PROTOS: A UNIFIED APPROACH TO CONCEPT REPRESENTATION, CLASSIFICATION AND LEARNING

ELLIS RAYMOND BAREISS JR.*

AUGUST 1988 AI88-83

* Support for this research was provided by the National Science Foundation under grant number IST-8510999 and the Army Research Office under grant number ARO DAAG29-84-K-0060.
PROTOS: A UNIFIED APPROACH TO CONCEPT REPRESENTATION, CLASSIFICATION, AND LEARNING

by

ELLIS RAYMOND BAREISS JR., B.S.

DISSERTATION
Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
August, 1988
Acknowledgments

I would like to acknowledge the support of my friends and colleagues in the Department of Computer Sciences (and elsewhere). Unfortunately, there are too many to detail their assistance over the years, so I will have to be content with merely listing their names: Agnar Aamodt, Brad Blumenthal, Karl Branting, Larry Clay, Dan Dvorak, Adam Farquhar, Rob Holte, Ben Kuipers, Vipin Kumar, James Lester, Rich Mallory, Ray Mooney, Ken Murray, Claudia Porter, Joe Ross, and Art Souther. I would also like to thank Bob Causey, Gordon Novak, Bob Simmons, and Craig Wier, the members of my committee, for their guidance in this endeavor. Finally, I would like to thank my advisor, Bruce Porter, whose unfailing support and collaboration made Protos possible.

Support for this research was provided by the National Science Foundation under grant number IST-8510999 and the Army Research Office under grant number ARO DAAG29-84-K-0060.

Ellis Raymond Bareiss Jr.

The University of Texas at Austin
August, 1988
PROTOS: A UNIFIED APPROACH TO CONCEPT REPRESENTATION, CLASSIFICATION, AND LEARNING

Publication No.

ELLIS RAYMOND BAREISS JR., Ph.D.
The University of Texas at Austin, 1988

Supervising Professor: Bruce W. Porter

The primary contribution of this research is a unified approach to concept representation, classification, and concept learning. This approach has been implemented as a computer program, Protos, which learns concepts as it performs classification under the guidance of a teacher. The soundness of the approach has been demonstrated by successfully applying Protos to the task of acquiring knowledge for performing heuristic classification at an expert level of proficiency.

The Protos approach addresses the complexities of representing, using, and learning natural concepts. These concepts are polymorphic and ill-defined. Most machine learning research is based on inductive learning and deductive classification, which are more suitable for artificial domains (e.g., mathematics) than natural domains (e.g., medicine). In contrast, Protos takes an exemplar-based approach. It represents concepts extensionally as sets of retained exemplars, classifies a new instance by recalling a similar exemplar and

vi
explaining its similarity to the instance, and learns when a classification failure indicates that knowledge is missing. Because Protos learns as a byproduct of classification, its performance continually improves.

Protos has been experimentally evaluated by training it to diagnose hearing disorders. An expert audiologist trained Protos with 200 cases of hearing disorder. Through this small amount of training, Protos evolved into an expert system whose classification performance was comparable to that of experienced human clinicians.
# Table of Contents

- Acknowledgments
- Abstract
- Table of Contents
- List of Tables
- List of Figures

## 1. Introduction

1.1 Preliminaries ............................................. 1
1.2 The Goal of this Research ................................. 2
1.3 Constraints on Concept Learning ......................... 2
    1.3.1 Representation .................................... 3
    1.3.2 Classification ................................... 3
    1.3.3 Learning ......................................... 4
1.4 Protos: A Unified Approach to Concept Representation, Classification, and Learning ............................................. 5
    1.4.1 Representation .................................... 5
    1.4.2 Classification ................................... 6
    1.4.3 Learning ......................................... 7
1.5 Evaluation of the Protos Approach ....................... 8
1.6 Introduction to Related Research ....................... 9
1.7 A Guide to this Dissertation ........................................ 11

2. An Overview of the Protos Approach .............................. 12
   2.1 Representation .................................................. 13
      2.1.1 Category Cohesiveness ................................. 14
      2.1.2 Indexing within the Category Structure ............ 15
   2.2 Classification .................................................. 17
      2.2.1 Hypothesis Formation ................................. 18
      2.2.2 Hypothesis Confirmation ......................... 20
   2.3 Learning ...................................................... 23
      2.3.1 Learning Exemplars ................................. 26
      2.3.2 Learning Explanations ............................. 26
      2.3.3 Generalizing Exemplars .......................... 27
      2.3.4 Learning Indices ................................. 29
      2.3.5 Learning through Focused Interaction .......... 33
   2.4 Summary ...................................................... 34

3. The Implementation of Protos .................................... 36
   3.1 Representation ................................................ 36
      3.1.1 The Representation of Cases ...................... 37
      3.1.2 The Representation of Explanations .............. 39
      3.1.3 Conditional Explanations ......................... 46
      3.1.4 The Representation of Indices ................. 47
   3.2 Classification ................................................ 48
      3.2.1 The Use of Indices in Classification .......... 49
      3.2.2 Knowledge-Based Pattern Matching .............. 54
5.1.4 The Assessment of Similarity ......................... 138
5.1.5 The Retention of Cases .............................. 141
5.1.6 Learning Indexing .................................. 143
5.1.7 Learning Additional Domain Knowledge ............. 145
5.1.8 Generalization During Learning ..................... 146

5.2 Automated Knowledge Acquisition .................... 149
5.2.1 Issues in Automated Knowledge Acquisition ........ 150
5.2.2 The Type of Assistance Provided ................. 151
5.2.3 Input Provided to the System ...................... 156
5.2.4 The Results of Using the System .................. 162
5.2.5 The Systems' Sources of Power ................... 165

6. Contributions and Future Work ....................... 170

6.1 Contributions ........................................ 171
6.1.1 Representation .................................... 171
6.1.2 Classification .................................... 172
6.1.3 Learning .......................................... 174
6.1.4 Experimental Evaluation of Protos ................ 175

6.2 Future Work .......................................... 177
6.2.1 Further Experimentation .......................... 177
6.2.2 New Research ..................................... 178

6.3 A Final Word .......................................... 179

A. Psychological Research on Concept Representation .... 181
A.1 The Classical View .................................. 181
A.2 The Probabilistic View ............................... 183
A.3 The Exemplar View .................................. 184
List of Tables

4.1 Characteristics of Cases Presented to Protos . . . . . . . . . . . . 112
4.2 Percentage of Correct Classifications . . . . . . . . . . . . . . . . 122
4.3 Number of Remindings Pursued to Find a Correct Match . . . . 122
4.4 Number of Matches Attempted to Find a Correct Match . . . . 123
4.5 Source of Correct Match . . . . . . . . . . . . . . . . . . . . . . 124
4.6 Average Number of Matches Discussed With Teacher . . . . . 124
4.7 Correct Classifications by Human Subjects . . . . . . . . . . . . 125
## List of Figures

2.1 A Sample Category Structure .................................................. 15
2.2 Indices Associated with the Sample Category Structure ............. 16
2.3 The Protos Classification Algorithm ....................................... 18
2.4 Learning During Problem Solving ........................................... 25
2.5 The Range of Coverage of an Exemplar ................................... 28

3.1 Correlational Explanation Links ........................................... 42
3.2 Noncorrelational Explanation Links ...................................... 43
3.3 The Protos Classification Algorithm ...................................... 50
3.4 The Protos Algorithm for Combining Reminders ......................... 51
3.5 Improving an Imperfect Match Using Difference Links ............... 54
3.6 The Knowledge-Based Pattern Matching Search Algorithm ........... 55
3.7 Learning During Problem Solving .......................................... 60

4.1 Protos’ User Interface ......................................................... 79
4.2 Features of the Example Case 1 ........................................... 81
4.3 Raw Remindings from the Features of Case 1 ........................... 82
4.4 Matching Case 1 to an Exemplar of Possible Menieres ............... 84
4.5 The Features of Example Case 2 ........................................... 93
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 Raw Remindings from the Features of Case 2</td>
<td>94</td>
</tr>
<tr>
<td>4.7 Matching Case 2 to an Exemplar of Normal_Ear</td>
<td>97</td>
</tr>
<tr>
<td>4.8 Matching Case 2 to an Exemplar of Possible_Menieres</td>
<td>100</td>
</tr>
<tr>
<td>4.9 The Features of Example Case 3</td>
<td>102</td>
</tr>
<tr>
<td>4.10 Raw Remindings from the Features of Case 3</td>
<td>103</td>
</tr>
<tr>
<td>4.11 Matching Case 3 to 1st. Exemplar of Cochlear_Poss_Noise</td>
<td>105</td>
</tr>
<tr>
<td>4.12 Matching Case 3 to 2nd. Exemplar of Cochlear_Poss_Noise</td>
<td>106</td>
</tr>
<tr>
<td>4.13 Growth of the Category Structure — Categories and Exemplars</td>
<td>113</td>
</tr>
<tr>
<td>4.14 Retention of Exemplars by Category</td>
<td>115</td>
</tr>
<tr>
<td>4.15 Distribution of Test Cases by Category</td>
<td>116</td>
</tr>
<tr>
<td>4.16 Growth of the Category Structure — Indices</td>
<td>117</td>
</tr>
<tr>
<td>4.17 Growth of the Category Structure — Explanation Links and Conditions</td>
<td>118</td>
</tr>
<tr>
<td>4.18 Teacher-Provided Explanations per Case</td>
<td>119</td>
</tr>
<tr>
<td>4.19 The Types of Explanation Links in the Category Structure</td>
<td>120</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

When these fellows are at fault, they come to me, and I generally manage to put them on the right scent. They lay all of the evidence before me, and I am generally able, by the help of my knowledge of the history of crime, to set them straight. There is a strong family resemblance about misdeeds, and if you have the details of a thousand at your finger ends, it is odd if you can't unravel the thousand and first.

— Sherlock Holmes

1.1 Preliminaries

Classification is fundamental to intelligent problem solving. By classifying a new entity (i.e., object, event, or state), a problem solver can avoid much of the effort required for processing a unique occurrence. The problem solver's past experience with similar entities can be brought to bear, and its limited computational resources conserved.

Classification is the process of recognizing an entity as an instance of a known concept. A concept is the intensional representation of a class of entities which are equivalent with respect to the problem solver's goals and experience. An entity is classified by recognizing that its attributes imply that
descriptions. Concepts have atypical instances which lack many of the features of prototypical instances. The system must be able to classify such instances as well as typical ones.

Second, the classification process must function with nonideal input. The environment often provides noisy descriptions of cases. Spurious information may be present, and important information may be missing or incorrectly observed. A classification system must be able to impose a consistent interpretation on its inputs and to deal with uncertainty in reaching its conclusions.

1.3.3 Learning

Learning must support the needs of classification by acquiring an appropriate representation for use in the classification process.\(^2\) Constraints on representation and classification are constraints on learning as well. Additional constraints are imposed by the nature of the training available in the system's environment and the opportunities for learning provided in the course of problem solving.

A learning system must be able to learn from the types and amount of training available in its environment. Input to the system is usually not ideal; it is noisy and incomplete. The learner may see few examples of important concepts. Yet, in spite of these impediments, it must learn accurately. To do so, the system must make use of other forms of training in addition to observation of concept instances, and it must synthesize this training to form accurate concepts.

\(^2\)Porter has made the more general observation that the role of learning is to acquire a representation that supports problem solving [Por84].
In many domains, experience cannot be neatly divided into learning and problem solving phases. During classification, new problems will arise necessitating new concepts. Atypical concept instances will be encountered. If learning is integrated with performing classification, the system can improve the robustness of its problem solving in ways specifically indicated by its experience.

1.4 Protos: A Unified Approach to Concept Representation, Classification, and Learning

Protos learns as a byproduct of performing classification under the guidance of a human teacher. When presented with the description of an entity (i.e., a case) to be classified, Protos attempts to recall a previous case and to explain its similarity to the new case. When it cannot correctly classify or adequately explain a case, Protos interacts with the teacher to obtain the correct classification and an explanation of why it is correct. It learns by selectively retaining cases and their explanations. This section discusses how the Protos approach satisfies the constraints on representation, classification, and learning introduced in the previous section.

1.4.1 Representation

Protos captures the polymorphism of natural concepts by representing them extensionally. The system does not attempt to compile its experience by inducing concept definitions. Rather, a category (the extensional equivalent of a concept) is represented by a set of retained cases called exemplars. As it learns, Protos decides which cases to retain to represent each category adequately; the exemplars of a category that are retained are determined by its observed degree of polymorphism.
Protos captures the *cohesiveness* of categories by retaining explanations. A simple collection of exemplars does not adequately represent the intension of a category. Its members must be equivalent with respect to the system's problem solving goal. This equivalence is explicitly represented by explanations of how exemplar features are relevant to category membership *(cf., [MM85])*.

### 1.4.2 Classification

Protos classifies a new case by retrieving an exemplar to serve as a model for interpreting the features of the new case. The retrieved exemplar model provides expectations as to which features the new case should possess and suggests the importance of fulfilling them. This model-based approach aids Protos in dealing with nonideal descriptions by allowing it to concentrate its effort on establishing the presence of important features while ignoring features which are likely to be spurious. Because categories are generally represented by many exemplars, a range of models is available to facilitate the classification of atypical as well as typical cases.

Because great variation is possible in case descriptions, Protos uses explanation to aid in classification. Equivalent, but syntactically different, descriptions may be provided. Important attributes may be missing in descriptions. Protos uses domain knowledge to construct explanations which allow it to infer the similarity of features or the presence of criterial attributes *(cf., converting representations in [FD86])*.
1.4.3 Learning

In Protos, learning improves the accuracy of classification. Protos learns from failures to classify cases and from failures to explain its classifications. A failure identifies a gap in Protos’ knowledge which is filled by retaining the case and a teacher-provided explanation of its classification. Knowledge acquired in response to a problem solving failure prevents the occurrence of similar failures in the future.

Learning also improves the efficiency of classification. A concern about an exemplar-based approach is that inefficient classification may result from the amount of information retained by the system. Protos maintains efficiency by learning an indexing structure as it acquires exemplars and explanations. It learns associations between features and categories, which restrict its search for classifications. Within each category, it learns the prototypicality of exemplars, which suggests the exemplars that are most likely to match new category members. Protos also learns how atypical cases may be misclassified, which reduces the likelihood of recurrent misclassifications.

The Protos approach of learning from explained cases facilitates the teacher’s role in two ways. First, cases need not be uniformly described because Protos represents concepts extensionally rather than attempting to induce general definitions from featural similarities. Second, few training cases are required because the explanation associated with each exemplar allows inferential matching, increasing the range of cases to which it can be matched during classification.

Because it integrates learning with performing classification, Protos is able to behave reasonably when classification fails. It does not require that its experience be divided into distinct learning and problem solving phases.
Rather, it learns any time when a classification or explanation failure occurs.

1.5 Evaluation of the Protos Approach

A new approach to concept representation, classification, and learning cannot be accepted without evidence of its validity. In our research paradigm, such evidence is provided by experimentation. A theory is evaluated by implementing it as a computer program, applying the program to a suitable task, and evaluating the results.

Protos has been experimentally evaluated by applying it to the task of knowledge acquisition for expert, heuristic classification. Clancey has noted that heuristic classification is the fundamental problem solving method of current expert systems [Cla85]. It is a method for performing classification under conditions of uncertainty. In many domains of expertise, classifications are uncertain because the relationships between the features of cases and their classifications are poorly understood. Recognition of these relationships is based on problem solving experience rather than exact, detailed knowledge of the underlying mechanisms of the domain. The goal of heuristic classification is generally to infer an unobservable state, such as a disease, on the basis of observable features, such as symptoms.

In contrast with an everyday task, such as the common sense classification of animals, an expert classification task provides objective standards for the evaluation of concept acquisition. A system's learning performance is comparable to that of human students learning the same task. After it has learned, the system's classification performance is comparable to that of human journeymen and experts in the domain.

Protos has been evaluated in the domain of clinical audiology; it has
learned to classify hearing disorders from featural descriptions. Audiology was chosen for three reasons: First, it is representative of heuristic classification tasks. Patients are assigned to known diagnostic categories on the basis of features such as symptoms and test results. Second, an interested expert was available to serve as Protos' teacher. Third, a graduate program in audiology at the University of Texas offered a population of expert and student diagnosticians against whom Protos could be compared.

1.6 Introduction to Related Research

Protos addresses a broad range of issues involving the representation, use, and acquisition of concepts. Given this breadth, there is a large body of previous research which is relevant to the design of Protos and to the understanding of its design. However, rather than presenting a series of superficial summaries and comparisons, this discussion will focus on two areas of research: case-based reasoning and automated knowledge acquisition. These areas are most directly related to Protos, and in-depth discussion of the issues they involve will facilitate an understanding Protos' unique contributions.

Research in case-based reasoning encompasses a class of problem solving techniques which make use of specific past experience rather than general knowledge. A new problem is solved by recognizing its similarity to a specific known problem and transferring the known solution to the new problem. Case-based reasoning has been applied to many tasks including classification [KD87, KK85], planning [Kol87, Ham86, Sim85], legal reasoning [RA87, Bai86], question answering [Kol83a, Kol83b], speech recognition [Bra87], and pronunciation [Leh87, SW86].

Protos is architecturally similar to many case-based reasoning sys-
tems. Research in this area shares the problems of locating cases in the system's knowledge base, assessing their similarity to a new case, and learning in the course of problem solving. Protos implements novel solutions to these problems based on its analysis and reuse of teacher-provided explanations.

