MOUSE: A teachable program for learning in conformational analysis

Daniel P. Dolata and W. Patrick Walters

AI in Chemistry Lab, Department of Chemistry, University of Arizona, Tucson, AZ, USA

MOUSE is a teachable program which learns concepts of conformational analysis from examples obtained from WIZARD. The algorithms are presented, and a fully worked example is used to demonstrate how MOUSE learns about the so-called "pentane rule."

Keywords: teachable program, automated learning, artificial intelligence, conformational analysis

INTRODUCTION

A learning program is one which can improve its performance at a given task either by being led by a teacher or by examining a set of examples on its own. There are at least two types of Learning Programs in current usage: neural networks[1] and concept formation programs.[2] Neural networks can provide efficient learning systems, but they currently lack the ability to communicate what they have learned in a human-useable form. They are sometimes referred to as "black box" systems. Concept learning programs attempt to create concepts or hypotheses in a symbolic language which can be understood by other programs and by humans. This paper describes our work in the area of concept learning.

There are many reasons to implement a learning program:

(1) Increased speed in achieving a given task
(2) Reduction of size of knowledge base by refinement of existing knowledge
(3) Learning new concepts by examining raw data
(4) Success at tasks previously beyond scope of knowledge
(5) Tailoring the knowledge base for a specific subdomain (e.g., dyestuffs versus pharmaceuticals).

The previous paper[3] described how WIZARD[4] has achieved the first goal through short-term learning. This paper describes the design and gives an example of the results of MOUSE, a concept learning program which attempts to satisfy Goals 1 through 4 above. MOUSE is a *teachable* program,[5] i.e., it learns from examples that a human teacher has extracted from the results of WIZARD's short-term learning. MOUSE is a separate program from WIZARD, but utilizes the same implementation and symbolic descriptor languages, so that the two programs can eventually be integrated. Once that is done, we plan to extend MOUSE so that it can first become an *apprentice* program which is capable of "watching over the shoulder" of WIZARD as it performs conformational analyses, and will learn without human intervention and selection of examples. The next step is to give MOUSE the ability to be an *explorer* program, which will identify deficiencies in WIZARD's knowledge base and actively direct WIZARD, so as to learn from the results and eliminate those deficiencies.

LONG-TERM LEARNING

The observations obtained in WIZARD's short-term learning are very useful during the analysis of a single molecule, as shown by the increases in speed mentioned in the previous paper. But since the symbols used to describe the observations are specific to the context of the current molecule, these observations cannot be directly applied to any other molecule. The fact that a bad van der Waals interaction is discovered when unit1 in one molecule is a chair cyclohexane and unit2 is a chair cyclohexane, will only be the basis of a valid critic for the analysis of a new molecule if unit1 and unit2 have identical contexts in both molecules. In a set of early experiments, such observations were obtained and converted into new rules by explicitly including context, as shown in Figure 1. In this example WIZARD finds that when both rings of a bicyclo-[3.3.1]-heptane system are chairs there is a bad van der Waals interaction between the interior protons. VLEARN (the failed program) posited that whenever two cyclohexyl rings are adjoined along three atoms that there will be a similar problem when both rings are chairs. However, this is not sufficient context; this rule would eliminate adamantane since the rule doesn't check to see if the offending hydrogens are in fact present. VLEARN failed because it was not able to adjust its contextual information to include that which was necessary and sufficient. The rules were either too specific or too general.

Another problem stemmed from the fact that WIZARD/VLEARN generated between 50 to 1000 such abstract critics for each new molecule. This meant that within a few analyses WIZARD would spend more time searching its knowledge base of learned critics than it would save by

Address reprint requests to Dr. D.P. Dolata at AI in Chemistry Lab, Department of Chemistry, University of Arizona, Tucson, AZ 85721, USA.
Received 20 June 1992; accepted 28 July 1992

