured anyshape> <anycolour round>} – ie. ary object which is either coloured or round - a disjunctive concept.

Thus, POSNEG is capable of learning disjunctive concepts, which cannot be handled by the standard algorithm. Since the two algorithms contain basically the same components, it is interesting to wonder exactly why this difference arises - it is not clear why postponing the discrimination process should lead to greater power.

In order to understand exactly why this is the case, it is necessary to reexamine our representation of the description space.

4. **THE STRUCTURE OF THE DESCRIPTION SPACE**

4.1. **A spatial representation**

The tree structures which are commonly used to represent the description space are not the only way of representing it. In fact, using tree structures obscures some important aspects of the focussing algorithm. In this section of the paper the description space will be described not as a tree, but as a physical space, in which the area occupied by positive and negative examples is explicitly represented. Thus, Figure 4 gives an example of the way the description space could look, with a clear boundary around all conceivable objects, and subsets within that representing the positive examples, the negative examples and those contained within {MGC}.

In both algorithms S and POS are equivalent, since they both summarise the area occupied by positive examples. However, whereas NEG summarises the area of negative examples, G summarises its complement (the area not occupied by negative examples). When examined in this way, it is hard to understand why the two algorithms should produce different results, since it would appear that from NEG it should always be possible to calculate G. The reason for the difference, however, lies in the fact that the space in Figure 4 is unstructured; once structure is imposed then assumptions must be made about that structure. since it is within this structure that generalisation and discrimination occur. Each of' the two algorithms described above makes a different assumption about how to utilise this structure, and hence gives different results. Without either of these assumptions the algorithms would, in fact, be the same.

Figure 5 displays a structured space for the description space of Figure 2, indicating the existence of 16 possible objects. Within this space we can mark the area of S, or POS, (and use a * to indicate an actual observed example) and we can also mark G, or NEG, using a - to indicate an observed negative example. The concepts to be learnt are the rows and columns of the space (and combinations thereof). It should be emphasised that this is purely an illustrative representation, since the internal representation of the description space is exactly the same. Figure 6 illustrates example 2 in this form, using the standard algorithm. Initially, G is the whole space, and S is no part of the space. The first positive example causes S to expand (through generalisation) , and the negative examples cause G to shrink (through discrimination). In this way it is possible to see more clearly the significance of a discrimination which generates two or more concepts within G (see Figure 6d). For clarity the two possibilities are labelled GI and G2, the former being necessary to exclude black objects and the latter to exclude triangular objects. These two possibilities are exclusive and the focussing algorithm will use the next examples to try and distinguish them. In Figure 6f a positive example serves to reject G2, even though no monochrome round objects have been observed. Thus, a single example can cause large changes in G and S. This fact prevents the algorithm from coping with the next example, which is monochrome and round. The reason that this occurs is that the standard algorithm has made the assumption that G must always contain S, and therefore when a positive example is observed which is not included in G2 (Fig. 6f), G2 can be rejected. I have termed this assumption, which is not essential, the "Containment Assumption".

The Containment Assumption states, quite reasonably, that any maximally general concept must contain the maximally specific concept. But it is this assumption that decrees that focussing can only learn conjunctive concepts, since the assumption only holds true for such concepts.

Figure 7 illustrates the same example for POSNEG. It reveals that this algorithm depends upon a different assumption, one which I have termed the "Structured Negative Assumption". This assumes that the negative of the concept being learnt will be structured similarly to the concept itself and, thus, that generalisation can be applied to the negative examples. Thus, in figure 7f, monochrome round objects are simply not known about - they have not yet been accepted into POS, or rejected by being in NEG. This assumption is in contrast to the standard algorithm which does not consider the negative of the concept.

However, it also becomes apparent from Figure 7 that POSNEG does not really learn disjunctive concepts. instead, it is learning the reverse of a conjunctive concept. This reveals why the presence of duality is an important aspect of focussing. For the standard algorithm it is very important which are the positive and which are the negative examples, whereas it makes no difference to the symmetrical POSNEG. This suggests that, of the two assumptions, the Structured Negative Assumption may be preferable, since it produces a more robust algorithm.