Research in automated knowledge acquisition is pragmatically motivated by the goal of efficiently transferring knowledge from a human expert to a knowledge-based system. Since classification is the primary function of most knowledge-based systems, research in automated knowledge acquisition largely addresses concept acquisition. Many approaches have been studied including systematic elicitation [Ben85, Boo84], learning from examples [Qui83, MC80], focused discussion of problem solving [Dav79], heuristic analysis of existing knowledge [KNM85, EEMT87], and compilation of existing knowledge [MMS85].

Protos' concept learning task is similar to the tasks performed by most automated knowledge acquisition systems. Previous research efforts can be characterized with respect to two factors. First is the degree of autonomy exhibited in knowledge acquisition. Second is the phase of the knowledge acquisition process for which assistance is provided. Protos is unique in that it combines autonomous learning with focused discussion of problem solving. It is thus able to provide a broad range of assistance in the acquisition of domain knowledge and its refinement during problem solving.

In addition to these discussions, the psychological literature on concept representation that influenced the design of Protos is surveyed in Appendix A.
1.7 A Guide to this Dissertation

The body of this dissertation provides the rationale for the design of Protos, explains the details of its construction, and evaluates its performance. Chapter 2 discusses how the Protos approach meets the requirements of representing concepts, using them for classification, and acquiring them from available training. Chapter 3 discusses the implementation of Protos. Chapter 4 provides examples of the application of Protos to audiology and evaluates its performance. Chapter 5 surveys related work in the areas of case-based reasoning and automated knowledge acquisition. Finally, Chapter 6 discusses the contributions of Protos and suggests future work.

Before proceeding, a note is in order regarding the examples used in this dissertation. To ensure that the ideas underlying Protos are presented in the most understandable form, examples are generally drawn from everyday concepts such as animals and furniture. Examples of the application of Protos to audiology are restricted to Chapter 4.
Chapter 2

An Overview of the Protos Approach

Imagine that Protos and its teacher have been discussing animals. After a few cases, the following idealized discussion would occur:

Teacher: Protos, a new case has the features wet nose, floppy ears, four legs, medium length fur, brown color, tail, medium weight, it barks, and has a collar.

Protos: Based on this description, I am most strongly reminded of the category dogs. I will retrieve a past example, dog017, and will try to explain the similarity of the two cases by comparing their features. Many of the features of the new case match dog017 directly, but there are some I cannot match. First, the color of dog017 is brindle, not brown.

Teacher: If an animal is a dog, brown and brindle are both acceptable colors.

Protos: Second, dog017 does not have a collar.

Teacher: That is true. However, a collar enables a person to control a dog so it is a relevant feature to dogs in general.

Protos: I now believe that having a collar is a moderately important feature for dogs. In future cases, if an animal has a collar, I will be reminded of dogs.

Teacher: O.K.
Protos: Since this case differs from dog017 in several respects, I am going to retain it as a new exemplar of dogs. Is this a satisfactory disposition of the case?

Teacher: Yes.

This dialogue illustrates Protos' approaches to representation, classification, and learning. Protos represents the concept dog with a set of exemplars of dogs. It classifies a new case by recalling an appropriate example and attempting to explain how they are similar. It learns by retaining exemplars and explanations.

This chapter provides an overview of how Protos addresses the interrelated requirements of representation, classification, and learning which were introduced in the first chapter. Details of how Protos is implemented are postponed until the next chapter.

2.1 Representation

Protos is able to capture the polymorphism of natural and goal-directed concepts because its concept representation is extensional. A category (the extension of a concept) is represented by a retained subset of its members. These exemplars have been chosen by the system to represent the variability of relevant features found in the entire range of category members. For example, trees might be represented by retaining the descriptions of an oak, a pine,

---

1A brief survey of the psychological literature on human concept representation is presented in Appendix A. Review of this research greatly influenced the design of Protos' concept representation.

2A glossary of Protos terminology is included at the end of this dissertation.
a *mesquite, et cetera*. No attempt is made to induce an intensional definition of the category; the definition remains implicit in the collection of exemplars.

Since no attempt is made to induce a category-wide definition, information which might prove useful in future classification is not abstracted away. An atypical instance of a category is easier to classify by comparison to a similar instance than by deductive reasoning from a general definition. For example, a *platypus* might be difficult to classify by reasoning from a general definition of *mammal*.

The choice of an exemplar-based representation raises two issues: representation of the cohesiveness of categories and representation of indexing information to facilitate classification.

### 2.1.1 Category Cohesiveness

A simple collection of exemplars does not adequately represent the intension of a category. Category cohesiveness requires that the underlying commonality of category members be explicit in the representation [MM85]. Therefore, a category in Protos is more than simply a collection of exemplars, represented by sets of features. Each exemplar feature has an associated *explanation* (i.e., domain knowledge) which explains its relevance to category membership. For example, in Figure 2.1 *pedestal* is relevant to *chairs* because “*pedestal* is a specialization of seat support which enables *holds*(person) which is the function of *chairs*.” Domain knowledge also contributes to category cohesiveness by explaining equivalences among the features of the exemplars of a category.³

---

³An explanation of featural equivalence is an explanation that, because of underlying domain knowledge, two features provide a similar type of evidence for a classification. Such
Figure 2.1: A Sample Category Structure

For example, *pedestal* is equivalent to *legs(4)* because both are seat supports. This network of domain knowledge and exemplars is called Protos' *category structure*.

### 2.1.2 Indexing within the Category Structure

Four types of indices enable efficient access to Protos' category structure: reminding, censors, prototypicality, and difference links (shown in Fig-
Figure 2.2: Indices Associated with the Sample Category Structure

These indices contribute in different ways to Protos' choice of an appropriate exemplar with which to compare a new case. Since these indices are discussed at great length in Section 2.2 on classification and Subsection 2.3.4 on learning indices, they will only be briefly described here.

Reminders (cf. [Sch82,Ros78]) associate features with categories or particular exemplars. Such associations provide Protos with hypotheses during classification, which restrict its search for a matching exemplar. For example, seat reminds Protos of chairs in Figure 2.2.

Censors (cf. [Win86]) are negative remindings. They are negative associations between features and categories which note causes of incorrect classifications. For example, fever would be a censor for the category healthy patient (No censors are shown in Figure 2.2).

Prototypicality ratings provide a partial ordering on exemplars within a category. Exemplars of a category which have the highest family resemblance (i.e., are most similar to other members of the category) are most prototypical
[RM75]. During classification, Protos uses prototypicality ratings to determine the exemplars of a category which are most likely to match a new instance. For example, in Figure 2.2, chair1 is considered to be more prototypical than chair2 based on Protos’ experience.

Difference links note near misses (cf. [Win75]) which occurred during previous problem solving episodes. They record important featural differences between exemplars, which may suggest alternate classifications and better exemplars for use during classification. For example, in Figure 2.2, legs(4) is a relevant difference between chair2 and chair1.

2.2 Classification

Protos classifies a new case by using an exemplar, which has some similarity to the new case, as a model for interpreting its features. The exemplar model provides expectations of the features the new case should possess and suggests the importance of fulfilling them. For example, chair1 in Figure 2.1 suggests that a chair should have a backrest, a seat, a pedestal, and armrests; it may also have wheels, but they are believed to be spurious to the classification. Because a category is generally represented by several exemplars, a range of models is available to classify atypical as well as prototypical cases. The model-based approach also allows Protos to deal with nonideal case descriptions by concentrating its effort on establishing the presence of important features while ignoring features that are likely to be spurious.

Classification is a two-step process. The first step is to form a hypothesis as to the classification of a new case. The second step is to confirm the hypothesis by selecting an exemplar of the hypothesized classification and constructing an explanation of the equivalence of the features of the new case
Input a case to be classified
Retrieve reminders based on the features of the case
   and heuristically combine them \{hypothesis formation\}

\textbf{REPEAT}
\textbf{REPEAT}
3 Use the strongest combined reminding to select an
   exemplar from the category structure
4 Evaluate the similarity of the new case to the
   exemplar by constructing an explanation of their
   similarity \{hypothesis confirmation\}
\textbf{UNTIL} an adequately explained match is found
5 Use difference links to improve the classification by
   considering neighboring exemplars as alternatives
6 Present the match and explanation to the teacher
   IF the teacher disapproves
      THEN \{Learn from the failure\}
         Request new domain knowledge from the teacher
8 Reassess the reminders which suggested the
   exemplar
\textbf{UNTIL} the teacher approves

Figure 2.3: The Protos Classification Algorithm

to those of the selected exemplar. An algorithmic description of how Protos
performs classification is presented in Figure 2.3.

2.2.1 Hypothesis Formation

Protos hypothesizes classifications by recalling categories of which it is
reminded by the features of the new case (Step 2 of the classification algorithm
in Figure 2.3). These categories are likely to contain similar exemplars which
can be used to confirm the hypothesized classification. The process employs two
types of indices introduced in the previous section: reminders and censors.

The reminding mechanism relies on associations between the features of the new case and categories. Each reminding (cf. [Sch82,Kol83a]) is an indexing link in the category structure relating a feature to a category or to a particular exemplar. A reminding suggests that the new case is a member of a category because it possesses a feature generally associated with category members or uniquely associated with one particular exemplar (cf. cue validity in [Ros78,Med83]). Reminders vary in strength according to the quality of teacher-provided explanations relating various features and categories encountered in past training.

The features of a new case typically provide many reminders, which are heuristically combined to yield a few hypothesized classifications for the case. Multiple reminders to the same category increase the system's belief in its relevance. For example, reminders associated with seat, backrest, and legs(4) combine to remind Protos strongly of chairs. Also, reminding strength is inherited taxonomically so Protos will pursue more specific reminders first. For example, if Protos is reminded of both chairs and furniture, chairs will inherit the strength of the combined reminders to furniture.

A censor is a negative association between a feature and a category such that if the feature is present in a new case, it tends to disconfirm the category as a hypothesis. Censors may offer a degree of disconfirmation ranging from mild to absolute. For example, a severely abnormal hearing test result would absolutely censor the hypothesis that a patient's hearing is normal.

When the reminders associated with the features of a new case are

---

4The acquisition of indices is discussed in Section 2.3.4.
heuristically combined to formulate hypotheses, any censors associated with case features are also taken into account. If a censor suggests absolute disconfirmation, the hypothesized category is excluded from further consideration. However, if a censor provides weaker disconfirmation, its negative strength serves only to weaken the hypothesis. Weakening affects the order in which hypotheses will be tried but does not preclude their consideration.

2.2.2 Hypothesis Confirmation

After Protos has hypothesized a classification, an exemplar is selected and used to confirm the classification (Step 3 in Figure 2.3). If Protos is not reminded of a particular exemplar of the hypothesized category, it relies on prototypicality ratings to choose an exemplar based on expectations of similarity. A prototypical exemplar is a priori more likely to be similar to a new category member.

The exemplar serves as a model for interpreting the new case. (cf. [NFAR82,WCA78]). It provides expectations as to which features a new case should possess and suggests the effort that should be put into confirming their presence. The similarity between the new case and the exemplar model is gauged by constructing explanations of the equivalence of their features by a process of knowledge-based pattern matching (Step 4 in Figure 2.3).

Knowledge-based pattern matching uses domain knowledge within the category structure to construct explanations relating the features of the new case to those of the exemplar. Such an explanation allows Protos to infer the equivalence of a case feature to an exemplar feature. For example, Protos might match the feature $legs(4)$ of a new case to $pedestal$ in the exemplar $chair1$ (of Figure 2.1) based on the explanation "$legs(4)$ is a specialization of $seat\ support$
which has another specialization *pedestal*.” This suggests that the features are equivalent because they have the common generalization *seat support*.

An explanation is a sequence of domain terms linked by relations from Protos’ *explanation language*. The terms are provided by Protos’ teacher during discussion of classification failures. The relations are predefined. They are discussed further in Section 3.1.2 and are enumerated in Appendix C. The relations fall into the categories:

- correlational
- definitional
- circumstantial-to-inferred evidence
- part-to-whole
- functional
- causal
- generalization/specialization
- mutual exclusion

Knowledge-based pattern matching is a heuristic search procedure. It searches for the strongest chain of inferences (i.e., known relations) in the category structure linking each feature of the exemplar model to a feature of the new case. A chain of inferences does not constitute a proof of equivalence but rather is the most plausible argument that can be made, based on existing knowledge.

Not all chains of inferences constitute valid explanations. When searching for an explanation, Protos uses heuristic guidance to choose the best step at each point in the process. Heuristic rules (enumerated in Appendix D)
evaluate alternatives in the context of the goal of the explanation and the partial explanation that has already been formulated. These heuristics inhibit and prime relations at each step of the process making various types of inferences more or less reasonable, depending on context.

Classification is not an all-or-nothing procedure. The similarity between a new case and the recalled exemplar model is determined by evaluating the quality of the explanations relating them. Evaluation depends on the quality of the inference chains relating case and exemplar features and on the believed importances of exemplar features which are not matched. The unmatched features of the new case are considered to be irrelevant (e.g., noise).

The assessment of similarity is, in essence, probabilistic. The notion of probability in Protos is however more closely related to the original notion of probability than to its current mathematical conception [CDD*85] — something is considered probable when there is a good argument or authority for it. In this sense, Protos uses a metric very similar to Cohen's theory of endorsements [Coh85].

If the match between a new case and a recalled exemplar is imperfect, Protos attempts to improve the match by examining difference links associated with the exemplar (Step 5 in Figure 2.3). Difference links record important featural differences between pairs of neighboring exemplars within the category structure (cf. [Kol83a]). They suggest alternative hypotheses or more similar exemplars within the current category. After an explanation of the similarity of the new case to the retrieved exemplar has been formulated, Protos notes features which remain unmatched. It attempts to locate a more similar exemplar by considering neighbors of the current exemplar, which may match the new case more closely. Difference links from the exemplar, which are associated
with unmatched features of the new case, are traversed to attempt to locate a more similar exemplar. For example, given the difference links in Figure 2.2, a new case that partially matched chair2 but had the additional feature arm rests, would cause Protos to traverse the difference link and to try to match the new case to chair1.

When knowledge-based pattern matching does not lead to an adequate explanation of the similarity between the new case and the chosen exemplar, Protos reconsiders its hypothesis. Protos is not restricted to comparing a new case to only the most prototypical exemplar of a category. Rather, the strength of the combined reminding to a category indicates the degree to which that classification is worth pursuing. A strong combined reminding can lead Protos to consider several exemplars (ranked by prototypicality) while a weaker one can lead it to consider only a single exemplar. It does, however, perform satisficing search [Sim81]. As soon as an exemplar is found which is adequately similar to the new case, the result is reported to the user. Additional exemplars (of the same or other categories) are considered only if the user requests this action.

When an adequately explained match is found, the evidence for the classification of the new case is presented to the user (Step 6 in Figure 2.3). The system does not simply report that the new case is a member of category \( X \) with a probability of 50%. Rather, it explains in symbolic terms why it believes the new case should be so classified.

2.3 Learning

Protos is a learning apprentice. A learning apprentice is a knowledge-based system that provides interactive aid in solving some problem and that
acquires new domain knowledge by generalizing from training examples acquired through the normal course of its use [MMS85]. Protos learns as a byproduct of attempting to perform classification under a teacher's guidance. The teacher presents a new case for classification. Protos attempts to classify the case and to explain its classification. If the classification is incorrect or the explanation is inadequate, the teacher is asked to supply additional information.

Protos learns from cases and explanations. Such training is natural for the teacher to provide. The teacher is never asked to communicate knowledge outside of the context of a particular case. Protos takes the initiative to acquire knowledge to prevent the recurrence of problem solving failures which occur in the course of classification. It retains exemplars and asks questions to elicit explanations from the teacher. These explanations are analyzed to extend the system's knowledge and to revise existing knowledge.

Because it learns from explanations, Protos is able to learn more efficiently than concept acquisition systems that only learn from examples. The explanation, which accompanies a retained exemplar, increases its usefulness by enabling knowledge-based pattern matching of its features. Fewer exemplars are needed to learn a (polymorphic) category because each exemplar implicitly covers a range of featural variations. Explanations (and other information gleaned from problem solving experience) also enable effective learning of the indices which allow efficient access to Protos' category structure.

Protos' general learning algorithm is presented in Figure 2.4 and is discussed in the following subsections.
<Search for a matching exemplar and present it to the teacher>

IF the teacher disapproves
1. THEN Request new domain knowledge from the teacher
2. Reassess the reminders which suggested the exemplar
3. Learn censors
   <try again>
ELSE {the classification and explanation are correct}
   IF the match is imperfect
4. THEN request explanations of featural equivalence
   IF the match is very strong
5. THEN Merge the case with the exemplar
6. Increase prototypicality of exemplar
ELSE Add the case to the category structure as a new exemplar
8. Learn new reminders
   IF the case was initially misclassified
9. THEN Install difference links to the exemplars involved

Figure 2.4: Learning During Problem Solving
2.3.1 Learning Exemplars

The bulk of Protos' knowledge is the collection of cases, which are retained as exemplars of the categories learned by the system. Protos builds its category structure by selective retention of the cases presented to it by the teacher. It attempts to retain only the cases that are required to represent the significant variability that it has encountered in descriptions of category members. A new case becomes an exemplar only when Protos cannot classify it, or its similarity to an existing category member cannot be adequately explained (Step 7 of the learning algorithm in Figure 2.4).

2.3.2 Learning Explanations

Protos acquires explanations from the teacher in two situations: when a correct, but imperfect, match is discussed and when a new exemplar is added to the category structure. When a new case is correctly matched to an exemplar, Protos expects to find a match for each nonspurious feature of the exemplar. It discusses each unmatched feature with the teacher, who is asked to identify an equivalent feature in the new case and to provide an explanation of its equivalence to the exemplar feature. If the teacher can do so for the majority of important exemplar features, the new case is not retained as an exemplar.