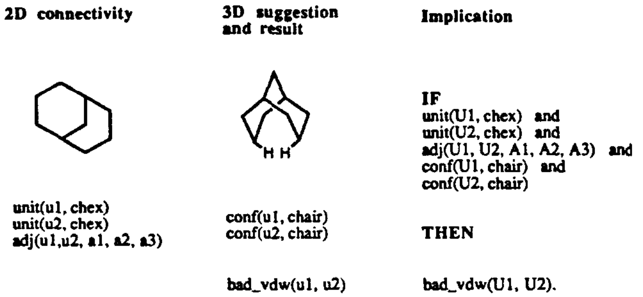

Figure 1. An observation could naively be turned into an implication by combining connectivity information with conformational suggestions and observed results by the process of positing causality. Small letters (u1) indicate specific bindings, capital letters (U1) indicate unbound variables. The term adj(u1, u2, a1, a2, a3) indicates that unit u1 is adjacent to unit u2 along the atoms a1, a2, and a3.

having those critics. This led us to realize that we needed a program which could take numerous examples from WIZARD and learn a minimal set of generalized rules. Since these rules will be useable for many molecules, this is called *long-term learning* to distinguish it from the short-term learning described in the previous paper.

The process of learning generalized concepts from examples requires both positive and negative examples. Positive examples are needed to form the first hypothesis, and then to remove extraneous details and to generalize trivial differences. Negative examples are used to discover which components or relationships are necessary, and which components cannot be part of a positive example. Without both positive and negative examples the rules will never converge on a minimal description of what is both necessary and sufficient.

EXAMPLE

The following section demonstrates how MOUSE learns a portion of the generalized pentane rule from a selected set of conformational examples. This set is shown in Figure 2. The demonstration does not use each conformation of each molecule, but uses a limited subset to illustrate the various steps in learning the generalized concept. In this example MOUSE is trying to learn a *critic*, i.e., a rule which will prove that the proposed conformation is *not* going to be stable. Thus, positive examples are conformations which were criticized and rejected as being unstable. Their descriptions are obtained from the results of WIZARD's short-term critic learning capability. The negative examples are abstract descriptions of conformations which were stable conformations and were not criticized. The descriptions were obtained from the list of noncriticized suggestions which is kept by WIZARD.

Pseudocode for MOUSE's algorithm for learning a single concept from a series of related examples can be described as follows;

```
    ($\forall$ == for all, $\in$ == an element of, $\notin$ == not an element of, & == logical and)
Start with a positive example—this becomes the working hypothesis
For $\forall$ examples$_k$
  Match example$_k$ to current hypothesis to obtain difference(s) $D_k$
  For $\forall$ $D_k$
    If example$_k$ == positive
      generalize differences                          (e.g. methyl; ethyl; propyl → alkyl)
      if $D_k \notin$ example$_k$ & $D_k \in$ hypothesis      then remove  $D_k$ from hypothesis}
    If example$_k$ == negative
      if $D_k \in$ hypothesis & $D_k \notin$ example$_k$      then update  MUST HAVE $D_i$
      if $D_k \in$ example$_k$ & $D_k \notin$ hypothesis      then update  MUST NOT HAVE $D_k$
```

The first example *must* be a positive example (i.e., a criticized conformation) to provide a working hypothesis. Thus the first example (E_1) is directly converted to the working hypothesis (H_1). This is shown schematically in Figure 3. The top circle says that we have an object called H_1. The dotted arrows indicate relations which have been "OBSERVED" but are not known to be necessary. In this case, H_1 has been observed to have four parts, labeled p1 through p4. The parts were observed to have the following adjacencies: adj(p1,p2), adj(p2,p3), and adj(p3,p4). Class membership is shown by arrows to square boxes. The abbreviations used in the diagrams are placed in square brackets in the text. Parts p1 and p4 were observed to be (isa) members of the class R_methyl [ch3], part p2 to be an isa member of 1X,2Y-ethyl_gauche- [but_gm], while part p3 is an isa member of 1X,2Y-ethyl_gauche+ [but_gp]. Solid arrows will be used to indicate relations which MUST_HAVE, and solid arrows with solid crosses will indicate