It is important to realise that if the assumptions are dropped then the performance of the two algorithms is the same. The standard algorithm will generate a set G which contains all concepts except the given negative examples, since it cannot use S to restrict it. POSNEG can only function without its assumption if it simply remembers all negative examples given so far. Thus the two algorithms reduce to the idealised representation of Figure 4. However, without any assumptions the concept learning reduces to little more than a memory task, and no predictions could be made about unlabelled examples.

It has already been suggested that the Structured Negative Assumption may be preferable, because it produces a symmetrical algorithm. A further reason for it to be preferred is that it can be generalised to different learning contexts.

5. **MULTIPLE FOCIJSSING (MULTI)**

Hunt, Marin and Stone (1966) comment how there are in fact very few negative instances of concepts, since almost everything is a positive example of some other concept. Whilst this may not be true for some rule-learning programs, it does seem to be a valid comment about concept learning. The significance of this is that whilst standard focussing can only handle conjunctive concepts, and POSNEG can only manage with conjunctive positives or conjunctive negatives, the Structured Negative Assumption can be applied to a disjunctive concept with a disjunctive negative, if the negative can be split into two or more conjunctive concepts, For the purposes of what follows it will be assumed that the negative concept(s) are naturally defined ones, which are given. It may be possible to develop techniques whereby different divisions of negative instances can be calculated automatically.*

This is similar to the use of rule shells for learning disjunctive concepts (as discussed by Bundy et al. 1985). The main difference, however, is that it is the negative examples which are split into two or more groups, not the positive examples. Also, the groups are labelled from the beginning of learning, rather than from the point where a contradiction occurs. The following program assumes the examples to be correctly labelled.

5.1. Example 3

Using the same description space again, consider the three concepts,

(1) Monochrome, round objects or red objects.

(2) Green objects.

(3) Monochrome, pointed objects.

Both algorithms described so far would fail, if they were required to learn concept 1. The standard algorithm would fail because of inconsistencies and POSNEG would fail because both POS and NEG would be <anycolour anyshape>, which would prevent successful discrimination.

However, POSNEG can be adjusted slightly to handle multiple concepts. The algorithm MULTI can be given labelled examples of all three concepts, and successfully learn all three concepts. However, to achieve this it is necessary that the concepts are both exclusive and exhaustive, since this is an implicit assumption of the discrimination process. The only adjustment necessary for the discrimination process is that instead of discriminatlng POS with NEG, it must discriminate the concept of interest with each or the negative concepts. This is done by first discriminating with one of them, then discriminating the result of this with another, and the result of this with another etc. etc. The resultant list of concepts represents disjunctively the concept of interest.

Consider the following sequence of training examples. Initially

CONI, CON2 and CON3 are all equal to <nocolour noshape>.

(1) Example of concept 1: <red square>.

CONI = <red square>; CON2 = <nocolour noshape>; CON3 = <nocolour ncshape>.

---

* For instance, it may be possible to use some measure which compares the distance of the current example from both the current negative concept and the positive concept. If the example is nearer the positive concept, then a new negative class is begun, otherwise the example is allocated to the

(2) Example of concept 3: <white triangle>.

CON1 = <red square>; CON2 = <nocolour noshape>; CON3 = <white triangle>.

(3) Example of concept 1: <black square>.

CON1 = <red square>; CON2 =<nocolour noshape>; CON3 = <monchrome pointed>.

(4) Example of concept 1: <white circle>.

CON1 = <anycolour anyshape>; CON2 = <nocolour noshape>; CON3 = <monochrome pointed>.

(5) Example of concept 2: <green circle>.

CONI = <anycolour anyshape>: CON2 <green circle>: CON3 = <monochrome pointed>.

(6) Example of concept 2: <green square>.

CONI = <anycolour anyshape>; CON2 <green anyshape>; CON3 = <monochrome pointed>.

Note how as with POSNEG none of the inconsistencies has been detected; this is because no discrimination has occurred. To describe each concept (or to classify an unlabelled example) then discrimination must occur. For concepts CON2 and CON3 this results in themselves being the concept, since they each overlap with, or do not contain the other two concepts. However, CON1 can only be described through the 'resolve' operation. Initially MULTI discriminates CON1 and CON2 which gives the result {<red anyshape> <monochrome anyshape>}, and then each of these is discriminated with CON3, giving the result {<red anyshape><monochrome round>}.