When a new exemplar is added to the category structure, Protos expects to be able to formulate an explanation of the relevance of each of its features to the category (i.e., classification). Protos invokes knowledge-based pattern matching to search for a chain of plausible inferences linking each feature to the category. If it fails to find a reasonably strong chain, the teacher is asked for assistance. Protos requests an explanation (in its predefined
explanation language) of why the presence of the feature increases the belief that the classification is correct. This explanation is analyzed to compile a reminding and to estimate the importance of the feature to the classification.

In both situations, Protos retains the teacher's explanations. New nodes are created in the category structure to correspond to new terms in an explanation. They are linked to each other and to existing nodes by the relations used in the explanation.

2.3.3 Generalizing Exemplars

Protos does not retain all of the cases which are presented to it. When a new case is closely matched by an existing exemplar, there is no need to retain the case. Any future case which would match the new case should also be adequately matched by the exemplar. Therefore the case is discarded, and the prototypicality of the exemplar is increased to reflect the fact that it closely matched another instance of the category (Steps 5 and 6 in Figure 2.4).

Each exemplar can be considered to be the center of a region within the space of its category. This region is called the range of coverage of the exemplar. The range of coverage is determined by Protos' domain knowledge. As Protos gains domain knowledge about the commonly occurring features of a category (Step 4 in Figure 2.4), its ability to use knowledge-based pattern matching to explain featural equivalences improves. The first instance of a category can match only new cases whose features are identical. As the teacher addresses problem solving failures by providing appropriate explanations, Protos is able to match equivalent features at increasing transformational distances. The range of coverage of an exemplar changes dynamically as domain knowl-
edge is gained; the exemplar is implicitly generalized.\textsuperscript{5} Figure 2.5 graphically represents this behavior.

An exemplar is also generalized when Protos estimates the importances of its features to category membership (during step 7 of Figure 2.4). If a feature is believed to be unimportant to category membership, its absence in a new case does not affect the assessment of the similarity of the case to an exemplar.

When a new feature is encountered, Protos' initial assessment of its importance is analytical. Protos formulates an explanation of the certainty with which the presence of the feature can be inferred from category membership. Heuristic evaluation of the explanation allows Protos to assign a qualitative importance to the feature, ranging from spurious to necessary. For example, the explanation "\textit{dog} sometimes exhibits \textit{color} (\textit{brown})" suggests that the feature \textit{color} (\textit{brown}) is of low importance to the category \textit{dog}.

Sometimes, Protos' estimates of importance are incorrect. When incorrect estimates might be contributing to an incorrect match, they are revised

\textsuperscript{5}An exemplar may also be specialized when an incorrect or overly general explanation is revised through discussion with the teacher.
by asking the teacher to provide better qualitative estimates (Step 1 of Figure 2.4). This approach was chosen over an empirically-based approach to reassessment in the belief that Protos would not see enough instances of possible category features to converge upon accurate empirical estimates of their importances.

Evaluation of explanations of featural equivalence can also cause explicit generalization of an exemplar’s features (during Step 5 of Figure 2.4). During discussion of a match, the teacher may provide an explanation of the equivalence of two features that appeals to a common generalization, function, or cause. Such an explanation is a fragment of a generalization, functional, or causal hierarchy. In such situations, Protos conjectures that it can “climb the hierarchy” and replace the existing exemplar feature with the common parent of the two features (subject to the teacher’s approval). For example, the teacher might provide the explanation “wood has generalization rigid material has specialization metal.” If metal is the exemplar feature, Protos would replace it with the generalization rigid material. This action is intended to make similar featural matches easier to achieve in future cases by compiling a portion of the explanation of featural equivalence into the exemplar. The exemplar’s original feature is retained in the system’s domain knowledge.

2.3.4 Learning Indices

The efficient use of exemplars during classification requires that they be indexed so that they can be efficiently retrieved when they are likely to be similar to a new case. The previous sections introduced four types of indices — remindings, censors, prototypicality and difference links. One of Protos’ primary learning tasks is the acquisition and revision of these indices as new
categories and exemplars are added to the category structure.

Remindings are heuristic estimates of the likelihood of particular exemplars or categories being relevant to the processing of a new case. Initially, remindings are learned by analyzing the teacher-provided explanations relating the features of a new exemplar to its classification (Step 8 of Figure 2.4). Each teacher-provided explanation is heuristically analyzed to determine if it should be compiled into a reminding. The first criterion is that all of the explanation's individual inferences must be of reasonable certainty. The second criterion is that the explanation does not contain an inference step that suggests the relationship is generally applicable to several categories. In such cases, the explanation may be truncated at that step and a reminding compiled from the remaining prefix.

A few examples of the process will make it clearer. The explanation "seat enables holding(person) which is the function of chairs" suggests that seat should remind Protos of chairs. This functional explanation is composed of uniformly strong relationships. In contrast, a weak explanation such as "color(brown) is occasionally consistent with chairs" does not suggest to Protos that color(brown) should remind it of chairs. If Protos is told "body covering(fur) is usually required by mammals which has specialization dogs" it will not compile a reminding to dogs from body covering(fur) because the specialization relation between mammals and dogs suggests that the feature may not be specific to dogs. However, because the relationship between body covering(fur) and mammals is strong, Protos will compile this prefix of the explanation into a reminding.

The validity of remindings is tested during classification. When a reminding provides a hypothesis that is incorrect or cannot be confirmed, Pro-
tos reassesses the reminding with respect to the current state of its domain knowledge (Step 2 of Figure 2.4).

The first step in reassessment is to generate the current best explanation linking the feature involved in the reminding to the classification. Because domain knowledge is acquired and revised incrementally, the chain of inferences underlying the reminding may have changed (or perhaps even been invalidated) since the reminding was compiled. If the newly generated explanation differs in strength from the reminding, Protos will recompile or remove the reminding.

The second step is to check the reminding for compatibility with other reminders associated with the feature. If the feature reminds Protos strongly of multiple categories, which are known to be incompatible, the current reminding is weakened on the assumption that it is responsible for the inconsistency. For example, if the current reminding associates *engine* with *lawnmower* and *engine* also reminds Protos of *car*, the reminding to *lawnmower* will be weakened.

If multiple, compatible reminders exist, they may be rearranged. If a feature reminds Protos of the majority of subcategories of a general category, the reminding is moved to the general category. For example, if *tail* reminds Protos of *dog*, *cat*, and *monkey*, the reminding will be moved to *mammal*. Correspondingly, a reminding shared by a general category and only one of its children is removed from the general category, in effect, specializing it to the child.

If these steps do not result in a change to the reminding, its strength is decreased to reflect diminished confidence. A reminding which continues to be weakened over time is eventually removed. The weakening process, however, is gradual enough to minimize the effects of faulty blame assignment for incorrect
hypotheses.

Censors are learned as a result of incorrect matches (Step 3 of Figure 2.4). Protos learns censors by being told. When a new case is incorrectly classified as an instance of a category, Protos asks the user if the case’s unmatched features offer disconfirmation of the hypothesized classification. If the user identifies such features, he is asked about the degree of disconfirmation that each provides. This information is then associated with the category under discussion.

Protos heuristically estimates the prototypicality of exemplars by rewarding their successful use as models for classification (Step 6 of Figure 2.4). The exemplar is rewarded in proportion to its degree of match to a new case. The more new cases that are successfully matched by an exemplar, the higher its prototypicality rating becomes. During classification, these prototypicality ratings allow Protos to identify the exemplars which have been used successfully in classifying previous cases.

A difference link is learned when a near miss is encountered during problem solving (Step 9 of Figure 2.4). A difference link is a pair of pointers between two exemplars which are annotated with important featural differences between them. Intracategory difference links are installed when a new exemplar is added to a category as the result of an imperfect, but correct, match to an existing exemplar. Intercategory difference links are installed between exemplars of different categories as a result of an incorrect classification.

Protos determines the annotation of a difference link by asking the teacher. An empirical alternative would be to simply include all unmatched features on a difference link and to incrementally remove those which did not prove to be relevant over time. However, based on the estimated amount of
training that Protos would receive, accurate convergence on an appropriate set of differences was deemed unlikely.

2.3.5 Learning through Focused Interaction

Protos acquires much knowledge from its teacher via focused interaction. It attempts to recognize situations in which specific questions about the domain are relevant, so the teacher is likely to give appropriate, specific answers. In some instances, it asks for explanations which are analyzed to compile remindications and to estimate importances. In others, Protos asks the teacher to analyze the situation and to give it the needed information directly, such as censors and difference links.

The question has been asked, “Why doesn’t Protos use empirical or similarity-based learning techniques instead of asking the teacher directly?” Protos’ primary objective is the efficient acquisition of knowledge from a human teacher. The large amount of training necessary to employ empirical or similarity-based learning techniques is often not available in the expert domains in which Protos is intended to be applied. Often, important categories are rare and few instances are available. However, they must be learned accurately. Also, it is demanding for a human expert to prepare and present a large amount of training.

Some learning is beyond the scope of limited knowledge and general heuristic techniques. For example, when providing Protos with censors, the teacher may choose features of the new case which are matched by features of the exemplar. This often occurs when the exemplar is an outlier of the category; some of its features may well be weakly incompatible with category membership. The choice of features to annotate intercategory difference links
poses similar problems. It is difficult to see how a general learning algorithm could perform satisfactorily in such situations without a large amount of pre-existing common sense and domain knowledge.

Although Protos learns from failure, it does not have the general knowledge necessary to differentiate the possible kinds of failures. A plausible misclassification is qualitatively different from a gross error. Likewise, a weak explanation is different from an incorrect one. In such situations, Protos must rely on the teacher's superior knowledge to avoid the possibility of being tricked into drastic action when none is needed.

The Protos approach attempts to expand the bandwidth of communication between the teacher and learning system to reduce the amount of training required for concept acquisition. Protos autonomously learns the things that it can reasonably be expected to learn. It depends upon interaction with the teacher to acquire knowledge which is outside of the scope of its abilities.

2.4 Summary

This chapter has provided an overview of how Protos represents concepts, performs classification, and learns. Protos represents concepts extensionally as collections of exemplars (i.e., as categories). Accompanying domain knowledge provides coherence within these categories. Classification is a two-step process. The features of a new case provide hypotheses as to its classification. Protos then seeks to verify a hypothesis by retrieving an appropriate exemplar and constructing an explanation of its similarity to the new case. When it cannot classify a new case or cannot adequately explain a classification, Protos learns by interacting with its teacher. It selectively retains exemplars and explanations and formulates indices which allow efficient
retrieval during future problem solving.

The next chapter discusses, in detail, how these mechanisms are implemented.
Chapter 3

The Implementation of Protos

This chapter sets forth the implementation of Protos, explaining how the Protos approach has been instantiated as a concept representation and a set of algorithms. The chapter elaborates and follows the organization of Chapter 2. First, the representation of Protos’ category structure is explained. The featural representation of cases is described, and Protos’ explanation language is defined. Second, the classification process is explained. Hypothesis formation, exemplar selection, knowledge-based pattern matching, and similarity assessment are described. Third, the details of Protos’ learning mechanisms are presented. The selective retention of exemplars, integration of explanations into domain knowledge, and creation and revision of indices are described.

3.1 Representation

Protos’ domain knowledge is represented in a unified category structure. The category structure is a semantic network whose nodes are categories and whose edges are relational links acquired from teacher-provided explanations. A category can simply be a named node, or it can be linked to one or more exemplars which are retained cases. The relational links between categories embody general domain knowledge which provides cohesiveness by relating categories and enables inferences during classification. The category structure also contains indices which associate exemplar features with cate-
gories (reminders and censors), categories with exemplars (exemplar links), and exemplars with imperfectly similar exemplars (difference links).\footnote{Protos' internal representation of the sample category structure of Chapter 2 (Figures 2.1 and 2.2) is illustrated in Appendix B.}

3.1.1 The Representation of Cases

A case is described by a name, a classification, and a set of features. Its name uniquely identifies the case among all of the cases that Protos has processed. If the teacher cannot provide a unique name, Protos generates a unique symbol to identify the case.

The classification of a case is the name of the category to which the case has been assigned by Protos or the teacher. If the case is newly presented to Protos, the presence of a classification is optional. For example, during training, the teacher might choose to preclassify only those cases that Protos has no chance of correctly classifying independently. All cases retained as exemplars, however, must have classifications.

A case (or exemplar) description is a collection of features which represent attributes that the teacher thinks are potentially relevant to its classification. Each feature represents an independent attribute of the case. If relationships between attributes are relevant to the case's classification, they must be explicitly included in its featural description. For example, an arch (cf., [Win75]) might be described by the features \( \text{block}(a) \), \( \text{block}(b) \), \( \text{block}(c) \), \( \text{above}(c, a) \), \( \text{above}(c, b) \), \( \text{touches}(c, a) \), \( \text{touches}(c, b) \), and \( \text{doesNotTouch}(a, b) \). In such descriptions, it is the teacher's responsibility to enforce consistency.

Each feature is represented as a proposition or predicate. If a feature
is a proposition, its presence or absence is the only relevant information. If a feature is a predicate, its name identifies the feature, and its arguments define the feature's value. The teacher has great latitude in expressing a feature. For example:

- hasTail
- color(blue)
- numberOfLegs(four)
- above(seat,legs)
- legs(madeOf(wood),number(four))

Although Protos permits great flexibility in the expression of features, features can represent only qualitative information. Measurements input to Protos must be categorized by the teacher, collapsing ranges of values into qualitative descriptors. Protos has no a priori notion of relative magnitudes, and the teacher has only an indirect means of teaching it such information. Values can only be ordered by providing explanations of equivalence. For example the teacher might provide the explanation "weak is occasionally equivalent to moderate is occasionally equivalent to strong" which orders the values by making weak more transformationally distant from strong than it is from moderate. Consequently, the equivalence of weak and strong is a less certain inference than the equivalence of weak and moderate.

Each feature of a retained exemplar has an associated importance value which estimates the necessity of the feature being present for the exemplar to be classified as a member of a particular category. The value of a feature's importance ranges from 0 (for a spurious feature) to 1.0 (for a necessary one). Importance determines the amount of effort that knowledge-based
pattern matching will expend on trying to find a match for an exemplar feature among the features of a new case and indicates the degree of disconfirmation if a match is not found.

Every feature is a (virtual) category which may be represented by exemplars at a finer level of detail. The teacher might choose to use exemplars to describe case features which are important entities independent of the context of the case. For example, Protos might know that cars have the feature engine, and it might also have acquired several exemplars of engines. Consequently, Protos can deal with case descriptions at varying levels of granularity. A new case can be classified as a car if it has the feature engine or if it has a collection of parts that Protos can recognize as an engine.

3.1.2 The Representation of Explanations

Protos and the teacher communicate domain knowledge in terms of a predefined explanation language. An explanation is a chain of known relations connecting a feature (or conjunction of features) to another feature or a category. Protos’ explanations are somewhat different than the causal explanations that are typically believed to play an important role in expert classification. A causal explanation describes how an underlying state or process (such as a disease) is manifested as the features observed in the case. In contrast, Protos classifies a new case by explaining its similarity to a known exemplar. The underlying process (or cause) is considered to be the same as the process active in the exemplar.

Explanations are expressed in a predefined language of relationships and qualifiers. The explanation language was formulated to disambiguate the right arrow relation typically used in expert system rules. It has long been
recognized that a great deal of information is collapsed into the implication within an expert system's rule [Cla83]. More importantly, the nature of the evidence supporting such relationships is obscured [CDD*85].

Protos' expanded explanation language provides two benefits. First, human interaction with Protos is improved. When the teacher provides an explanation, the expanded vocabulary of relationships permits him to express it in terms that he feels captures his meaning more accurately than a simple chain of implications. Correspondingly, when Protos produces an explanation, it is expressed in the teacher's terms and hence is more suitable for human evaluation.

Second, relationships which should be processed differently are distinguished. When generating and evaluating explanations, Protos uses heuristics which are specific to the relational links comprising it. When generating inverse relationships during integration of an explanation into the category structure, Protos is able to determine their types and strengths more accurately.

Protos' explanation language is intended as a reasonable first step towards disambiguation of the right arrow. Particular domains which have unique requirements could benefit from additions to the language. The focus of the present work is, however, on the general problems of exemplar-based classification and learning. The explanation language was designed to provide reasonable support for this endeavor. No claims are made as to its sufficiency for expressing all relationships that may be of interest.

Protos' explanation language provides ten types of relational links (which were introduced in Section 2.2.2). They were chosen in an attempt to categorize the kinds of reasons underlying belief in the equivalence of two features or belief in the evidential relationship between a feature and a cate-
gory. Each type, in turn, provides from one to five specific links to be used in explanations (for a total of 29 links). Each link has a predefined inverse which allows it to be used bidirectionally in formulating explanations.

The strength of a relational link is determined by two factors. First, a link has a default numeric strength, between 0 and 1.0.\textsuperscript{2} This strength is an indication of the confidence that the system can place in an inference (i.e., the use of a link in an explanation) in the absence of a broader context for evaluation (e.g., the category in which a match is being attempted).

Second, the explanation language provides \textit{qualifiers} which allow the teacher to modify the certainty of a relational link. A link may be qualified with respect to strength (\textit{strongly}, \textit{moderately}, or \textit{weakly}), breadth of applicability (\textit{always}, \textit{usually}, \textit{sometimes}, or \textit{occasionally}), and degree of belief (\textit{certainly}, \textit{probably}, or \textit{possibly}). This variety of qualifiers is provided for expressive convenience rather than to denote qualitative processing differences.\textsuperscript{3}

Protos' explanation language provides three types of correlational links: simple, circumstantial-to-inferred, and definitional. Figure 3.1 lists these links.