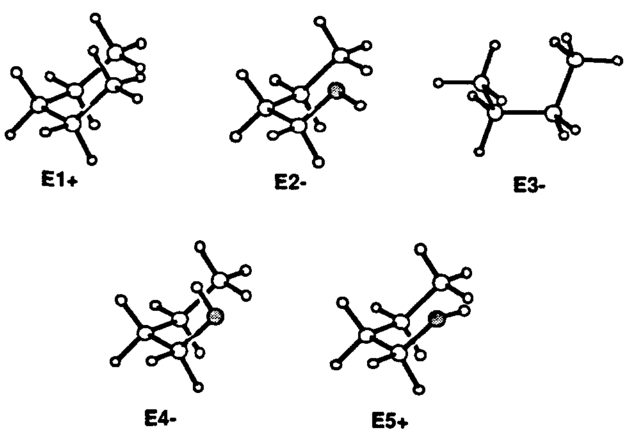

Figure 2. Limited training set used for this example.

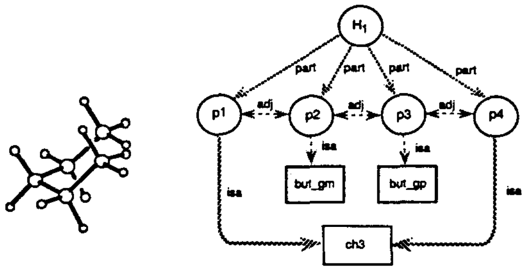

Figure 3. The first example is converted directly to the working hypothesis. Dotted lines are observed relations, which may be necessary or sufficient.

relations which MUST_NOT exist. The working hypothesis contains no such certainties.

Once the first example has been used to create the working hypothesis, the order of positive and negative examples is not important. The next example (E_2-) shown in Figure 4 is a negative example obtained from a stable conformation of *n*-butanol. The first job of the mapper is to determine the best mapping of the variables in the example (A, B, C, D) to permanent parts in the hypothesis (p1, p2, p3, p4). In all of the examples (except Figure 5) part A will map onto p1, and so forth. The description of E_2- is very similar to H_1, with two differences: The first difference "part D is R_ch2oh_anti [oh_an]" comes from the example and is missing from the hypothesis. The second difference arises because the hypothesis contains "part 4 has been observed to be R_methyl," which is missing in the example. Since this is a negative example, the updater responds to the first difference (which is part of E_2 and not H_1) by adding "part D MUST_NOT be R_ch2oh_anti" to H_1. This is shown in H_2 by the addition of a descriptor box [oh_an] with a solid cross through the arrow. The updater responds to the second difference (which is part of H_1 and not E_2) by changing the observed relationship to a MUST_HAVE relationship. This is shown in H_2 by changing the dashed arrow from p4 to the [ch3] box to a solid arrow.

The next example is a negative example (i.e., it was not criticized as containing a fatal van der Waals repulsion) and is shown in Figure 5. The mapper finds that the best mapping for example E_3- to H_2 is A:p1, B:p2, and C:p4. This gives one difference (NULL versus p3). Other mappings, such as A:p1, B:p2, and C:p3, give a greater number of differences (for example, C/methyl versus p3/butyl AND NULL versus p4). The matcher then finds that E_3- lacks the adjacency relationship between parts B and C found in H_2. MOUSE can determine that this feature is necessary (MUST_HAVE) to the concept since:

(1) E_3- is a negative example.
(2) There are differences which are found in H_2 and which are missing in E_3-.

This information is used to update H_2, and affords H_3. The adjacency relationships between p2, p3, and p4 are changed to solid arrows to indicate that these are MUST_HAVE relationships. Note that parts *must exist* if they *must be adjacent* to each other; thus, MOUSE can also infer that the existence of p2, p3, and p4 is necessary, so that these "part of" relationships become MUST_HAVE relationships as well.