Thus, MULTI can learn more complex disjunctive concepts than POSNEG can. Even so it depends upon the existence of only 1 disjunctive concept, since more than one would lead to overgeneralisation on both of them, and the discrimination process would be unable to distinguish which values of which attributes discriminate the two. However, it is important to realise that this generalisation is only possible with the Structured Negative Assumption, and not with the Containment Assumption, because of the symmetricality of the algorithm. But, apart from the difference in duality, there is not much difference between the algorithms. They use the supplied structure of the description space to generalise and discriminate. Neither of these processes is particularly complicated, and what the spatial representation of the description space makes clear is that the real determinant of the success of focussing is the structure of the description space itself.

6. **SUMMARY AND IMPLICATIONS**

This paper has presented an alternative focussing algorithm, and it has demonstrated that there is a core algorithm underlying both, which simply compares the positive examples seen with all possible positive examples, given the observed negative examples. The standard algorithm and POSNEG (and MULTI, too) make assumptions about the relationship between the structure of the description space and the algorithm, in order to prevent the brute force search of all possibilities.

However, the Containment Assumption used by the standard algorithm produces a program which lacks duality, which can only handle conjunctive concepts and which cannot be generalised easily. By contrast, the Structured Negative Assumption produces a symmetrical algorithm, which handles a small set of disjunctive concepts, and which can be generalised to the learning of more than one concept.

By discussing small examples, and representing the description space as an n-dimensional space, it has become clear that the focussing methods discussed depend upon the existence of conjunctive concepts, even when the main concept is not conjunctive. This may be true for the two assumptions discussed here, but there may be others to which it does not apply. An area of interesting research would be to examine other possible ways of constraining the search for the concept.

However, the main implication of this analysis is the overwhelming importance of the structure of description space. The set of concepts which can be learnt by these algorithms depends crucially on whether the given structure maps onto the structure within the concept. Although the new assumptions described here make the algorithms slightly more powerful, the only way to dramatically improve concept learning performance is to be able to change, dynamically, the structure of the description space. This is being attempted by Wielemaker and Bundy (1985), and by van Someren (1986). However, their approaches are only to be used when inconsistencies occur in the example set, whereas it would seem desirable to find a way of checking for patterns in the examples.

In much of human learning the relevant structure of an attribute's values has to be inferred from the examples. For example. people cope well with concepts which involve numbers, even though such an attribute cannot be readily represented as a tree structure. People can structure a numerical attribute into primes, even numbers, multiples of <n>, etc. etc. An important question, therefore, for machine learning is to discover techniques for dynamically varying the structure of the description space, and to discover whether there are any constraints (psychological or otherwise) on the restructurings which can be performed.

ACKNOWLEDGMENTS

I would like to thank John Self, Stephen Payne and Louise O'Callaghan for their fruitful discussions on the problems of concept learning, both by machines and by people. This research was carried out under a UJK Science and Engineering Research Council grant, number GR/D/16079.

REFERENCES

Bruner, 1.S., Goodnow,).A. and Austin, G.A. (1956) A Studv Of Thinking, Wiley, New York.

Bundy, A., Silver, B. and Plummer, D. (1985) "A review of rule-learning programs." Artificial Intelligence, 27, 137 - 181.

Holte, R.C. (1986) "Alternative information structures in incremental learning systems." Paper presented at the European Working Session on Learning, Orsay, Paris, January, 1986.

Hunt, E.B.. Marin, J. and Stone, P.). (1966) Experiments In Induction, Academic Press, New York.

Quinlan. j.R. (1983) "Learning efficient classification procedures and their application to chess end games." In Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.) Machine Learning: An Artificial Intelligence Approach, Tioga Pub. Co., Palo Alto.

van Someren. M.W. (1986) ..Constructive induction rules: reducing the description space for rule learning." Paper presented at the European Working Session on Learning, Orsay, Paris, January, 1986 .

Wielemaker. J. and Bundy, A. (1985) "Altering the description space for focusslng." Paper presented at Expert Systems - 85 Warwick, England.

Winston, P.H. (1975' "Learning structural descriptions from examples." In Winston, P.H. (Ed) The Psychologv Of Computer Vision, McGraw-Hill.

Figure 1: Standard focussing algorithm using Holte's (1986) representation.