A simple correlational link denotes that one feature is “statistically” related to another or to a category. For example, in Protos' language, one might say “barks is \textit{consistent} with dogs.”

Circumstantial-to-inferred links represent another type of correlational relationship which expresses mappings between observed features and inferred properties that are more relevant to a classification. These links are corre-

\textsuperscript{2}The default strengths of relational links are enumerated in Appendix C.

\textsuperscript{3}The degrees of effect of the qualifiers are enumerated in Appendix C.
• Simple Correlation
  - co-occurs
  - requires/is required by
  - is consistent with/exhibits
  - implies/is implied by
  - spurious

• Circumstantial-to-Inferred
  - if and only if
  - suggests/is inferred from

• Definitional
  - is equivalent to
  - definition implies

Figure 3.1: Correlational Explanation Links

ational because the details of the mappings are not specified. For example, “pointed canine teeth suggest carnivorous.”

A definitional link denotes an invariant relationship which is not elaborated. Common sense or a domain fact indicates the equivalence of two features. For example, “adolescent definition implies minor.”

Protos provides seven types of noncorrelational links: generalization/specialization, part-to-whole, causal, functional, predicate-to-argument, conjunction, and mutual exclusion. Figure 3.2 lists these links.

Generalization/specialization links provide the widely used taxonomic relationship between categories. The names chosen for these links require some discussion. Rather than simply adopting “generalization” and “specialization,”
- Generalization/Specialization
  - has typical generalization/has typical specialization
- Part-to-Whole
  - part of/has part
- Causal
  - causes/is caused by
- Functional
  - has function/is function of
  - enables/is enabled by
- Predicate-to-Argument
  - acts on/is acted on by
  - affects/is affected by
- Conjunction
  - and
- Mutual Exclusion
  - is mutually exclusive with

Figure 3.2: Noncorrelational Explanation Links
the names *has typical generalization* and *has typical specialization* were chosen because it was felt that introducing the notion of typicality in these relationships would make the semantics of any associated qualifiers clearer. A qualifier modifies the typicality of a link, indicating how common the given generalization or specialization is relative to other possibilities. For example, "bird *sometimes has typical specialization* crow" and "bird *occasionally has typical specialization* ostrich."

Part-to-whole links relate components to an assembly. For example, "wheels *part of* car."

A causal link strongly implies that one state causes another by a known process. For example, "air pollution *causes* acid rain."

A functional link states the function of a system or that a feature is known to play a role in that function. For example, "wings *enable* flight is *function of* airplane."

Predicate-to-argument links are used in special cases to relate featural predicates explicitly to their arguments. Normally, the relationship between a featural predicate and its arguments is implicit in the feature itself. Inferences relating the feature to its arguments are not possible. This is generally a wise restriction because the arguments rarely have a meaningful existence independent of the feature (*e.g.*, legs(four)). However, in special circumstances, allowing this type of inference may be desirable. For example "rupture_in(pipe) *affects* pipe."

A conjunctive link allows combining multiple pieces of evidence in a single explanation. A conjunction relates features only in the context of a particular explanation; it does not imply any general relationship between
the features. For example, 

"(hotTap(on) and coldTap(on)) is equivalent to

warmWaterFlow."

Negation is also necessary in explanations. To simplify Protos’ processing of explanations, the only form of negation provided in the explanation language is mutual exclusion. The inclusion of a general negation term (e.g., not) in the language would greatly complicate Protos’ evaluation of explanations. This complication is especially apparent if qualified relationships are considered. For example, if “X is usually not caused by Y,” should the presence of X in a new case provide negative evidence for Y (interpreting the relationship as “the presence of X makes Y unlikely”) or should it provide weak positive evidence (“interpreting the relationship as “X is occasionally caused by Y“)? The mutual exclusion link restricts negation to the former interpretation; the presence of a feature offers counter-evidence for a conclusion. This counterevidence, however, need not be absolute; the use of qualifiers allows expression of the degree of negative evidence. For example, “pulse(None) is usually mutually exclusive with live patient.”

Explanations in many domains could benefit from additional relationships to express such things as the spatial properties of objects or the temporal properties of events. However, the flexibility permitted in the representation of features as predicates offers one solution to the problem of an incomplete explanation language. The teacher can represent specialized relationships as features in which the predicate is the relationship and the arguments are the related entities. For example, consider above(seat, legs) which expresses a structural relationship between the parts of a chair. To relate this feature to the category

---

*A truly general purpose explanation language requires a family of negation terms. Protos’ language currently provides only one.*
chairs, the teacher might give the correlational explanation "above(seat, legs) is required by chairs." If the arguments of such a predicate are also case features, it is the teacher's responsibility to ensure that the overall case description is consistent.

A relational link is recorded in the category structure in symbolic form. It is a 4-tuple consisting of a from node, a list of qualifiers (which may be empty), a relational link name, and a to node. The strength of the link is determined, in context, when it is used in an explanation. The link's inverse is also derived and is similarly recorded. For example, the relationship "tail is usually part of dog" would be represented by the 4-tuple (tail, usually, partOf, dog) and its inverse (dog, usually, hasPart, tail).

3.1.3 Conditional Explanations

Early experience using Protos suggested that expert teachers were rarely willing to give unconditional explanations of featural equivalence. Consequently, the explanation language was extended to permit conditional explanations. A conditional explanation has the general form "if condition then explanation." For example, "if the category is apples then color(green) is sometimes equivalent to color(red)."

Three forms of conditional antecedents are provided:

1. the category into which Protos is trying to classify the new case.
2. other features of the new case.
3. the features of the exemplar being matched.

An explanation condition is independently associated with each relational link that is constructed from a conditional explanation. Such a link
is a 5-tuple, which is identical to a relational link with the addition of the explanation condition. For example, the conditional relationship about apples used as an example at the beginning of this section is represented by the 5-tuple (category=apples, green, sometimes, equivalentTo, red), and its inverse is similarly represented.

### 3.1.4 The Representation of Indices

In addition to being connected by relational links, the nodes of Protos' category structure are connected by indexing links. There are four types of indexing links: remindings, censors, exemplar links, and difference links. These indexing links connect different types of nodes in the category structure and serve different purposes in classification.

Remindings link features to categories. A reminding is the compilation of a feature-to-category explanation which conveys the teacher's belief that the presence of a feature in a new case suggests a particular classification. A reminding is a triple of the form (feature, category, strength). The strength of a reminding is a number between 0 and 1.0 which is a heuristic estimate of the strength of the relationship expressed by the underlying explanation. Multiple remindings can be associated with a feature; however, unlike probabilities, their strengths are not constrained to sum to 1.0.

Censors represent negative associations between features and categories. A censor can be viewed as a negative reminding. The strength of a censor conveys the degree of disconfirmation that a feature provides for a category; the censor may be absolute, or it may have a value between 0 and -1.0, mirroring the values taken by remindings.

Exemplar links associate exemplars with the categories of which they
are members. An exemplar link is a pointer from a category name in the
category structure to a case representation consisting of an exemplar name
and a set of features. Each exemplar link is annotated with a prototypicality
rating which is a positive real number. An exemplar's prototypicality rating
is determined by its successful use in classification; the exemplar is credited
according to the quality of every correct match in which it participates.⁵

Difference links associate “near miss” exemplars in the category struc-
ture. Difference links may be intercategory or intracategory. An intercategory
difference link records the reasons that a previous match was incorrect; it is
annotated with important featural differences between two exemplars. An in-
tracategory difference link notes a previous correct match in which an existing
exemplar was inadequately similar to a new case that was subsequently retained
as an exemplar; the difference link is typically annotated with all unmatched
features which are not believed to be spurious. A difference link is represented
by a pair of 5-tuples of the form: (exemplar name, category name, neighbor’s
name, neighbor’s category, list of unmatched features of neighbor).

### 3.2 Classification

Classification is a two-step process of hypothesis formation and con-
firmation. First, remindingcs and censors associated with the features of a new
case are heuristically combined to produce an ordered list of categories which
are possible classifications. Second, Protos pursues these hypotheses in order

---

⁵Prototypicality ratings are not normalized to the range 0 to 1.0 because normalization
would discard useful information about the relative prototypicalities of exemplars in the
population as a whole. This information currently is not used but is believed to be useful for
extensions to Protos. For example, if Protos is reminded of a general category that does not
contain exemplars, it could use prototypicality ratings of exemplars in categories subordinate
to that category to determine the best exemplars of the general category.
of preference until one is confirmed by locating an exemplar which matches the new case. Within a category, exemplars are selected for attempted matches in order of their prototypicality; however, difference links may provide shortcuts in the search for a match. A process known as knowledge-based pattern matching attempts to match the case to an exemplar by using domain knowledge to explain the equivalence of their features. The algorithm for classification is presented in Figure 3.3.

This section discusses the roles of indices in classification, the process of knowledge-based pattern matching, and Protos' similarity metric.

3.2.1 The Use of Indices in Classification

Reminders and Censors: When a new case is input for classification, initial hypotheses are supplied by reminders associated with the features of the case (Step 1 of the classification algorithm in Figure 3.3). The reminders are retrieved and heuristically combined to determine likely classifications (Step 2 in Figure 3.3). The algorithm for combining reminders is presented in Figure 3.4.

First, the numeric strengths of all reminders to each category and exemplar are summed. This calculation also takes into account any censors (triggered by the features of the new case) which are associated with categories of which Protos is reminded. Nonabsolute censors, which are negative numbers, are included in the sum. Any categories whose totals are less than or equal to 0 or for which absolute censors have been triggered are removed.

Second, reminding strength is inherited through taxonomic relationships. For example, "whippet has generalization dog has generalization mammal" is such a relationship between whippet and mammal. If Protos is reminded
1 Examine case features to collect reminders and censors
2 Heuristically combine reminders and censors to create hypothesis list
3 Sort hypothesis list in order of decreasing strength

REPEAT
    REPEAT
        Remove strongest hypothesis from hypothesis list
        If hypothesis is a category
            then Determine its most prototypical exemplars
        Retrieve features of best exemplar
        Add names of other exemplars to hypothesis list
        in order of decreasing prototypicality
        else \{hypothesis is an exemplar\}
        Retrieve features of exemplar

9 Similarity := 1.0
For each non-spurious feature of exemplar
    do Knowledge-based pattern matching to case features
    Similarity = Similarity * Degree-of-match
If Similarity < 1 \{not a perfect match\}
    then Improve match \{try matches to other exemplars suggested by difference links\}
UNTIL Similarity >= threshold

13 Present match to teacher
UNTIL teacher approves match

Figure 3.3: The Protos Classification Algorithm
Retrieve remindings and censors using features of case.
If case is classified
then add strong reminding to classification.
Sum duplicate remindings and nonabsolute censors.
Remove all remindings to categories blocked
by absolute censors.
Strengthen related remindings.
Strengthen remindings to exemplars.

Figure 3.4: The Protos Algorithm for Combining Remindings

of both *whippet* and *mammal*, the more specific category *whippet* will inherit
the strength of the reminding to *mammal*.

Third, reminding strength is transferred from categories to specific exemplars whenever it is possible to do so. If Protos is reminded of both a category and an exemplar within the category, the exemplar inherits the strength of the combined reminding to the category. The category is retained in the set of hypotheses because the reminding to the exemplar may not result in an adequate match. The inheritance operation ensures that remindings to specific exemplars within a category will be pursued before a more general reminding to the category itself.

The result of the process of heuristically combining remindings is an ordered list of remindings to exemplars and exemplar-containing categories which contains no duplicates and favors remindings to specific categories and particularly to specific exemplars when they occur (Step 3 in Figure 3.3).

**Prototypicality:** When Protos' best hypothesis is a category, the prototypicality ratings of exemplars within the category are used to select candidates for knowledge-based pattern matching (Step 5 in Figure 3.3). Prototypicality
defines a partial ordering of exemplars within the category. The most prototypical exemplars have the greatest similarity to other category members and hence are \textit{a priori} most likely to match new instances of the category.

During classification, Protos will potentially try to match several exemplars of a category in the order of their prototypicality. The number of exemplars tried should be a function of factors such as the strength of the reminding to the category, the polymorphism of the category, the competing remindings, and the importance of reaching a classification (\textit{e.g.}, how much effort should be expended on the task). However, the current implementation of Protos takes a simpler approach. When Protos considers a category, it is willing to attempt matches to its four best exemplars, ranked by prototypicality. The use of these four exemplars \textit{plus} all of the exemplars connected to them by difference links generally provides adequate testing of a hypothesized classification.

Since the exemplars of a category differ in prototypicality, they are not equally likely to match a new case. In many instances, another category may provide an exemplar that is more likely to match the new case than a less prototypical exemplar of the current category. Therefore, Protos apportions the strength of the combined reminding to a category to the selected exemplars based on their relative prototypicality. The most prototypical exemplar receives the full reminding strength. Other exemplars receive reminding strength in proportion to their prototypicalities relative to the prototype. For example, an exemplar which is rated as half as prototypical as the prototype would receive half of the strength of the reminding to the category. Knowledge-based pattern matching is then attempted between the features of the prototype and new case. The other selected exemplars are added to the list of hypotheses.
(Step 7 in Figure 3.3) and compete with other hypotheses for selection in the event that a match cannot be established.

**Difference Links:** When knowledge-based pattern matching results in an imperfect match between the features of the selected exemplar and the features of the new case, difference links associated with the exemplar may suggest better matches (Step 12 Figure 3.3). Each difference link points to another exemplar that was incorrectly (or imperfectly) matched to the current exemplar during a past classification attempt. The difference link is annotated with features which represent significant featural differences between the two exemplars (i.e., nonspurious features which were not matched when the two exemplars were compared).

Every difference link with features matching those of the new case is a candidate for traversal. The candidate difference links are ordered by the summed importances of the matching features. The one offering the most potential improvement is selected and traversed. However, the exemplar associated with the difference link may not be more similar to the new case. The degree of similarity between the new case and this exemplar is assessed by knowledge-based pattern matching. If no improvement is found, the next best difference link is traversed (if one exists). The process of traversing difference links to improve a match is presented in Figure 3.5.

The process of improving an imperfect match via difference links is hill-climbing search. When a difference link leads to a better match, the old exemplar is discarded, and the process continues recursively; it stops when no improvement is possible. Difference links may take Protos to exemplars of new categories or to different exemplars within the same category.
function IMPROVE MATCH(Exemplar)
For all difference links from exemplar
  do If difference link has features matching new case
      then Calculate the total importance of the matched
          features
          Add the target exemplar to the ordered list
          of candidates
While list of candidates is not empty
  do Match new case to candidate
      If Similarity > Current Similarity
      then IMPROVE MATCH(candidate)
          Exit while
Return currently matched exemplar

Figure 3.5: Improving an Imperfect Match Using Difference Links

3.2.2 Knowledge-Based Pattern Matching

Protos uses knowledge-based pattern matching to assess the similarity
of a new case to a recalled exemplar (Step 10 of Figure 3.3). Knowledge-based
pattern matching searches for a chain of known relations linking two nodes
in the category structure (e.g., features). It is a uniform cost procedure for
searching an and/or graph. The procedure begins at the target node (e.g., an
exemplar feature) and backward-chains until it finds a chain of relations linking
a source node (e.g., a case feature) to the target or until the effort allocated
for the search has been expended. If a chain of relations is found, it is taken
to be an explanation of how the source implies the target. The algorithm is

---

6The "or" nodes of the graph represent features and categories. The "and" nodes represent
conjunctions of features which occurred in previous, teacher-provided explanations.
While path cost < allocated search effort
do Determine best frontier node of search graph
  If best frontier node is feature of new case
    then Report success; Exit while
  else If best frontier node has no successors in
    category structure
    then Report failure; Exit while
  else Retrieve the nodes linked to best frontier
    node in the category structure
    Add nodes to search graph as successors of
    best frontier node
    Backpropagate cost
    Re-evaluate active paths
    If cost > allocated search effort
      then Report failure; Exit while
{end do}

Figure 3.6: The Knowledge-Based Pattern Matching Search Algorithm

presented in Figure 3.6.

The quality of a chain of relations is determined by heuristic evaluation of the relational links which comprise it. The quality of each link in the chain is determined by:

- the default strength of the type of relational link.
- the qualifiers modifying its strength.
- the other relational links involved in the explanation.
- the type of node to which the link leads.

The latter two determinants of explanation quality are assessed by heuristics which are invoked during knowledge-based pattern matching each time Protos chooses a new link from the category structure to extend a chain of relations.
Protos has 36 such heuristics, which are listed in Appendix D. These heuristics operate by inhibiting and promoting links based on the context of the category in which the match is being attempted and the other inferences in the explanation.

The majority of the heuristics are inhibitory. They prevent Protos from finding chains of inferences which do not constitute valid explanations. For example, in an explanation of featural equivalence, an explanation cannot involve the category in which the match is being attempted. This heuristic prevents bogus explanations of featural equivalence such as "barks is consistent with dog which has part tail."

Heuristics which promote inferences are less numerous than inhibitory heuristics because of the way in which heuristics were formulated during the initial testing of Protos. Every time Protos produced a poor explanation or failed to produce a good explanation which was known to be possible, the situation was analyzed to determine if a general heuristic applied. The bulk of this effort was directed towards eliminating poor explanations.