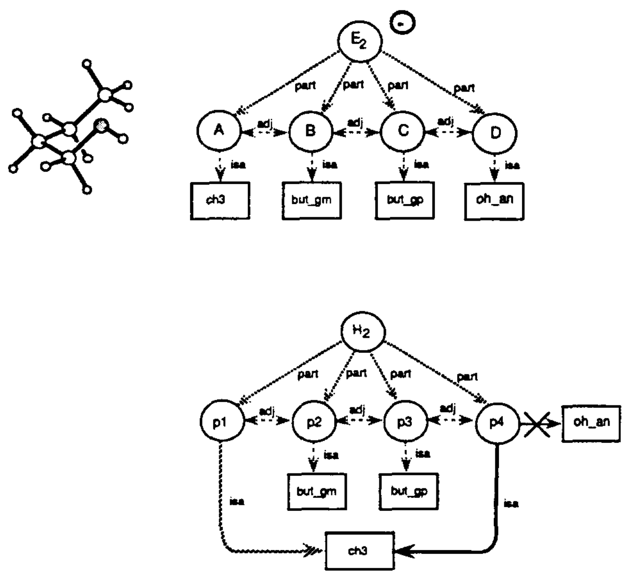

Figure 4. A negative example (E2-) is applied to the working hypothesis H1 shown in Figure 2 to obtain a new working hypothesis (H2).

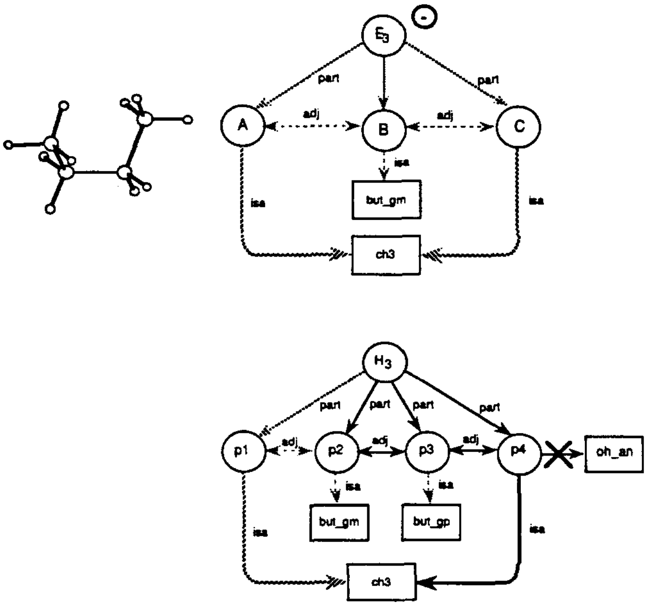

Figure 5. MOUSE determines that p3 is necessary, and that it MUST BE adjacent to p2 and p4.

Example E_4-, shown in Figure 6, is another stable conformation of *n*-butanol. The matcher finds that the gauche minus OH [oh_gm] group is the only significant difference between H_3 and E_4. This is a negative example where the significant difference is part of the example, so the updater adds that to H_3 as another constraint (MUST_NOT) to p4, and obtains hypothesis H_4. The final example E_5+, shown in Figure 7, is a criticized conformation of *n*-butanol, and therefore it is a positive example (unstable conforma-

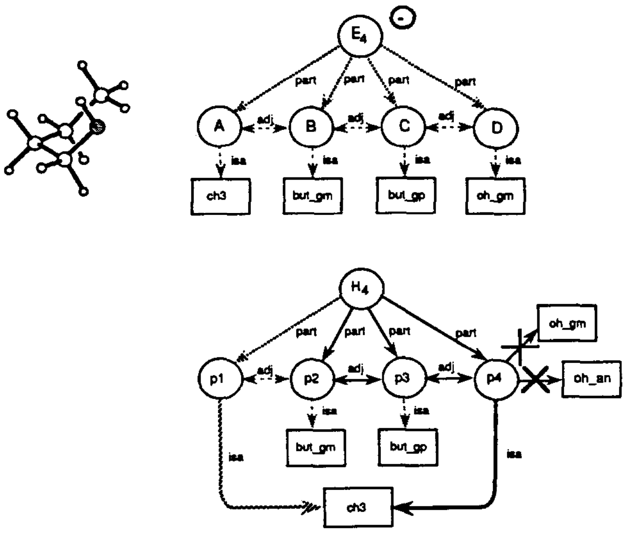

Figure 6. Example E4 is used to update the working hypothesis to obtain hypothesis H4.