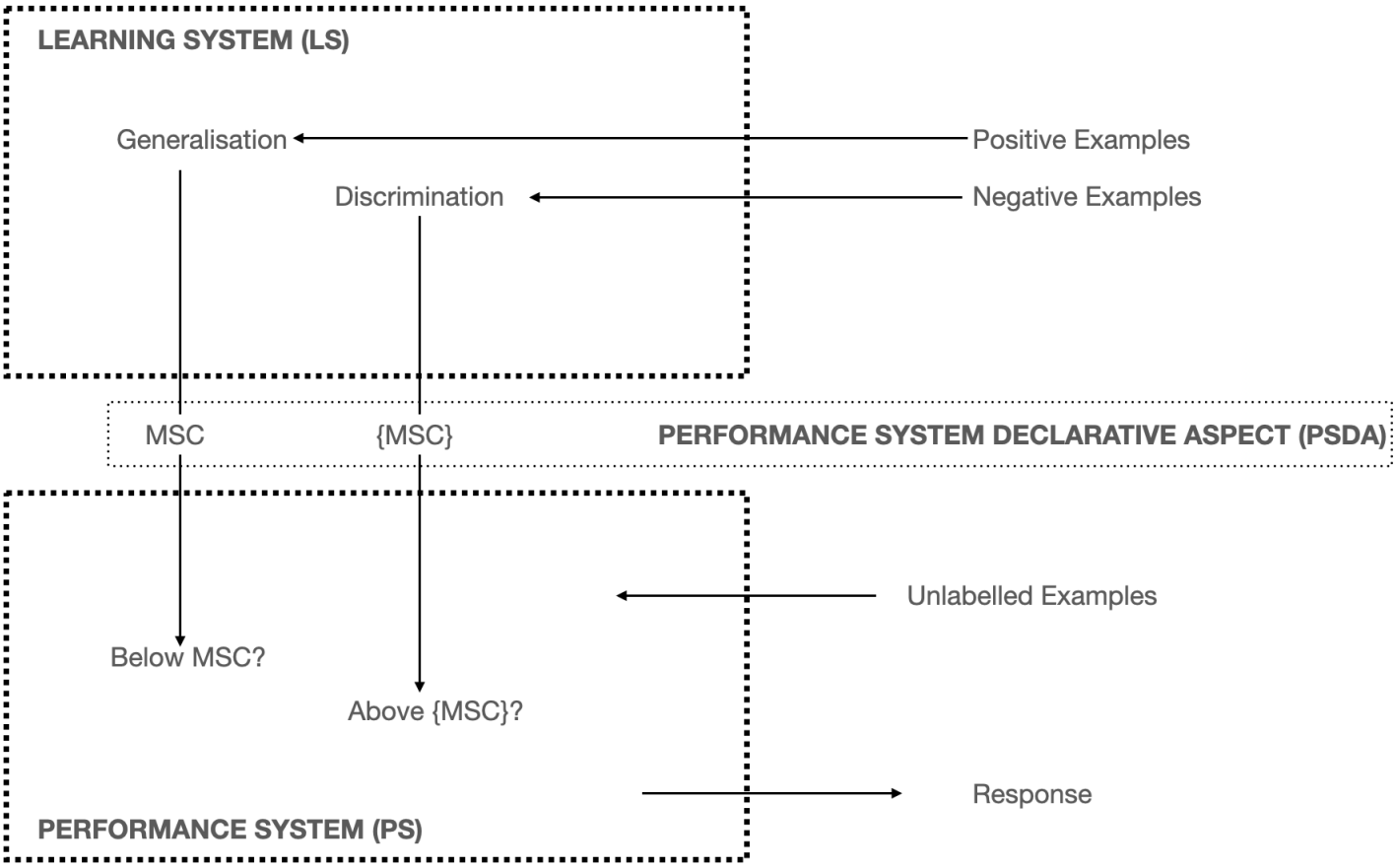


Figure 2: The Description Space used throughout this paper.