The certainty (i.e., strength) of an explanation linking two features is distilled into a numerical score, ranging from 0 to 1.0. This value is computed by multiplying the heuristically determined certainties associated with each relational link of the explanation (and is used in Step 11 in Figure 3.3). The computation is, in effect, an assessment of the plausibility of the explanation. The general assumption is made that each link is independent, but the evaluation heuristics are intended to recognize dependencies and to adjust link certainties accordingly. A direct match between features always has a certainty of 1.0.

The effort expended to match an exemplar feature is a function of its
importance. The search for an explained equivalence to an exemplar feature
terminates when the quality of the best inference path from the feature falls
below 1.0 minus importance without encountering a case feature (importance
ranges from 0 to 1.0 with 1.0 being considered a necessary feature). When a
feature is not matched, the certainty of its presence is taken to be 1.0 minus
importance. The rationale is that this value represents an upper bound on the
certainty of any possible explained equivalence.

Because classification in Protos is model-directed, only exemplar fea-
tures are considered in the assessment of similarity between a new case and an
exemplar. Unmatched case features are considered to be potential noise in the
case description.

To this point, the discussion has assumed that a chain of relational
links between a case feature and an exemplar feature is an explanation of equiv-
alence. In one case, however, a connection can provide disconfirming evidence
of similarity. If a mutual exclusion link is involved in the chain, the match
between cases is disconfirmed to a degree corresponding to the qualification of
the mutual exclusion link. The effect of a mutual exclusion link on the match
is determined after the quality of the match is assessed without consideration
of this disconfirmation. Its possible effects are:

- *always* or *strongly mutually exclusive* disallows the match.
- *usually mutually exclusive* disallows all but an otherwise perfect match.
- *sometimes* or *moderately mutually exclusive* disallows all but a strong
  match.
- *occasionally* or *weakly mutually exclusive* only disallows weak matches.
If a match involves more than one mutual exclusion relationship, they are combined additively (e.g., two usually mutually exclusive relationships are equivalent to one always mutually exclusive relationship).

3.2.3 The Assessment of Similarity

The similarity of a new case to the recalled exemplar is a function of the certainties of the explanations linking their features and of the importances of exemplar features that are not matched. The similarity rating of the match is computed incrementally (Step 11 in Figure 3.3). Protos ranks the exemplar’s features in order of decreasing importance and then attempts to explain the equivalence of a case feature (or set of features) to each nonspurious exemplar feature. Initially, the two cases are assigned a similarity rating of 1.0. After each exemplar feature is processed, the similarity rating is multiplied by the certainty of the feature’s presence (as determined by successful or failed knowledge-based pattern matching). If it falls below a predetermined threshold, the match is abandoned.  

8 The use of a multiplicative similarity function allows each exemplar feature to affect the assessment of similarity in proportion to its importance (cf., the similarity function of the Context Model in [DS78]). Important features

---

7The threshold is set to 0.2 which is slightly less than the product of the default strengths of two moderate correlational explanations of featural equivalence involving important exemplar features (e.g., $f_1$ is sometimes consistent with $f_2$). In other words, a match is abandoned if it involves more than two such explanations.

8The implementation of Protos makes use of several numeric parameters. These parameters can be viewed as heuristics compiled from common sense possessed by the system builder. Their purpose is to enable Protos to act reasonably in the absence of specific domain knowledge. Experimentation provides some support for the claim that the emergent behavior of the system is determined by the knowledge that it acquires through apprenticeship rather than its parameter values.
dominate in the calculation. Matching a number of unimportant features does not compensate for the failure to match an important one. For example, an unmatched necessary feature would contribute a factor of 0 to the calculation (1.0 minus its importance rating of 1.0), resulting in a similarity rating of 0 regardless of other featural matches.

3.3 Learning

Learning occurs when problem solving fails. When independent problem solving fails, Protos interacts with the teacher to acquire new information or to revise its category structure. Figure 3.7 presents a high-level description of how Protos learns.

Protos learns categories by selectively retaining exemplars. When Protos fails to classify a new case independently or fails to find an adequately similar exemplar within the correct category, the case is retained as a new exemplar. A case which Protos fails to classify is retained only if the teacher can provide its classification. A case which is successfully classified is retained if the match to a known exemplar does not meet the criteria for merging.

3.3.1 Merging Cases

A new case is merged with an existing exemplar when they match so closely that future cases that would match the new case would match the exemplar equally well (Steps 3 and 4 of the learning algorithm in Figure 3.7). Operationally, a new case is “mergable” if its features match all of the exemplar’s features that are not considered to be of low or spurious importance to the category.

An additional constraint is that the majority of the exemplar’s fea-
<Execute classification algorithm to find a matching exemplar>

if match is found
1 then Ask if match is correct
   If match is correct
      then If match is strong
               then Ask for explanations of unmatched features
               If match meets merge criteria
                  then Increase exemplar's prototypicality substantially
               Install pending difference links
      else Increase exemplar's prototypicality moderately
       ADD NEW CASE AS EXEMPLAR
    else Increase exemplar's prototypicality slightly
     ADD NEW CASE AS EXEMPLAR
9 else Reassess remembrings that led to exemplar
   If incorrect match is strong
      then Reassess importances of unmatched features
      Ask for new differentiating features
      Ask for censors
      Remember for future difference link
   <Try next best classification hypothesis>
else Ask for classification
   If classification is given
      then ADD NEW CASE AS EXEMPLAR
   else <fail>

procedure ADD NEW CASE AS EXEMPLAR
16 Record features of new case in category structure
17 Learn remembrings
18 Learn featural importances
19 Install pending difference links

Figure 3.7: Learning During Problem Solving
tures must be matched. Two similar cases should have a high degree of overall featural similarity. Since Protos' assessment of the importances of features can change, an unmatched feature may represent a distinguishing characteristic whose importance has not yet been realized. This constraint on merging is intended to minimize the possibility of such an occurrence while still allowing Protos to merge frequently.

Before merging a new case, Protos asks for the teacher's permission, allowing the teacher to decide if the new case represents an instance of the category that might be important to distinguish in the future. If the teacher concurs with Protos' decision, the new case is not retained, and the prototypicality of the exemplar is increased.

3.3.2 Acquiring a New Exemplar

When the decision is made to retain a new case as an exemplar, quite a bit of work may be necessary to integrate it into the category structure (Steps 16-19 in Figure 3.7). Protos must determine how the features of the new case contribute to belief in its classification. Protos uses knowledge-based pattern matching to generate an explanation of how each feature provides evidence for category membership (i.e., a chain of known relations linking the feature to the category). If it fails to find such an explanation, the teacher is asked to provide one. Protos uses the explanation to formulate a reminding (Section 3.3.5) and an inverse explanation to estimate the importance of the feature (Section 3.3.4). When a new exemplar is being retained as a result of a correct, but imperfect, match, Protos minimizes its effort by only processing features of the case which do not directly match features of the existing exemplar.
3.3.3 Acquiring Domain Knowledge via Explanations

Protos asks the teacher for explanations in two situations. The first is during the addition of a new exemplar to the category structure (as discussed in Section 3.3.2). The second is when it cannot infer the equivalence of a case feature to an unmatched exemplar feature in the context of a correct match (Step 2 in Figure 3.7). When the teacher approves a match, Protos attempts to perfect it by asking about all nonspurious exemplar features for which it cannot find equivalences within the new case.

When Protos obtains a new explanation, it is added to the category structure one relational link at a time (Section 3.1.2). Before adding a new link between two nodes (features or categories) in the category structure, Protos checks to see if one already exists. If it does, Protos must decide if the new link should be added between the nodes. It is added when:

- no previous link exists.
- the existing link and the new link are both unconditional, and the new link is stronger. In this case, the old link is removed.
- the existing link and the new link are both unconditional, the new link is weaker, but the teacher tells Protos to replace the existing link (when queried by Protos).
- the existing link and the new link have different explanation conditions associated with them. The new link is added without removing the existing one.
- the existing link has an associated condition and the new link is unconditional. If the existing conditional link is stronger, the unconditional link is added without removing it. Otherwise, the conditional link is replaced.
the existing link is unconditional and a stronger conditional link is given. The existing link is not removed.

- the existing link is unconditional and a weaker conditional link is given. The teacher is queried and the existing link is replaced if he concurs.

When a conditional explanation is added to the category structure, the explanation condition is independently associated with each relational link constructed from the explanation (Section 3.1.3). This allows the links to be combined into new explanations in contexts where such combinations are likely to be valid.

When a relational link is added to the category structure, its inverse is heuristically determined and is also added. Each relation in the explanation language has a known inverse. For example the inverse of is the function of is has function. If the strength of the relationship tends to be bidirectional (e.g., is equivalent to), the inverse relation is given the same qualifier(s) as the original relation. If it tends to be unidirectional (e.g., causes), the inverse is given a qualifier of moderate strength (e.g., sometimes). More accurate qualification is determined by interaction with the teacher if Protos uses the inverse link to generate future explanations.

3.3.4 The Generalization of Exemplars

The exemplars retained by Protos are generalized both implicitly and explicitly as learning progresses. As discussed in Section 2.3.3, implicit generalization occurs when domain knowledge (i.e., explanations) is acquired which increases the featural variation allowable in an acceptable match to an exemplar (Step 2 in Figure 3.7). Valid explanation paths from the features of an exemplar determine which matches to new cases can be inferred by knowledge-based pattern matching.
This section discusses the two ways in which exemplars are explicitly generalized. First, exemplars are explicitly generalized when low importance values are assigned to some of their features. Second, some feature-to-feature explanations associated with close matches cause exemplar features to be generalized.

Learning the Importance of Features: The importance of an exemplar feature determines the amount of effort that will be expended during knowledge-based pattern matching to attempt to match it to a feature of a new case. If a feature is considered to be unimportant, little effort will be expended, and failure to find a match will have little effect on the calculation of similarity.

The importance of a feature to a category (or to a particular exemplar) is estimated when the feature occurs in a new exemplar being added to the category structure (Step 18 in Figure 3.7). Knowledge-based pattern matching is invoked to determine a path of inferences from the category to the feature (e.g., "dog usually has part tail"). The explanation is evaluated to assess how strongly category membership implies the presence of the feature. As the explanation is assembled, each inference in the explanation is assigned a numerical certainty determined by the a priori strength of the relation, its qualifiers, and heuristic evaluation of the current context (as described in section 3.2.2). These certainties range between 0 and 1.0. They are multiplied to arrive at a numeric estimate of the feature's importance. A feature which is determined to be necessary for category membership is assigned an importance of 1.0, while a spurious feature (one that cannot be explained) is assigned an importance of 0.
Protos generally estimates features to be of moderate or low importance because of the way in which qualifiers are typically assigned to the links that comprise the category-to-feature explanation. When a new case is added as an exemplar, the teacher generally explains the relevance of its features to category membership in terms of how the presence of each feature implies category membership. This is opposite to the direction of an explanation used to determine importance. Since Protos determines and installs inverse links itself when a new explanation is given, the explanations from which importance is assessed are often composed entirely of system-generated links. These links are typically assigned moderate strength through qualification. Consequently, the product of link certainties is typically moderate or low.

When Protos misclassifies a new case by incorrectly matching an exemplar, it has the opportunity to revise the importance of the exemplar’s features. A misclassification often results from a match in which many of the exemplar’s features are unmatched, but the unmatched features are believed to be of low importance. When the teacher rejects such a match, Protos examines the exemplar’s unmatched features to determine which are believed to be of moderate or low importance and asks the teacher to reassess the importance of each (Step 10 in Figure 3.7). The teacher is told the importance in qualitative terms and is asked if he thinks the feature is of necessary, high, moderate, low, or spurious importance. If his answer differs from Protos’ current estimate, the feature’s importance is adjusted accordingly.

Protos also reassesses the importance of features when difference links are created (Steps 4 and 19 in Figure 3.7). When a feature is named as a

---

9The corresponding numeric importance values are 1.0, 0.75, 0.5, 0.25, an 0.
relevant difference between two cases, Protos conjectures that the feature has
general importance to the category to which the exemplar possessing it belongs.
If the feature is believed to be of low or spurious importance, the teacher is
asked for a reassessment.

Sometimes the importance of a feature needs to be revised downward.
Protos can do this when it fails to classify a new member of a category of which
it already has exemplars (i.e., reassessment is possible when it tried to match
the new case to existing exemplars of the category but failed to find an ade-
quate match). Such a failure may indicate that some of the exemplars’ features
are believed to be overly important. The most prototypical exemplar of the
category is retrieved to participate in the reassessment of featural importances.
Since prototypicality is based on family resemblance, the prototype is expected
to have the highest degree of featural commonality with other category mem-
bers. Protos determines which features of the prototype do not match those of
the new case and asks the teacher to reassess those considered to be of moderate
or greater importance.

**Generalizing Exemplar Features:** Protos can explicitly generalize an ex-
emplar by replacing some of its features with more general ones. Such gen-
eralization can only occur as a result of a close match between an exemplar
and a new case which results in merging. When a merge occurs, Protos exam-
ines the explanations of featural equivalence to see if any involve fragments of
generalization, functional, or causal hierarchies (e.g., “chair legs has function
seat support is the function of pedestal”). When it recognizes such an expla-
nation, it asks the teacher for permission to replace the exemplar feature with
the common generalization, function, or cause (e.g., seat support).
This operation makes a generalization which is implicit in Protos' domain knowledge explicit in the exemplar. The result is to make similar matches to the exemplar easier to achieve in the future by compiling a portion of the explanation into the exemplar feature.

3.3.5 Learning Indices

For Protos to make effective use of the exemplars that it retains, it must acquire indices that guide it to the exemplar most relevant to classifying a new case. These indices are reminding, censors, prototypicality ratings, and difference links. Protos learns and reassesses them based on the analysis of explanations, focused interaction with the teacher, and noting problem solving failures and successes.

**Learning Reminders**: Reminders are initially learned by analyzing the feature-to-category explanations given by the teacher when a new exemplar is added to the category structure (Step 17 in Figure 3.7). When given an explanation of the relevance of a feature to a category (or exemplar), Protos heuristically examines it to determine if a reminding should be compiled and which node in the category structure (i.e., term in the explanation) should be associated with the feature. First, the explanation is checked to ensure that it does not contain a mutual exclusion relation. If it does, no reminding is compiled. Next, the relational links in the explanation are examined one at a time to determine if the new reminding should be to the category itself or to an intermediate category (e.g., a generalization) which occurs in the explanation.

The presence of a has typical specialization link denotes that a feature is relevant to the category because it is associated with a superordinate (gen-
eral) category of which the category is a specialization. This suggests that the feature may be relevant to membership in sibling categories as well, and hence the reminding should be associated with the superordinate. For example, the explanation “fur is usually required by mammal has typical specialization cat” results in a reminding to mammal.

The presence of a has part link suggests that the feature is relevant to a larger assembly of which the category is only a part. The reminding therefore should be associated with the assembly as a whole. For example, the explanation “madeOf(metal) is usually required by car has part window crank” results in a reminding to car.

The presence of a link in the explanation weaker than the correlational relationship is sometimes consistent with also causes Protos to truncate an explanation. A weaker relationship than this gives Protos little confidence that the corresponding inference step is sound. The reminding is associated with the term preceding this weak link if the term is not the feature itself; otherwise no reminding is compiled. For example, the explanation “color(pink) is occasionally consistent with flower” does not result in a reminding because the feature color(pink) is the only term preceding the weak link.

These heuristics for determining the target of a reminding are not applied if the explanation relates a feature directly to the exemplar being added to the category structure. Because relationships between features and particular exemplars have proven to be relatively rare in the expert classification domain examined, it was felt that such relationships should be preserved whenever they are provided by the teacher. For example, the explanation “dented fender is part of my pickup is an exemplar of trucks” results in a reminding to my pickup.
After the target of the reminding is determined, its strength is determined by evaluating the strength of the underlying explanation using the heuristics associated with knowledge-based pattern matching. The reminding is assigned a strength between 0 and 1.0.

Originally, it was believed that only strong remindings should be retained. However, informal experiments were conducted in the domain of audiology to determine the effects of purging weak remindings, and it was discovered weak remindings provided important evidence during the hypothesis formation process. Many classifications were suggested by a preponderance of evidence rather than by the presence of a few unique features. Consequently, removing weak, but correct, remindings actually hurt the performance of the system.

Remindings are reassessed when they contribute to classification failures (Step 9 in Figure 3.7). A failure occurs if the teacher rejects a classification or Protos is reminded of a category which does not contain an exemplar similar to the new case. Protos has no way of assigning the blame to particular remindings, so all remindings which led it to try the category are reassessed.

The first step in reassessing a reminding is to attempt to regenerate the explanation linking the feature to the category of which it reminded Protos. Because knowledge is acquired incrementally, the explanation underlying a reminding may have changed after it was compiled. Protos performs knowledge-based pattern matching between the feature and the category to generate the current best explanation linking them. If no explanation can be generated, the reminding is removed. Otherwise, the strength of the reminding is replaced with the strength determined by evaluating the explanation. This may raise, lower, or preserve the reminding’s existing strength.

Next the reminding is checked for consistency with other remindings
associated with the feature. All reminders associated with the feature are retrieved and are checked for consistency. The reminding being reassessed is believed to be in conflict with reminders to categories which are not known to be superordinates, subordinates, or siblings of its target category. If these reminders are weak, the conflict is not thought to be serious, so no action is taken. However, if they are strong, the reminding being reassessed is weakened to reflect the knowledge that there are conflicting reminders.\(^\text{10}\) The other reminders are not weakened because they are not involved in the current problem solving failure and hence may be reliable.