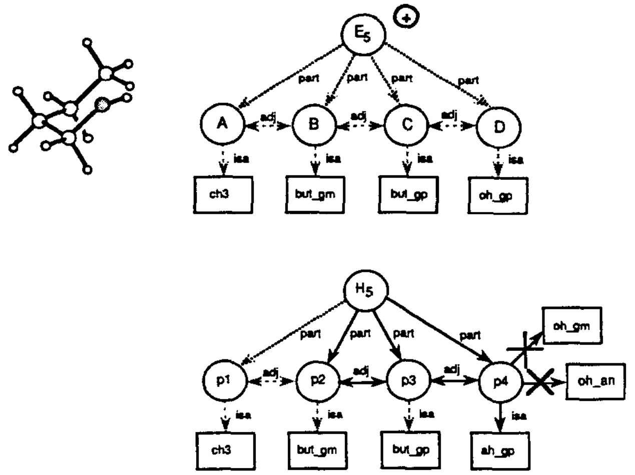

Figure 7. The operation of generalization *upon H4 and E5 gives a general class that part 4 may belong to; the class of any group that contains a hydrogen which is in the* gauche + *position.*

tion). The matcher finds that the only significant difference between H_4 and E_5+ is that "part D isa [oh_gp]" in E_5+, while "p4 isa [ch3]" in H_4. However, since E_5 is a *positive* example, the updater tries to find a general description that satisfies all of the knowledge about p4.

The generalization algorithm can be described thus:

```
If $\exists$ class C such that
  {For $\forall$ observed "part X isa rel$_k$" relationships where rel$_k \in C \wedge$
        For $\forall$ observed "part X MUST_NOT isa rel$_j$" relationships where rel$_j \notin C$}
then
  {replace $\forall$ observed "part X isa rel$_k$" relationships with "part X isa C"}
```

This states that we can replace a number of positive observations with a single class description if all of the observed objects are elements of the proposed class, and if no MUST_NOT observation is a member of the proposed class. In our example, E5+ "part D isa [oh_gp]" and H4 "p4 isa [ch3]" are both members of the class "R_anyatom_H_gp [ah_gp]." This class consists of those units which have a hydrogen attached to any atom which is further attached to a reference atom R, and the hydrogen is arranged in a gauche + fashion with regard to the reference atom. The second requirement that neither of the MUST_NOT relations "R_ch2oh_anti" or "R_ch2oh_gm" are members of the class "R_anyatom_H_gp" is also true. Thus, MOUSE can generalize the isa relationship of part D to "R_anyatom_H_gp [ah_gp]." MOUSE can not discard the MUST_NOT relationships since we have not proved that the relationship "p4 isa R_anyatom_H_gp" is both necessary and sufficient. This leads to the final hypothesis H_5.

If hypothesis H5 is examined, a mixture of OBSERVED, MUST_HAVE, and MUST_NOT relationships is found, as well as an example of generalization. MOUSE has learned that all of the positive examples of the van der Waals exclusion it has seen have contained at least 4 parts. Three of them (p2, p3, and p4) are considered to be necessary. These parts have the following properties and relationships:

- Part 1 has always been a R_methyl group, and has always been adjoining to part 2.
- Part 2 has always been a 1X,2Y_ethyl_gm group, and MUST BE adjoining part 3.
- Part 3 has always been a 1X,2Y_ethyl_gp group, and MUST BE adjoining part 4.
- Part 4 MUST BE a member of the R_anyatom_H_gp class (or at least must have some of the properties of that class), and MUST NOT be an R_oh_gm or R_oh_an.

Five carefully chosen examples were sufficient to allow MOUSE to learn quite a bit about the pentane rule. It can be seen that MOUSE could learn more details from more examples. All of the adjacency relationships could be converted to MUST_HAVE relationships by giving one more negative example that would be forced to match p2 through p4 (such as *n*-propanol). MOUSE would learn that the actual atomic identity of each atom is not important, if we provide more positive examples which contain heteroatoms (such as di-ethylether and methyl-propylether) at each heavy-atom position. This would also teach it that all but two hydrogens are superfluous for the purpose of the pentane rule critic.