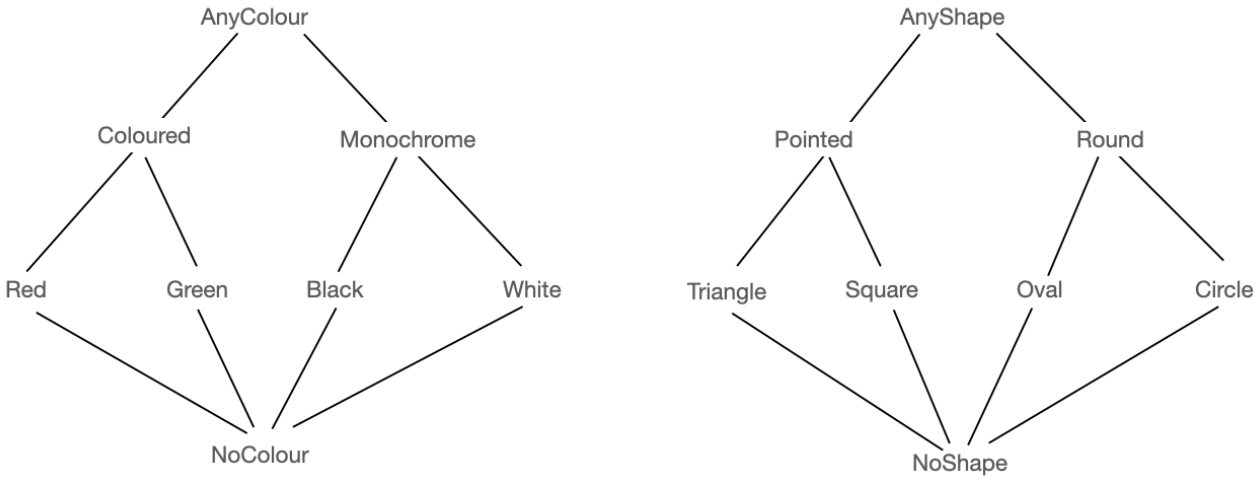


Figure 3: The POSNEG focussing algorithm using Holte's (I86) representation.