If neither of these reassessment steps results in weakening the reminding, its strength is decremented very slightly to reflect diminished confidence. The reminding is removed from the category structure if its strength falls below a predetermined threshold (0.2),

After the strength of a reminding has been reassessed, its position in the category structure is also reassessed. A reminding may be moved to either a more general or specific category. If a feature reminds Protos of at least two of three categories that have a common superordinate, the reminding is moved to the superordinate, and it is given the strength of the strongest reminding to one of the subordinate categories. Conversely, if a general category with at least three subordinate categories shares a reminding with only one of the subordinates, the reminding to the general category is removed. This specializes

---

\(^{10}\)If the reminding is strong, its strength is reduced by 0.25 times the sum of the two strongest conflicting reminders. If it is moderate, its strength is reduced by 0.125 times the sum. If it is weak, it is reduced by the constant 0.025. This reassessment was originally conceived as the application of heuristic rules (e.g., If the reminding is strong and conflicts with two or more strong reminders, its strength should be reduced to moderate). The arithmetic operations were noted to approximate the results of the heuristics reasonably and efficiently.
the reminding by associating it uniquely with the subordinate category.

Another dimension on which remindings should be reassessed has been identified, but the mechanism has not been implemented. Sometimes the remindings associated with a feature are too diffuse; they point to too many unrelated categories. Such remindings do not provide information useful in discriminating among competing hypotheses, so they can be removed without depriving Protos of useful information.

**Learning Censors:** Censors are negative remindings associated with categories. A censor corresponds to an explanation of the form "f is mutually exclusive with this category." Censors are learned from discussion with the teacher (Step 12 in Figure 3.7). When a classification is rejected, the teacher is asked if any of the features of the new case are mutually exclusive with the classification. If he identifies such features, he is asked to rate the strength of the negative evidence as weak, moderate, strong, and absolute. The first three ratings correspond to remindings considered to be weak, moderate, and strong. A weak censor cancels the effect of a weak reminding, et cetera. An absolute censor prevents consideration of a category regardless of how many positive remindings exist.

**Learning Prototypicality:** The prototypicality of an exemplar is determined by its similarity to the other members of its category. Protos incrementally estimates the prototypicality of an exemplar by crediting its successful use in problem solving. When an exemplar is involved in a successful match, it gains prototypicality (Steps 3, 5, and 7 in Figure 3.7). Such a match can

---

11The first three ratings have the numeric values \(-0.25, -0.5\) and \(-1.0\).
potentially increase the exemplar's prototypicality rating by one unit. The exemplar is given 0.25 units for being involved in the match. If the match is close enough to justify merging the new case with the exemplar, it receives an additional 0.5 units. If the teacher approves the merge, the exemplar receives another 0.25 units (for a total of 1.0). The alternative of simply crediting the exemplar for the proportion of its features which match the new case would not make a distinction between matches to merged and unmerged cases. Because a merge is considered to be a greater problem solving success, such an occurrence is given greater weight.

**Learning Difference Links:** Protos learns difference links as the result of "near miss" matches during classification. When the teacher rejects a strong match, the system remembers this near miss by noting the unmatched features of both the new case and the incorrect exemplar (Step 13 in Figure 3.7). When the new case is finally added to the category structure as an exemplar or is merged, the near misses are recalled for discussion with the teacher. The teacher is reminded of each near miss and is told which features did not match. He is then asked to identify the features that represent important distinctions between the new case and the incorrectly matched exemplar, and vice versa. Protos installs difference links between the two exemplars, annotated with the discriminating features (Steps 4 and 19 in Figure 3.7).

This chapter has presented details of Protos' concept representation and of its classification and learning algorithms. It is hoped that sufficient information has been provided to enable accurate reimplementation from this
description.\textsuperscript{12}

The next chapter discusses the evaluation of Protos by its application to knowledge acquisition for heuristic classification in the domain of clinical audiology.

\textsuperscript{12}The truly obsessive reader can request a copy of the program from the author (being aware that it is approximately 7000 lines of Prolog code and 6000 lines of C code and is not always documented as well as one might hope).
Chapter 4

Evaluation of Protos

Protos has been evaluating by applying it to the problem of knowledge acquisition for expert, heuristic classification. In contrast to an everyday task, such as the common sense classification of animals, an expert classification task provides objective standards for evaluating the performance of a concept acquisition system. The system's learning performance is comparable to that of human students learning the same task and its classification performance is comparable to that of human journeymen and experts.

This chapter reports on the experimental application of Protos in the domain of audiology. The first section provides an overview. It discusses the domain and our experimental methodology. The second section presents three examples of Protos processing cases during its training. The final section presents and analyzes the results of the experiment. The knowledge acquired by Protos is characterized, and its classification performance is reported and compared with that of human subjects.

4.1 Knowledge Acquisition for Audiological Diagnosis

Protos' task in audiology was to learn to classify hearing disorders from patient descriptions. Dr. Craig Wier, a professor of Speech Communication at the University of Texas, served as Protos' teacher. He directly interacted
with Protos to present and discuss a sequence of cases. Unlike a traditional knowledge acquisition scenario, a human knowledge engineer did not serve as an intermediary between the expert and the system.

4.1.1 Comparison of Protos with a Human Audiologist

The task of diagnosing hearing disorders is typically performed by clinical audiologists, who also perform hearing tests and fit hearing aids. Audiologists are trained in university graduate programs at the Masters degree level. An audiologist must pass a national examination and then serve a one year clinical fellowship before achieving full professional status.

The interaction between Protos and an expert teacher is similar to that between a human student and a clinical instructor; both attempt to classify cases (and to explain their classifications) under expert supervision and receive detailed feedback on performance. However, some fundamental differences exist.

A human audiologist (or student) performs tests of a patient's hearing which produce raw numerical data. The audiologist abstracts these test results into qualitative descriptors, which are the fundamental data for the diagnostic process. In contrast, Protos requires direct input of these qualitative data. The teacher must perform the tests and transform the results for presentation to Protos. The required transformations embody common clinical abstractions; a human student learns similar ones by rote. For example, a patient's ability to hear a tone is termed "mildly diminished" if he requires 10-30 decibels more volume than the statistical average of the population. Because data abstraction is only a small part of the diagnostic process, Protos is not given an unfair advantage by having this work done for it. However, to minimize criticisms of
experimental methodology, the featural input given to Protos was restricted to a small set of standard test results, patient history information, and patient-reported symptoms.

Another difference involves the diagnosis of multiple disorders in a single ear. Because an ear is composed of loosely coupled subsystems, multiple problems can exist which are not causally related. An audiologist may reason constructively in such situations — constructing a hypothesis of how the presence of multiple disorders can explain the observed state of a patient's hearing. Protos has two different ways of recognizing multiple disorders. The first is to find more than one classification for a case. This is possible in situations where the symptoms of the disorders do not interact to mask one another. The second is to classify a case with multiple disorders by matching it to an exemplar in which the same combination of disorders was present. This requires that the teacher assign such cases directly to categories corresponding to multiple disorders. Such an enumeration of categories was possible in audiology, but in some domains, combinatoric problems may make this approach unrealistic.

The most fundamental difference between Protos and a human audiologist (or student) is the vast difference in total information that each can use in the diagnostic process. An audiologist reasons from sensory input, systematically acquired knowledge of audiology, and a broad range of general knowledge. The audiologist also has potential access to the knowledge of others via consultations. In contrast, Protos reasons from very limited knowledge. It is not systematically provided with a domain theory for audiology; it does not possess general common sense knowledge, and it does not have a realistic means for accessing the wealth of information possessed by its teacher.
4.1.2 An Overview of the Experiment

Protos was expected to exhibit demonstrably good performance after processing a number of cases comparable to that seen by a human student during graduate school — about 200 cases. The expert obtained 230 sequential cases of adult hearing disorder from the files of the Audiology Clinic at the Baylor College of Medicine in Houston, Texas.¹ Each patient record provided two cases (one for each ear). The cases were divided into a training set of 200, a test set of 25, and a reserve of 5 cases to replace any cases having diagnoses that the expert felt were questionable. Four of the training cases were replaced by reserve cases. The extra case was added to the test set, expanding it to 26 cases.

The cases were presented to Protos in the order in which they appeared in the files, which was essentially random. It was believed that this training order would provide a baseline test of Protos' performance and avoid the danger that the training order would unduly bias the outcome of the experiment.

The expert interacted directly with the system to present and discuss cases. He described each case directly to Protos in featural terms. When Protos reported a classification, the expert decided if it was correct and evaluated the quality of the match between the case and the exemplar recalled by Protos. When Protos asked for additional domain knowledge, he decided which explanations were appropriate and formulated them in Protos' explanation language.

A large amount of data was collected during training. Protos recorded

¹Professor James Jerger of the Baylor College of Medicine graciously provided access to these files.
every significant action that it performed (e.g., an attempted match), every interaction with the teacher (e.g., an explanation), and the outcome of processing every case. In addition, several intermediate versions of its category structure were retained. These data enabled accurate characterization of the knowledge acquired by Protos and assessment of the results of its learning.

Protos was modified during the course of the experiment. Two substantial changes were made after preliminary evaluation (of the results after the first 100 training cases) revealed inadequacies which affected system performance. First, censors were added. The addition of censors made the search process more efficient by providing a mechanism that enabled pruning of hypotheses. It also reduced Protos' susceptibility to mirages during classification (a common affliction of model-based approaches). Second, conditional explanations were added in response to a more fundamental problem. The expert was generally unwilling to give unconditional explanations of featural equivalence, so Protos was rarely able to merge cases which differed in the description of nonspurious features. Conditional explanations allowed the expert to restrict the contexts in which explanations of featural equivalence were valid. Consequently, he was more willing to provide them.

4.2 Examples of Protos in Action

This section discusses the processing of three cases used in Protos' training. These cases were in the second half of the 200 training cases. They

---

2 The experiment was continued after these modifications rather than restarted because the expert's available time was limited.

3 A mirage is a strong, but incorrect, match between an exemplar and a new case. Such matches are possible because the assessment of similarity is based only on exemplar features which are unmatched; unmatched case features are considered to be noise. When classifying a case, Protos employs censors to avoid categories in which such matches might occur.
illustrate Protos' interaction with its teacher, the acquisition of explanation material, the generation of explanations, the formation and revision of indices, and the selective retention of exemplars.

Each case is presented in the form of an annotated log of Protos' interaction with the teacher. The teacher communicates with Protos using a graphic and textual user interface which is illustrated in Figure 4.1. The interface graphically displays the reminders associated with the features of a case and abstract representations of any matches which are found during classification. Details about the combination of reminders and explanations of the matches are presented textually.
Interaction with the teacher takes two forms. Focused discussion, initiated by Protos, is primarily a textual question-and-answer process. Unfocused discussion, initiated by the teacher, is a mixed graphical and textual process used to provide general domain knowledge as explanations and to perform operations such as adding, modifying, and deleting features and remindings.

4.2.1 Example Case 1

The first two examples illustrate Protos' actions in classifying the ears of a patient who has a complex overall diagnosis. The patient's right ear clearly exhibits Meniere's Disease, a severe cochlear disorder. His left ear is probably developing the same disorder but does not yet display the symptoms strongly enough to be definitely diagnosed as such.⁴

The processing of the first case illustrates the addition of a new exemplar to Protos' category structure, the acquisition of feature-to-category explanations, and the compilation of new remindings.

The teacher begins the session by entering the features of the new case (Figure 4.2).⁵ He does not preclassify the case, indicating that Protos is to attempt a classification on its own.

As discussed in Chapter 3 (Section 3.2), classification is a two-step process of hypothesis formation and confirmation. Protos examines the features of the new case to formulate its diagnostic hypotheses. Figure 4.3 displays the remindings associated with the features of the new case.⁶ After the remindings

---

⁴These diagnoses were made by the Baylor clinicians who originally handled the case and were accepted by our expert.
⁵Definitions of audiological terms are included in the glossary.
⁶For clarity, only remindings to exemplar-containing categories are shown.
Case: p8590R
Unknown

m_sn_gt_4k
m_sn.lt_1k
ar_u(normal)
ar_c(normal)
o_ar_u(normal)
o_ar_c(normal)
tymp(a)
speech(normal)
air(mild)
history(vomiting)
history(dizziness)
history(fluctuating)

Figure 4.2: Features of the Example Case 1

are combined, two diagnoses emerge as the best possibilities. However, if neither of these diagnoses can be confirmed, Protos will try weaker hypotheses.

When the raw reminders are combined, the strongest reminders are:

normal_ear
possible_menieres

Protos is retrieving an exemplar of normal_ear.
Protos is trying to match p8452R of category normal_ear.
Case: p8590R
Unknown

m_sn gt .4k
m_sn lt .1k
ar_u(normal)
ar_c(normal)
o_ar_u(normal)
o_ar_c(normal)
tymp(a)
speech(normal)
air(mild)
history(vomiting)
history(dizziness)
history(fluctuating)

normal_ear
possible_menieres
cochlear_age
cochlear_unknown
bells_palsy
otitis_media
possible_brainstem
mixed_age_fixation

Figure 4.3: Raw Remindings from the Features of Case 1
Protos attempts to confirm its strongest hypothesis by establishing that one of its exemplars is similar to the new case. Because the new case has the abnormal test result air(mild), Protos cannot match the selected exemplar of normal_ear. Because the reminders to possible_menieres are almost as strong as those to normal_ear, Protos tries this hypothesis next instead of attempting to match another exemplar of normal_ear.

Protos is retrieving an exemplar of possible_menieres.
Protos is trying to match p8571R of category possible_menieres.

The new case is found to be similar to exemplar p8571R of the category possible_menieres (as discussed in Section 3.2.3). Protos examines difference links associated with the exemplar to attempt to locate an exemplar which is more similar to the case, but no better match is found. The match is displayed for discussion with the teacher (Figure 4.4). Many of the exemplar's features are directly matched by the new case, but important featural mismatches are noted. Protos lacked the knowledge necessary to formulate explanations of the equivalence of some of the case features to the unmatched exemplar features (as discussed in Section 3.2.2). The teacher is asked to assess the match.

The match that Protos found is a little shaky. Please examine the match and help Protos improve it.

Is possible_menieres the correct classification?
Case: p8590R
Unknown

Case: p8571R
Possible_menieres

m_sn.gt.4k
m_sn.lt.1k
ar.u(normal)
ar.c(normal)
o_ar.u(normal)
o_ar.c(normal)
tymp(a)
speech(normal)
air(mild)
history(vomiting)
history(dizziness)
history(fluctuating)

history(roaring)
bone(mild)
ar.u(normal)
ar.c(elevated)
o_ar.u(elevated)
o_ar.c(normal)
tymp(a)
speech(good)
air(mild)
history(dizziness)
history(fluctuating)

Figure 4.4: Matching Case 1 to an Exemplar of Possible_Menieres
>>yes

Do you think that the exemplar that Protos recalled is a good choice for interpreting this new case?

>>yes

The teacher is asked to explain the equivalence of case features to the unmatched features of the exemplar which are believed to be important to its classification.

Protos would like to improve this classification by discussing some of the exemplar features that it could not account for.

Protos could not account for the exemplar feature speech(good). Please explain how a feature of the new case is equivalent to speech(good) or type <none>.

>>none

Protos could not account for the exemplar feature ar_c(elevated). Please explain how a feature of the new case is equivalent to ar_c(elevated) or type <none>.

>>none

Protos could not account for the exemplar feature bone(mild). Please explain how a feature of the new case is equivalent to bone(mild) or type <none>. 
The teacher is now given a chance to provide Protos with unfocused instruction but chooses not to do so. He could have provided additional domain knowledge, altered the reminding, or altered the features of the case or the recalled exemplar.

Please make any desired changes to reminding, explanations, the current case, or the matched exemplar. This is your final opportunity to alter aspects of this match.

Please select something to edit or type <done>.

The match is not considered to be close enough to permit merging because important exemplar features have not been matched by the case. Protos decides to retain the case as an exemplar (as discussed in Sections 3.3.1 and 3.3.2).

Protos attempts to explain the relevance of the features of the new case to its classification, possible menieres. It only processes the features of the new case which were not directly matched by those of the exemplar. Before beginning, it asks the teacher if it should attempt to generate explanations independently before asking for them. The teacher opts to provide the explanations himself because he does not believe that Protos' independent attempts will be successful.
Protos is going to attempt to determine the relevance of each of the features of the new case to possible_menieres.

Should it ask for explanations before attempting to search?
>>yes

Protos could not account for the case feature m_sn_lt_1k. Please explain the relevance of the feature or type <none>.
>>m_sn_lt_1k sometimes caused by possible_menieres

The teacher's explanation is used to formulate a reminding, and its inverse is used to estimate the importance of the feature (as discussed in Sections 3.3.5 and 3.3.4).

Protos is creating a moderate reminding from m_sn_lt_1k to possible_menieres.

m_sn_lt_1k is estimated to be of moderate importance.

Protos could not account for the case feature m_sn_gt_4k. Please explain the relevance of the feature or type <none>.
>>none

Protos is trying to relate m_sn_gt_4k to possible_menieres.
When the teacher does not provide an explanation, Protos searches for one before deciding the feature in question is spurious. In this case, it finds an explanation which is used to formulate a reminding and to estimate the importance of m_sn_gt_4k. The reminding associates the feature with the general category cochlear.

Protos is creating a moderate reminding from m_sn_gt_4k to cochlear.

m_sn_gt_4k is estimated to be of moderate importance.

Protos could not account for the case feature o_ar_u(normal). Please explain the relevance of the feature or type <none>.

>>none

Protos is trying to relate o_ar_u(normal) to p8590R or possible_menieres.

Protos is creating a moderate reminding from o_ar_u(normal) to cochlear.