MULTIPLE HYPOTHESIS

The previous example shows how MOUSE can learn a single hypothesis from a set of carefully chosen examples which all pertain to the same concept. This capability is sufficient for a *teachable program* which learns from well-chosen examples which all pertain to a single concept. But our eventual goals are to create an *apprentice program* that learns by watching an expert solve randomly chosen examples, and eventually an *explorer program* which finds gaps in its knowledge and tries to fill in the gaps on its own. In each of these cases, the program will have to work with potentially disparate and unrelated examples, and will have to build multiple independent hypotheses. The algorithm for building multiple hypotheses is shown here:

```
Start with a positive example$_1$—this becomes hypothesis$_1$
For $\forall$ examples$_k$ {
  if {$\exists$ Hypothesis$_i$ such that
      {{Match example$_k$ to hypothesis$_i$ yields a small set of differences D$_i$} $\wedge$
       {D$_i$ can be applied to update Hypothesis$_i$}}}
  then
      {update Hypothesis$_i$ with D$_i$}          update same as single hypoth. above
  else
      {if example$_k$ = = positive then example$_k$ becomes hypothesis$_{n+1}$}
}
```

The control of this algorithm rests in the underlined terms *a small set* and *can be applied*. Either of these predicates can be false if the differences are too large. This assures that MOUSE creates individual hypotheses which are internally consistent, at the possible cost of generating a number of somewhat related hypotheses. This is shown schematically in Figure 8. The first example is used to create the first working hypothesis, HA. The next example cannot be mapped onto HA with *a small set* of differences (*small set* = 1 in this example), and so a new working hypothesis HB is created. The third example shows a small set of differences to HA, and can be applied to update HA to include the term that "unit 3 MUST_NOT be anti." However, the third example is not applicable to HB. The fourth and final example shows a *small set* of differences to both HA and HB, and can be used to update each. Note how the unit information which is updated is different in HA (unit 2 MUST_NOT be *gauche–*) from HB (unit 3 MUST_NOT be *gauche*). In this example MOUSE eventually learned two versions of the pentane rule which only differed in the signs of the torsion angles (i.e., gauche + versus gauche –). Since MOUSE does not understand symmetry or have the ability to utilize *exclusive or* logic, it cannot generalize these two rules into one.

Example	HA	HB
+ Me gp gm Me	+ Me gp gm Me	
+ Me gm gp Me	not applicable	+ Me gm gp Me
- Me gp an Me	·· + u3 ≠ an	not applicable
- Me gm gm Me	·· + u2 ≠ gm	·· + u3 ≠ gm

Figure 8. A schematic example of how multiple hypotheses are generated. The examples are n-*pentane conformations, where the terminal Methyls = = Me, and the internal disubstituted ethyl groups are designated by* gp *for* gauche +, gm *for* gauche–, *and* an *for* anti.

DISCUSSION

MOUSE has shown that it is capable of learning from carefully selected examples. However, the current algorithms are not perfect. The generalization algorithm has several problems:

- The algorithm is too hasty; it can overgeneralize based on a few observations, and if a counterexample is found, another part of MOUSE has to retract this generalization. However, as long as such an "ungeneralization" capability exists, the only damage will be a degree of inefficiency. More work is needed to find the proper balance between overgeneralization and overspecification.
- MOUSE cannot select from a number of general classes—sometimes there is more than one general class which fits the observations, and MOUSE cannot determine which is most appropriate.
- MOUSE depends on premade class hierarchies; currently MOUSE depends on supplied "a kind of" relationships. An example is that the knowledge that a methyl group is "a kind of" alkyl group must be provided by a human chemist beforehand. Thus, MOUSE is limited by the types of general classes which the programmers foresee as being important. MOUSE needs a method of determining class generalizations based on physicochemical properties.