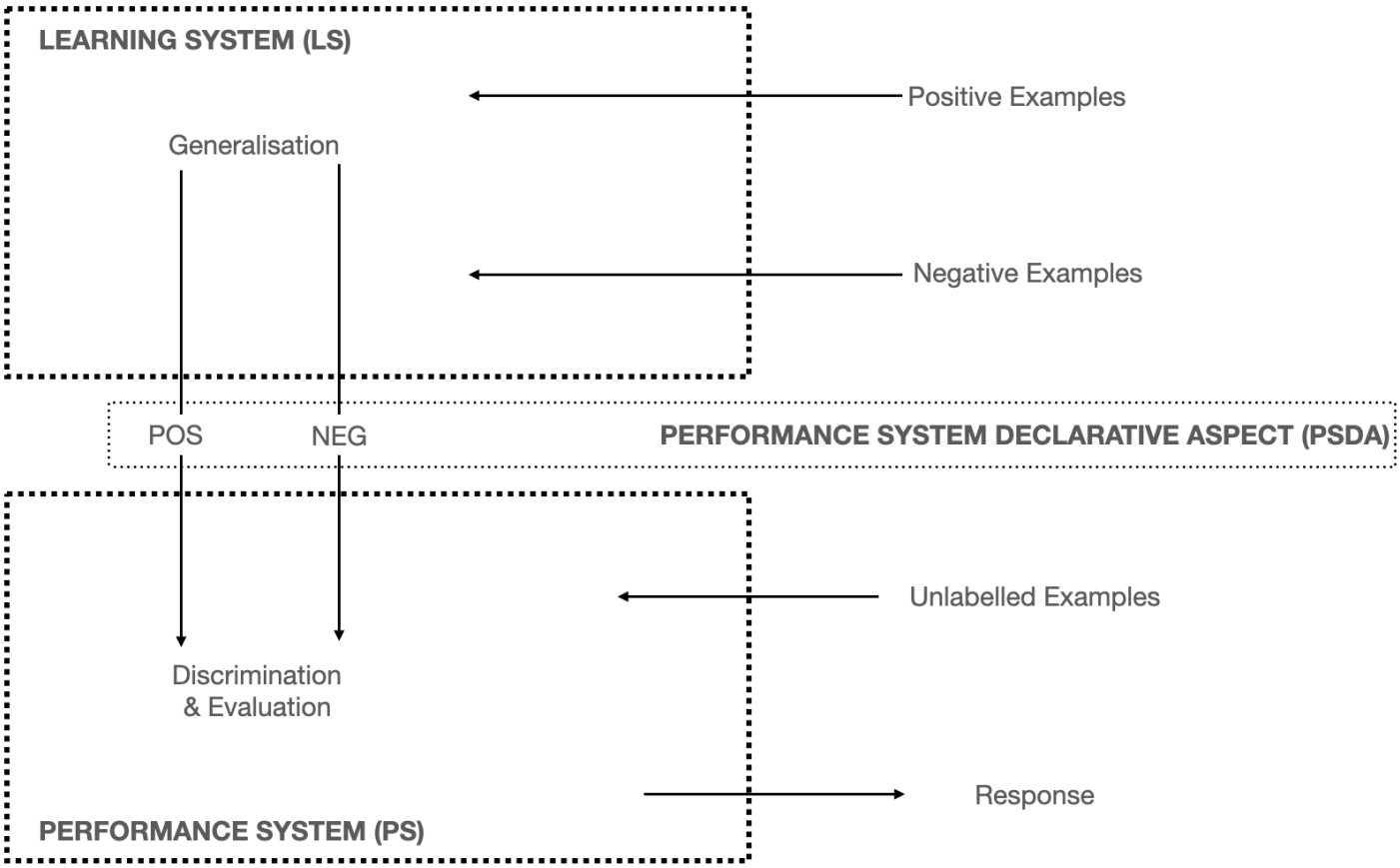


Figure 4: A spatial representation of the description n:,. space. (The outer box contains all possible objects in the space.)