Protos notes that there are many, potentially conflicting, remindings associated with this feature and asks the teacher to modify them, but he declines. Protos' action exemplifies a useful type of static analysis that a knowl-
edge acquisition tool can perform to improve the quality of the knowledge base being acquired.

There are multiple moderate remindings from o_ar_u(normal):
  mixed
  bells_palsy
  cochlear_unknown
  normal_ear
  cochlear

Remove or adjust any remindings which are inappropriate
(or type <done>).
>>done

o_ar_u(normal) is estimated to be of moderate importance.

Protos could not account for the case feature history(vomiting).
Please explain the relevance of the feature or type <none>.
>>history(vomiting) sometimes caused by vestibular_problem
usually cooccurs possible_menieres

Protos is creating a moderate reminding from
history(vomiting) to possible_menieres.

history(vomiting) is estimated to be of moderate importance.
Similarly, the features ar_c(normal) and speech(normal) are processed.

Because the new case was fairly similar to p8571R, Protos installs intracategory difference links between them (as discussed in Section 3.3.5). It reminds the teacher of featural differences and asks which are important. Because the two cases are in the same category, he includes all unmatched features. When future cases of possible_menieres are processed, this information will help Protos find the best match within the category.

Protos previously mistook this case for the exemplar:

Name: p8571R
Category: possible_menieres
Features:
  bone(mild)
  o_ar_u(elevated)
  o_ar_c(normal)
  ar_u(normal)
  ar_c(elevated)
  tymp(a)
  speech(good)
  air(mild)
  history(roaring)
  history(dizziness)
  history(fluctuating)
The features possessed by this exemplar which were not matched by the current case are:

- o_ar_u(elevated)
- history(roaring)
- speech(good)
- ar_c(elevated)
- bone(mild)

Are <all>, <none> or <some> of these features criterial differences?
>>all

Protos notes that the exemplar's features o_ar_u(elevated) and history(roaring) included on the difference link are of spurious importance and asks the teacher to reassess their importances.

The features possessed by this case which were not matched by the exemplar are:

- m_sn_lt_1k
- m_sn_gt_4k
- o_ar_u(normal)
- ar_c(normal)
- speech(normal)
- history(vomiting)
Are <all>, <none> or <some> of these features criterial differences?

>>all

The teacher is given a final opportunity to provide unfocused instruction. He decides to give Protos some additional domain knowledge and indicates this by graphically selecting "new explanation" from an on-screen menu.

Please type an explanation or type <reject>

>>history(dizziness) sometimes caused by vestibular_problem

4.2.2 Example Case 2

The teacher presents the patient's other ear to Protos (Figure 4.5). This ear has a mild impairment but is probably developing Meniere's Disease. The processing of this case illustrates how classification failures result in the reassessment of Protos' reminders, the addition of censors, and the reassessment of the importance of features with respect to categories.

Figure 4.6 displays the reminders associated with the features of the case. After heuristically combining them, Protos favors two hypotheses. The combined reminding to mixed_cochlear_aging.fixation is considered to be stronger because Protos is also reminded of the related categories
Case: p8590L
Unknown

m_sn_gt_4k  tym(a)
ar_u(normal)  speech(normal)
ar_c(normal)  air(normal)
o_ar_u(normal)  history(vomiting)
o_ar_c(normal)  history(dizziness)

Figure 4.5: The Features of Example Case 2

cochlear_age and cochlear.7

When the raw reminders are combined, the strongest reminders are:

mixed_cochlear_age_fixation

normal_ear

Protos is retrieving an exemplar of mixed_cochlear_age_fixation.
Protos is trying to match p8433L of category
mixed_cochlear_age_fixation.

Protos rejects this classification because cases involving fixation must
exhibit another type of tympanogram (i.e., another value of the feature tymp).

7The individual reminders to general categories, including cochlear, are not shown in
the figure.
Case: p8590L
Unknown

m_sn_gt_4k
ar_u(normal)
ar_c(normal)
o_ar_u(normal)
o_ar_c(normal)
tymp(a)
speech(normal)
air(normal)
history(vomiting)
history(dizziness)

normal_ear
possible_menieres
cochlear_age
cochlear_unknown
bells_palsy
otitis_media
possible_brainstem
cochlear_pos_noise
mixed_age_fixation

Figure 4.6: Raw Remindings from the Features of Case 2
Because this classification failed, Protos reassesses the remainings that suggested it. Protos' objective is to identify remainings which can be altered to prevent similar failures in the processing of future cases.

Protos is reassessing the remainings to mixed_cochlear_age_fixation which led it to try this match.

Reassessing reminding from speech(normal) to mixed_cochlear_age_fixation

Reassessing a reminding is a three step process (as discussed in Section 3.3.5). First, the explanation linking the feature to the target of the reminding is regenerated to determine if it is still valid. If the certainty of the explanation differs from the strength of the reminding, the reminding is revised. Second, the reminding is checked for consistency with other remainings associated with the feature. If such remainings exist, the current reminding is weakened. Third, if neither of the previous steps changed its strength, the reminding is weakened slightly to penalize its role in suggesting an incorrect classification.

Reassessing reminding from ar_c(normal) to cochlear

Moving reminding to cochlear_age.

Because cochlear_age is the only subcategory of the general category cochlear of which ar_c(normal) also reminds Protos, the system moves the reminding to the more specific category.
After similarly assessing reminders associated with c_ar_c(normal) and c_ar_u(normal), Protos pursues its next best hypothesis, normal_ear.

Protos is retrieving an exemplar of normal_ear.
Protos is trying to match p8452R of category normal_ear.

Protos finds an adequately matching exemplar of normal_ear (Figure 4.7), but the teacher rejects this classification. When the teacher rejects a classification, Protos performs two types of actions. First, it reassesses the reminders which suggested the classification. Second, it seeks ways to differentiate further the new case from the exemplar. It asks the teacher to reassess the importances of unmatched features, asks for new discriminating features, and asks for censors.

One of the unmatched features of the exemplar of normal_ear was believed to be of spurious importance (and hence irrelevant to the match). Protos asks the teacher to consider raising its importance, thereby decreasing the similarity between the two cases.

Some of the unmatched exemplar features may have greater importance to p8452R or normal_ear than Protos currently believes. Please reassess the importances of these features as you are asked.

static(normal) is believed to be of spurious importance.
What do you believe its importance to be?
Case: p8590L
Unknown

<table>
<thead>
<tr>
<th>m_sn_gt_4k</th>
<th>ar_u(normal)</th>
<th>ar_c(normal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar_c(n(normal)</td>
<td>o_ar_u(normal)</td>
<td>o_ar_c(normal)</td>
</tr>
<tr>
<td>tymp(a)</td>
<td>speech(normal)</td>
<td>air(normal)</td>
</tr>
<tr>
<td>history(vomiting)</td>
<td>bone(normal)</td>
<td>static(normal)</td>
</tr>
</tbody>
</table>

Case: p8452R
Normal_ear

Figure 4.7: Matching Case 2 to an Exemplar of Normal_Ear
Next, the teacher is asked to provide new case or exemplar features that can help Protos to discriminate between the two cases, but he does not do so.

Finally, he is asked if any of the features of the new case should act as censors for the category normal_ear.

Protos mistakenly classified this case as normal_ear.
Enter any features of the new case, which are inconsistent with this classification.

Type or pick an object feature or type <done>
>>m_sn_gt_4k

How inconsistent is m_sn_gt_4k with normal_ear?
(<weakly>, <moderately>, <strongly>, <absolutely> or <reject>)
>>strongly

He also identifies history(vomiting) and history(dizziness) as censors of moderate strength.

Because Protos' two strongest hypotheses were disconfirmed, it pursues a weaker hypothesis, possible_menieres. This was the diagnosis of the patient's other ear, and as noted earlier, the teacher suspects that this ear is developing the same disorder. The first exemplar retrieved is not adequately
similar to the new case. However, because this is by far Protos’ strongest remaining hypothesis, it tries to match another, less prototypical, exemplar of possible_menieres.

Protos is retrieving an exemplar of possible_menieres.
Protos is trying to match p8571R of category possible_menieres.

Protos is retrieving exemplar p8590R.
Protos is trying to match p8590R of category possible_menieres.

The second exemplar, p8590R (which is the other ear of this patient), is somewhat similar to the new case. Examination of difference links does not locate a more similar exemplar, so this match is displayed for discussion (Figure 4.8).

All featural matches are direct except for a fairly elaborate explanation relating history(vomiting) to history(fluctuating):

history(vomiting) sometimes causedBy vestibular_problem
usually cooccurs possible_menieres usually cooccurs
with history(fluctuating)

In effect, this explanation relates the two features because they are both features of possible_menieres. This is a poor explanation, which Protos would have heuristically rejected had both of the features been directly linked to
<table>
<thead>
<tr>
<th>Case: p8590L</th>
<th>Case: p8590R</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unknown</strong></td>
<td>Possible_menieres</td>
</tr>
<tr>
<td>m_sn_gt_4k</td>
<td>m_sn_lt_1k</td>
</tr>
<tr>
<td>ar_u(normal)</td>
<td>ar_u(normal)</td>
</tr>
<tr>
<td>ar_c(normal)</td>
<td>ar_c(normal)</td>
</tr>
<tr>
<td>o_ar_u(normal)</td>
<td>o_ar_u(normal)</td>
</tr>
<tr>
<td>o_ar_c(normal)</td>
<td>o_ar_c(normal)</td>
</tr>
<tr>
<td>tymp(a)</td>
<td>tymp(a)</td>
</tr>
<tr>
<td>speech(normal)</td>
<td>speech(normal)</td>
</tr>
<tr>
<td>air(normal)</td>
<td>air(mild)</td>
</tr>
<tr>
<td>history(vomiting)</td>
<td>history(vomiting)</td>
</tr>
<tr>
<td>history(dizziness)</td>
<td>history(fluctuating)</td>
</tr>
</tbody>
</table>

Figure 4.8: Matching Case 2 to an Exemplar of Possible_Menieres
possible_menieres. However, the presence of the intervening node vestibular problem prevented Protos from recognizing the weakness of the explanation.

Although the classification is reasonable, the teacher rejects it because the patient's hearing is still fairly normal (as Protos earlier surmised).

Protos treats the match as a classification failure. It reassesses the findings which suggested the classification then asks the teacher to reassess the importances of the unmatched exemplar features. Next, the teacher is asked if any features of the new case should serve as censors for the category possible_menieres. He chooses to associate a weak censor with the feature speech(normal) even though it occurs in case p8590R, which is an exemplar of the category. Information such as this is difficult for a learning program to acquire autonomously.

Protos subsequently tries exemplars of four more categories, which are weak hypotheses, before finding an exemplar of the "catchall" category cochlear_unknown which weakly matches the new case. The teacher accepts this classification, and case p8590L is retained as a new exemplar of this category. Intercategory difference links are added to the exemplars of normal_ear and possible_menieres that were matched by this case.

4.2.3 Example Case 3

The final example illustrates Protos' use of a difference link to find the exemplar of the category cochlear_poss_noise which most closely matches a new case. It also illustrates how teacher-provided explanations can supply Protos with the knowledge that it needs to recognize the strong similarity of two cases, allowing them to be merged. Figure 4.9 presents the features of the case, and the associated findings are shown in Figure 4.10.
Case: p8600L
Unknown

m_sn.gt.2k  ar_u(normal)
notch_at.4k  ar.c(normal)
history(noise)  static(normal)
speech(normal)  tymp(a)
o_ar.c(normal)  air(normal)
o_ar.u(elevated)

Figure 4.9: The Features of Example Case 3

Protos' best hypothesis, *mixed_cochlear_age_fixation*, is rejected because the patient has *tymp(a)* which is mutually exclusive with the required type of tympanogram in cases involving fixation. Therefore, Protos reassesses the reminders that suggested this hypothesis.

The next best hypothesis is *cochlear_age_and_noise*. This hypothesis is rejected when Protos cannot find a case feature that is equivalent to *age_gt_60*, a required feature of age-related diagnoses.

Protos tries the more general hypothesis *cochlear_pos_noise*. The exemplar retrieved by Protos is a fairly close match (Figure 4.11), but a better match is found by traversing an intracategory difference link.

Protos is trying to improve the match to p8594R of category *cochlear_pos_noise*.
Figure 4.10: Raw Reminders from the Features of Case 3
p8527R of category cochlear_poss_noise is a closer match
to the new case.

The second match is presented in Figure 4.12. Most of the features of
the new case match those of the exemplar directly. However, when the teacher
described the patient to Protos, he used the feature notch_at_4k, a different
term than he had used when some of the earlier cases were entered. Because of
an explanation failure in a past case, Protos had acquired the knowledge that
allowed it to explain that notch_at_4k is equivalent to notch_4k. The teacher
accepts the classification and choice of a matching exemplar.

Protos asks the teacher to explain the equivalence of features of the
new case to the unmatched exemplar features, which Protos believes to be
nonspurious to the classification. The second explanation he provides is condi-
tional; it is only valid in the context of this diagnostic category (as discussed
in Section 3.1.3).

Protos would like to improve this classification by
discussing some of the exemplar features that it
could not account for.

Protos could not account for the exemplar feature
m_m_sn_gt_2k. Please explain how a feature of the new case
is equivalent to m_m_sn_gt_2k or type <none>.
>>m_sn_gt_2k sometimes equivalent m_m_sn_gt_2k
<table>
<thead>
<tr>
<th>Case: p8600L</th>
<th>Case: p8594R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>Cochlear_pos_noise</td>
</tr>
</tbody>
</table>

|                               |                               |
| m_sn_gt_2k                     | mod_sn_gt_3k                   |
| notch_at_4k                    | notch_at_4k                    |
| history(noise)                 | history(noise)                 |
| speech(normal)                 | speech(normal)                 |
| o_ar_c(normal)                 | o_ar_c(normal)                 |
| o_ar_u(elevated)               | o_ar_u(normal)                 |
| ar_u(normal)                   | ar_u(normal)                   |
| ar_c(normal)                   | ar_c(elevated)                 |
| static(normal)                 | static(normal)                 |
| tymp(a)                        | tymp(a)                        |
| air(normal)                    | air(normal)                    |

Figure 4.11: Matching Case 3 to 1st. Exemplar of Cochlear_pos_noise
<table>
<thead>
<tr>
<th>Case: p8600L</th>
<th>Case: p8572R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>Cochlear_pos noise</td>
</tr>
<tr>
<td>m_sn_gt_2k</td>
<td>m_m_sn_gt_2k</td>
</tr>
<tr>
<td>notch_at_4k</td>
<td>notch_4k</td>
</tr>
<tr>
<td>history(noise)</td>
<td>history(noise)</td>
</tr>
<tr>
<td>speech(normal)</td>
<td>speech(normal)</td>
</tr>
<tr>
<td>o_ar_c(normal)</td>
<td>o_ar_c(normal)</td>
</tr>
<tr>
<td>o_ar_u(elevated)</td>
<td>o_ar_u(elevated)</td>
</tr>
<tr>
<td>ar_u(normal)</td>
<td>ar_u(normal)</td>
</tr>
<tr>
<td>ar_c(normal)</td>
<td>ar_c(elevated)</td>
</tr>
<tr>
<td>static(normal)</td>
<td>static(normal)</td>
</tr>
<tr>
<td>tymp(a)</td>
<td>tymp(a)</td>
</tr>
<tr>
<td>air(normal)</td>
<td>air(normal)</td>
</tr>
</tbody>
</table>