Another limitation in MOUSE arises due to the fact that the current matching and difference discovery algorithms of MOUSE are very general and thus very inefficient. A new matcher will have to be constructed to be able to handle a large number of examples with hundreds of parts and relationships. This matcher will probably benefit if it contains knowledge which will help it efficiently decide if and when it should try to match parts and properties. For example, MOUSE shouldn't try to match a phenyl ring in an example to a 1X,2Y_ethyl fragment in a hypothesis until more chemically sensible matchings (e.g., phenyl ring to pyridine) have been exhausted. However, we are concerned that this knowledge of what is chemically sensible might accidentally introduce a prejudice, and that MOUSE would miss some rules arising through unusual analogy. The best solution to this problem will have to be determined experimentally.

CONCLUSIONS

MOUSE has shown that it is possible to write a teachable program that can learn symbolic rules of conformational analysis from selected examples. The rules can be utilized as critics or axioms for conformational analysis programs such as WIZARD. These additional critics will enhance the performance of WIZARD substantially. Although WIZARD cannot use the output of MOUSE yet, a hand simulation was performed to see what sort of rate enhancement could be expected. Short-term critics which arose from the analysis of structures related to steroids (decalins and indanes) were generalized into critical rules of conformational analysis. These rules were then entered into WIZARD. Subsequent analyses of steroid structures exhibited speed increases of up to 22-fold when compared to runs where the same molecules were run without the additional critics, without any loss of predictive ability.

The construction and testing of MOUSE is the first step in creating an apprentice program which will automatically learn generalized rules from the results of WIZARD's analysis of nonselected examples, and is a step towards our ultimate goal of creating an explorer program which will devise its own problems and direct its own learning. MOUSE has revealed several problems which must be resolved before we can proceed onto the next steps, chief among them questions about the proper method for generalizing class membership, reducing redundant hypotheses, and utilizing prior knowledge without prejudicing future learning. However, even without the solutions to these problems the linkage of MOUSE and WIZARD will provide a tremendous enhancement in the abilities of a user to tailor WIZARD to unusual domains through the automated learning from selected examples and to obtain significant speed increases in more common domains. The accompanying paper on short-term learning mentioned that the techniques are general and can be utilized in distance geometry, ''template joining,'' and other programs. MOUSE is also a general program which is not tied to WIZARD's methods. If these other programs were enhanced with a short-term learning ability, then they could utilize MOUSE or the MOUSE techniques directly. Although the speed increases obtained might not be as substantial as exhibited by WIZARD (since WIZARD was designed with this capability in mind) this combination of a short-term learning facility and long-term learning should provide two enhancements for any such program: a learned decrease in the amount of time required to perform subsequent analyses, and the creation of generalized rules which could be used by future programs, chemists, and students.

ACKNOWLEDGMENTS

Acknowledgment is made to the donors of The Petroleum Research Fund, administered by the ACS, for partial support of this research. We would also like to acknowledge the support of the Stardent Corporation through the loan of a Titan-2 minisupercomputer.

REFERENCES

1 (M.E. Lacy, Ed.) Neural Networks. *Tetrahedron Comp. Method.* 1990, **3**

2 Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. *Machine Learning* Vols. 1 and 2, Morgan Kaufmann Publishers, Palo Alto, 1986

3 Dolata, D.P. and Walters, W.P. Short-Term Learning in Conformational Analysis. *J. Mol. Graphics* 1993, **11,** 112

4 Dolata, D.P. Molecular Modeling By Symbolic Logic. In: *Collected Papers from the 3'rd European Seminar on Computer Aided Molecular Design,* IBC Ltd, 1987; Leach, A.R., Dolata, D.P., and Prout, K. Automated Conformational Analysis and Structure Generation: Algorithms for Molecular Perception. *J. Chem. Inf. Comp. Sci.,* 1990, **30,** 316; Leach, A.R., Prout, C.K., and Dolata, D.P. The Application of AI to the Conformational Analysis of Strained Molecules. *J. Computational Chem.,* 1990, **11,** 680

5 Tanimoto, S.L. *The Elements of Artificial Intelligence* Computer Science Press, Rockville, 1987