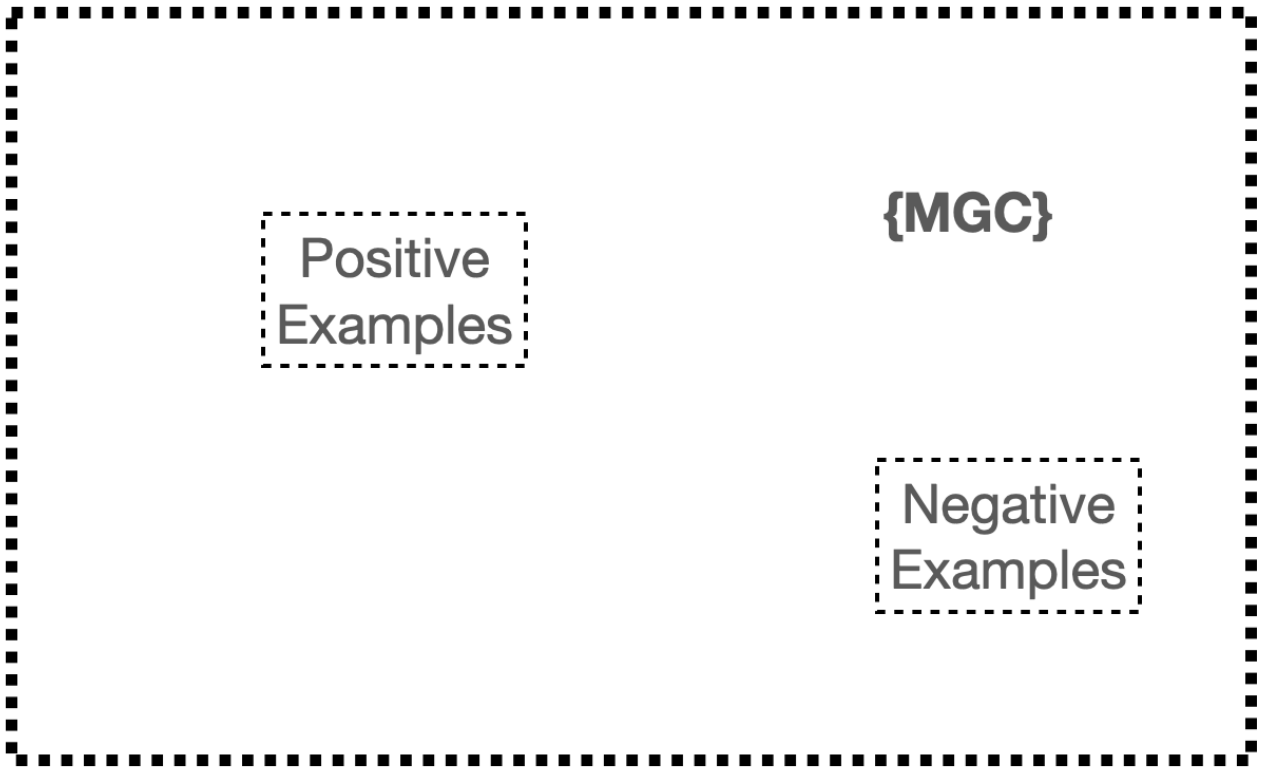
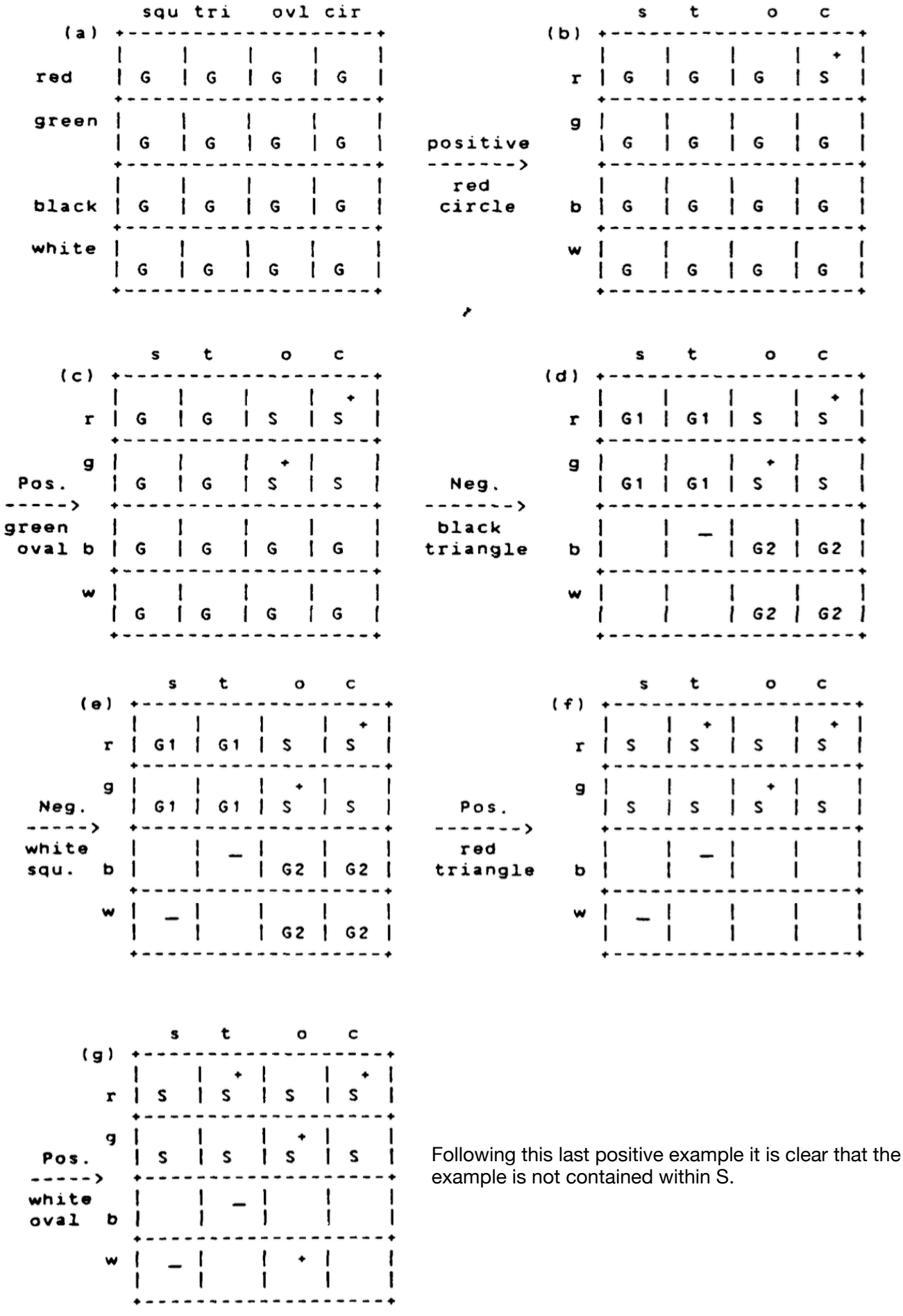


Figure 5: An alternative representation of the description space of figure 2.

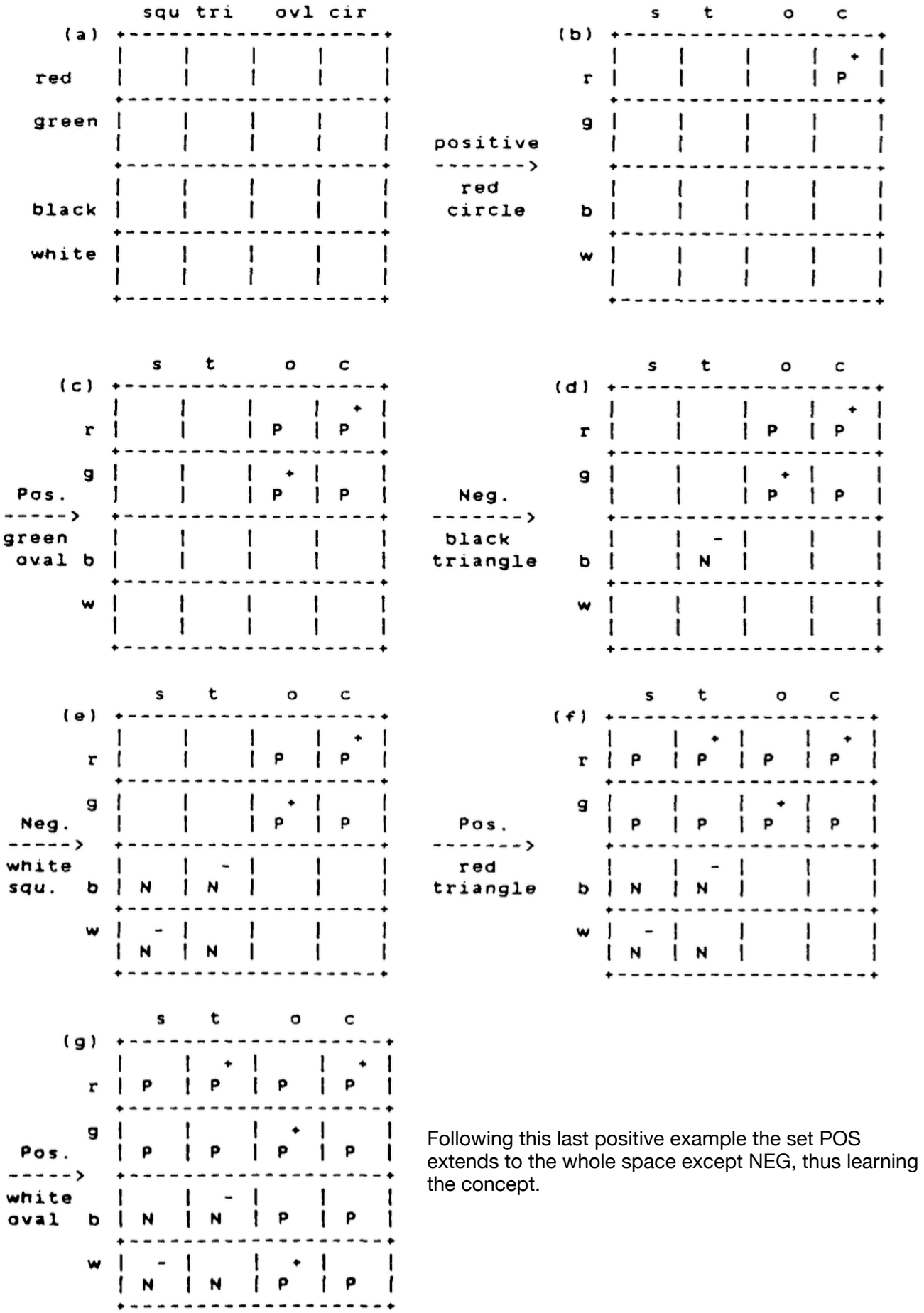
			Shape			
			Pointed		Round	
			Square	Triangle	Oval	Circle
Colour	Coloured	Red				
		Green				
	Monochrome	Black				
		White				

Figure 6: An illustration of the description space during example 2 for the standard algorithm.



Following this last positive example it is clear that the example is not contained within S.

Figure 7: An illustration of the description space during example 2, for POSNEG.



Following this last positive example the set POS extends to the whole space except NEG, thus learning the concept.