Figure 4.12: Matching Case 3 to 2nd. Exemplar of Cochlear_Pos_Noise
Protos could not account for the exemplar feature ar_c(elevated). Please explain how a feature of the new case is equivalent to ar_c(elevated) or type <none>.

```>>>if category is cochlear_pos_noise then
ar_c(normal) sometimes equivalent ar_c(elevated)```

Because all of the exemplar's features are matched, Protos asks the teacher's permission to merge the new case with the existing exemplar, and he approves.

### 4.2.4 A Comment on Explanations

A striking aspect of these examples, and indeed of all of Protos' training in audiology, is the simplicity of the associated explanations. Explanations of featural equivalence were somewhat uncommon; usually features matched directly, or they were not matched. Explanations of the relevance of features to categories tended to express simple correlations. Five factors contributed to the nature of explanations in this experiment:

1. the restricted description of cases.
2. the types of associations recalled by an audiologist during diagnosis.
3. the teacher's mental model of Protos.
4. Protos' explanation language.
5. the range of knowledge available to the system.

First, as discussed earlier, it was decided to restrict input to a small set of standard test results, symptoms, and patient-history information. These
features are disjoint and have standard names. Consequently the only possible explanations involved the equivalence of two values of a single feature in the context of a particular diagnosis. For example, the teacher might be willing to state that "if the category is normal ear then ar_c(elevated) is sometimes equivalent to ar_c(normal)," which would tell Protos that the precise type of acoustic reflex present in the ear opposite the test ear does not really matter as long as one is present.

Second, explanations of the relevance of features to category membership were almost always correlational. Features were probabilistically associated with a particular diagnostic category or one of its generalizations. In situations in which the association was very strong, the expert was occasionally willing to ascribe causality. From this observation, it is tempting to conclude that diagnostic reasoning in this domain does not generally involve deep reasoning about the function of the auditory system. Instead, this knowledge is used in compiled form as associations between findings (bits of evidence) and diagnoses. The diagnostic process consists of heuristically combining evidence to determine the (subjectively) most probable diagnosis. The relationships between features and diagnoses which were elicited by Protos are essentially the same as might be embodied in the rule base of a traditional expert system.

Third, the teacher's mental model of Protos inhibited his expression of complex explanations. He had expectations about how Protos would respond to various types of explanations and formulated his input accordingly. In particular, his general familiarity with Protos' mechanism for generating explanations by heuristic search led him to believe that Protos would not deal satisfactorily with long, complex explanations. Considering the weakness (for the sake of generality) of Protos' heuristics for evaluating explanations, there
was some justification for this belief. A complex explanation often involves both a long sequence of inferences and a branching structure. When Protos does not have heuristics specific to the structures involved, it tends to discriminate against such explanations; it prefers short, linear explanations unless its explanation evaluation heuristics provide specific guidance to the contrary. By providing one or two step correlational explanations, the teacher could be relatively sure of their effects on the system's behavior.

Fourth, the teacher found Protos' explanation language to be difficult to use. Although the explanation language offers a rich vocabulary of relations compared to that of other expert system building tools, it is restrictive in comparison to the natural language in which the teacher normally communicates. Formulation of an explanation for Protos is, of necessity, a two step process. The teacher must conceive the explanation in his own terms and then must translate it into Protos' language. Performance of this knowledge engineering task proved to be difficult for the teacher. He tended to skip the first step and to try to formulate explanations directly in the explanation language. This contributed to the preponderance of correlational explanations and sometimes resulted in complaints that appropriate explanations could not be formulated.

An independent experiment [Ros87] involved manual construction of an audiology expert system using Protos' concept representation. A colleague, Joe Ross, used the same explanation language as Protos but initially obtained all explanations in unrestricted natural language. He then carefully translated them into Protos' restricted vocabulary. The results suggest that the explanation language is adequate to represent almost all of explanations the teacher cared to give. However, this experiment clearly illustrates the complexity inherent in the translation process. Translation may be prohibitively difficult for
the teacher to perform during direct interaction with the system. Also, much time is required for careful translation of explanations. Ross spent as much time on representing 30 cases as the expert did presenting Protos with 200.

Fifth, like other current AI systems, Protos does not have the broad range of knowledge necessary to form truly satisfying explanations, especially in unusual cases. Consider, for example, the case of a poor, elderly southern woman whose ability to hear tones was not impaired, but who had trouble identifying words in a standard test of speech discrimination. Such a test result might suggest a neural problem. However, in this case, the sociocultural context suggests that the result may not be diagnostically significant. The test involves identifying a series of words from a recording of a Midwestern speaker whose speech patterns are very different than those the woman hears in her daily life. Cultural differences could cause her to misunderstand several words. It is difficult to see how a system like Protos could acquire the breadth of knowledge necessary to formulate such explanations and the heuristics necessary to adequately restrict search in such a knowledge base.

In fairness to Protos, it should be noted that it performs very well in spite of these problems. The explanations that it obtains are adequate for compiling reminders and estimating featural importance. Justifying a classification in terms of a recalled exemplar, whose important features are matched directly or by simply explained equivalences and whose unmatched features are claimed to be unimportant, has proven adequate both for Protos’ internal assessment of similarity and for discussion with a user.
4.3 Empirical Evaluation of Protos

Data collected during construction of the audiology expert system allowed detailed evaluation of Protos' performance. Protos recorded every significant event that occurred during its use (e.g., an attempted match or a teacher-provided explanation) for later analysis. Several intermediate versions of the category structure were saved to allow analysis of how the system's knowledge changed over time. This section presents an empirical evaluation of Protos' performance based on these data.\(^8\)

The characteristics of the training and test cases are summarized in Table 4.1. Protos was trained with 200 sequential cases from the files of Baylor Medical School and then was tested with the twenty-six cases that followed the training set in the files. During training, the teacher's interaction with Protos was not restricted. He gave explanations as he saw fit. He also reassessed Protos' indices and the importances of features when asked to do so. Although he would have been free to provide considerable unfocused instruction as well, this was rarely done.

During testing, input to Protos was minimal. The program was restarted for each case to eliminate any learning that might have occurred while processing a previous test case. The tester input a case description and then disapproved matches until an exemplar in the (known) correct category was matched. He provided no additional training such as additional explanations. The objective of this procedure was to assess Protos' performance as a classification system after it had received the allotted training. It should

\(^8\)This experiment should be viewed as exploratory because of its informal nature. For example, extensive discussion occurred between the expert and Protos' designer, and the system was modified during the experiment.
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Training Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cases</td>
<td>200</td>
<td>26</td>
</tr>
<tr>
<td>Number of Categories</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>Exemplars Retained</td>
<td>120</td>
<td>—</td>
</tr>
<tr>
<td>Mean Features/Case</td>
<td>10.6</td>
<td>11.5</td>
</tr>
</tbody>
</table>

Table 4.1: Characteristics of Cases Presented to Protos

be recalled, however, that in normal operation, classification and learning are tightly integrated in Protos.

One way to characterize Protos' learning performance is by the information which it retained and how the rates of retention changed over time. Figure 4.13 illustrates the retention of exemplars as a function of the number of training cases processed. For comparison, the growth of the number of categories is also plotted. During the first 100 cases, exemplar retention was very high: 86 of the first 100 cases were retained. There are two reasons for this high rate of retention. First, Protos was learning 2/3 of its diagnostic categories from these cases, including highly polymorphic categories such as `cochlear_unknown` and `cochlear_age`. Second, Protos' original explanation language did not allow conditional explanations. The teacher proved to be generally unwilling to provide unconditional explanations of featural equivalence. Consequently, Protos was unable to merge cases when important features were not identical.

During a postmortem discussion, the teacher conjectured that the ability to give conditional explanations would have enabled explanations which would have allowed 30 to 40 additional merges. This is consistent with Protos' exemplar retention during the second hundred cases, when conditional explanations were possible. Conditional explanations (combined with the introduction of fewer new categories) allowed Protos to merge 66 of 100 cases.
Figure 4.13: Growth of the Category Structure — Categories and Exemplars
The retention of exemplars by category is reported in Figure 4.14. Protos retained the exemplars of each category that its heuristic criteria determined were necessary to represent the category's significant variation. No claim is made that Protos retained the minimum set of exemplars required for accurate classification. Instead, its conservative criteria for exemplar retention caused it to retain every case whose similarity to an existing exemplar could not be strongly explained. It is, however, possible to use these data to assess the relative polymorphism of categories. For example, a category such as cochlear unknown (Coch.Unk) is highly polymorphic because it is a catchall category for all cases of cochlear hearing loss which cannot be specifically diagnosed. In contrast, a category like fixation is extremely homogeneous because a few features are indicative of fixation, and other features are spurious to the diagnosis.

Another interesting thing to note in Figure 4.14 is the small number of training cases in many of the diagnostic categories. Half of the categories are represented by only one or two cases. If this training set is representative of the training that is typically available in an expert domain, a learning program cannot rely on examination of large numbers of examples to acquire categories. For comparison, the distribution of cases in the test set is presented in Figure 4.15. This distribution is consistent with that of the training set.

Figure 4.16 shows the growth of indexing information in the category structure. Growth of the number of remindings was initially quite rapid. It slowed after Protos had seen the majority of the domain's features in the contexts in which they are diagnostically significant. The slower growth rate during the second 100 cases was also influenced by the teacher's decision to be more selective in giving explanations relating features to categories (from which
Figure 4.14: Retention of Exemplars by Category
Figure 4.15: Distribution of Test Cases by Category
reminders are compiled). Growth of the number of difference links showed a similar pattern. During early training, many "near miss" matches resulted in the addition of difference links. As the system's knowledge grew and its indices were refined, such near misses became less frequent. Also, the addition of the censor mechanism eliminated many near misses, reducing the need for intercategory difference links.

Figure 4.17 illustrates that Protos' general domain knowledge, as embodied in explanation links in the category structure, grew uniformly throughout training. As Protos' knowledge grew, however, the nature of the explanations requested from the teacher changed. Figure 4.18 indicates a gradual shift from feature-to-category explanations to feature-to-feature explanations which occurred as learning shifted from acquiring an adequate set of exemplars to
acquiring the domain knowledge necessary to recognize the similarity of cases. The rising trend in feature-to-feature explanations is also visible in the steady increase in the number of conditional explanation links (Figure 4.17), which are only associated with feature-to-feature explanations.

Figure 4.19 characterizes the types of teacher-provided explanations by showing the relative frequencies of various types of relational links in the final category structure. As discussed in Section 4.2.4, explanations were generally correlational. Although no breakdown is available to illustrate changes over time, observation suggested that during the later period of training, the teacher became more comfortable with giving causal explanations, and he presented increasingly many relationships in these terms.
Figure 4.18: Teacher-Provided Explanations per Case
Figure 4.19: The Types of Explanation Links in the Category Structure
The most fundamental assessment of Protos’ performance is the accuracy of its classifications. Table 4.2 presents the percentages of correct classifications resulting from the first, strongest, and other matches found. A classification was considered to be correct when our expert, Dr. Wier, and the Baylor clinicians agreed on the diagnosis of the case. Protos performed quite well on both the training and test sets. On average, its strongest match produced a correct classification more than 80% of the time. Protos’ misclassifications were not random errors; its strongest classifications were almost always plausible. When one was rejected, it was generally because Protos classified a case into a more specific category than the expert felt the evidence permitted (e.g., Example Case 2).

Because Protos does not distinguish between learning and problem solving, it is reasonable to discuss its classification performance during training. The classification results for each group of 50 training cases can be viewed as a test of previous learning. These results should be taken as a lower bound on Protos’ performance because additional learning occurred in parallel with classification.

In addition to considering Protos’ accuracy, it is important to consider its efficiency. Table 4.3 reports the number of hypotheses (combined reminders) that Protos had to pursue to determine a correct classification. The mean, median, 90th., and 95th. percentiles are reported. The mean and the

---

9Protos’ ability to perform heuristic classification (classifying cases into known categories) is being reported. The 24 training cases which introduced new categories were excluded from these calculations. Also, the fact that Protos correctly classified 100% of the test cases should not be taken as an indication that Protos has learned the corresponding categories “in the limit.” It simply demonstrates that the system’s performance is very good on a representative set of test cases.

10The mean is commonly reported in conjunction with the standard deviation to allow
<table>
<thead>
<tr>
<th>Cases</th>
<th>1st. Correct</th>
<th>Strongest Correct</th>
<th>Any Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-50</td>
<td>55.0</td>
<td>82.5</td>
<td>87.5</td>
</tr>
<tr>
<td>51-100</td>
<td>72.7</td>
<td>93.2</td>
<td>100.0</td>
</tr>
<tr>
<td>101-150</td>
<td>59.1</td>
<td>81.8</td>
<td>86.4</td>
</tr>
<tr>
<td>151-200</td>
<td>44.7</td>
<td>70.2</td>
<td>83.0</td>
</tr>
<tr>
<td>1-200</td>
<td>57.7</td>
<td>81.7</td>
<td>89.1</td>
</tr>
<tr>
<td>test</td>
<td>92.3</td>
<td>100.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 4.2: Percentage of Correct Classifications

<table>
<thead>
<tr>
<th>Cases</th>
<th>Mean</th>
<th>Median</th>
<th>90th. Percentile</th>
<th>95th. Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-50</td>
<td>2.7</td>
<td>2.2</td>
<td>3.8</td>
<td>4.0</td>
</tr>
<tr>
<td>51-100</td>
<td>2.8</td>
<td>2.0</td>
<td>5.3</td>
<td>5.8</td>
</tr>
<tr>
<td>101-150</td>
<td>2.5</td>
<td>2.1</td>
<td>3.8</td>
<td>4.8</td>
</tr>
<tr>
<td>151-200</td>
<td>4.0</td>
<td>2.4</td>
<td>8.1</td>
<td>11.7</td>
</tr>
<tr>
<td>1-200</td>
<td>3.0</td>
<td>3.1</td>
<td>5.2</td>
<td>6.2</td>
</tr>
<tr>
<td>test</td>
<td>3.7</td>
<td>3.0</td>
<td>4.4</td>
<td>4.7</td>
</tr>
</tbody>
</table>

Table 4.3: Number of Remindings Pursued to Find a Correct Match

median represent average performance, while the 90th. and 95th. percentiles aid in assessment of the system's worst case performance. For example, in the training set as a whole, 95% of the cases (which could be correctly classified) were classified by considering exemplars from 6.2 or fewer categories. The substantial increase that occurred during training cases 151-200 was largely due to a high number of atypical cases in that group. The figures for the test set (cases 201-226 from Baylor’s files) suggest that the increase was not indicative of a general upward trend.

These statistics show a gradual increase in the mean number of hypotheses pursued as the size of the category structure increased. However, as

assessment of the variability of the data's distribution. However, given the skewness of these data, the standard deviation is difficult to interpret, so some percentile ranks are reported instead. The median, 90th., and 95th. percentiles represent the points below which 50%, 90%, and 95% of the cases fall respectively.
Table 4.4: Number of Matches Attempted to Find a Correct Match

<table>
<thead>
<tr>
<th>Cases</th>
<th>Mean</th>
<th>Median</th>
<th>90th. Percentile</th>
<th>95th. Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>101-150</td>
<td>4.6</td>
<td>3.1</td>
<td>7.3</td>
<td>13.1</td>
</tr>
<tr>
<td>151-200</td>
<td>7.4</td>
<td>5.2</td>
<td>15.1</td>
<td>17.5</td>
</tr>
<tr>
<td>101-200</td>
<td>6.0</td>
<td>4.0</td>
<td>13.2</td>
<td>17.1</td>
</tr>
<tr>
<td>test</td>
<td>5.3</td>
<td>4.0</td>
<td>6.5</td>
<td>6.9</td>
</tr>
</tbody>
</table>

size increased, the number of possible hypotheses also increased. The percentage of hypotheses (i.e., exemplar-containing categories) tried by Protos was fairly constant as the category structure grew.

Table 4.4 reports on the number of matches attempted in reaching correct classifications. Again, the substantial increase that occurred during training cases 151-200 was due to the unusual nature of the final 50 training cases, which included many atypical cases. Since the search for a matching exemplar within a category was largely guided by prototypicality, several matches had to be attempted en route to acceptable matches for these atypical cases. Again, the figures for the test set suggest that the increase was not indicative of a general upward trend.

Table 4.5 shows that prototypicality was the major determinant of which exemplars were tried during the matching process. After the reminding process provided a hypothesis, exemplars of that category were selected for matching almost exclusively in order of prototypicality. Difference links only occasionally offered shortcuts during the matching process. No trend was evident indicating that the role of difference links increased as the category structure grew.

---

11 This information is only available for the second 100 cases of the training set and for the test set.
12 The reminders compiled during the experiment were exclusively associated with
<table>
<thead>
<tr>
<th>Cases</th>
<th>Prototypicality</th>
<th>Difference</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>101-150</td>
<td>83.3</td>
<td>16.7</td>
<td></td>
</tr>
<tr>
<td>151-200</td>
<td>89.7</td>
<td>10.3</td>
<td></td>
</tr>
<tr>
<td>101-200</td>
<td>86.7</td>
<td>13.3</td>
<td></td>
</tr>
<tr>
<td>test</td>
<td>80.8</td>
<td>19.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.5: Source of Correct Match

<table>
<thead>
<tr>
<th>Cases</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-50</td>
<td>1.7</td>
</tr>
<tr>
<td>51-100</td>
<td>1.6</td>
</tr>
<tr>
<td>101-150</td>
<td>1.5</td>
</tr>
<tr>
<td>151-200</td>
<td>1.9</td>
</tr>
<tr>
<td>1-200</td>
<td>1.6</td>
</tr>
<tr>
<td>test</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table 4.6: Average Number of Matches Discussed With Teacher

Another way to assess Protos’ problem solving efficiency is to consider how much of its effort is independent of the teacher. Table 4.6 reports the average number of matches per case that Protos discussed with the teacher. The bulk of Protos’ problem solving effort was independent of the teacher. It typically tried several matches and rejected almost all incorrect ones on its own. The system’s performance was fairly constant across the training and test sets.

4.3.1 Human Classification Performance in Audiology

Nineteen human subjects were asked to classify the audiology test cases to provide a standard for evaluating Protos’ classification performance. The subjects were recruited from the Department of Speech and Hearing of the University of Texas. Two were the clinicians, who supervise the department’s
<table>
<thead>
<tr>
<th>Subjects</th>
<th>Preferred Class. Correct</th>
<th>Any Class. Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinician 1</td>
<td>69%</td>
<td>81%</td>
</tr>
<tr>
<td>Clinician 2</td>
<td>85%</td>
<td>92%</td>
</tr>
<tr>
<td>Student (mean)</td>
<td>69%</td>
<td>73%</td>
</tr>
</tbody>
</table>

Table 4.7: Correct Classifications by Human Subjects

clinical practicum. They are considered to be expert diagnosticians. The other 17 were graduate students, who had more than one year of clinical experience.

The 26 cases were presented in a questionnaire. Each case was described on a page, accompanied by a list of the 24 possible diagnoses. The test taker was instructed to consider only these possibilities. If the features suggested a single diagnosis, it was to be indicated by a check-mark. If multiple diagnoses were possible, they were to be ranked by numbering them in order plausibility. The cases were described by the same features that the expert presented to Protos. However, unlike the presentation to Protos, feature names were not abbreviated (e.g., *mild-to-moderate sensorineural* was reported instead of *m.m.sn*).

The results of the experiment are presented in Table 4.7. The clinicians' results are reported separately and the students' results are averaged. The left column of the table reports the percentage of cases in which the correct diagnosis was chosen as the first (or only) choice. The right column reports the percentage in which the correct diagnosis was specified as any choice.

---

13 A sample page of the questionnaire and the instructions to the subjects are presented in Appendix E.

14 Again, a diagnosis was considered to be correct when our expert and the Baylor clinicians agreed on the case.
4.4 Discussion of the Experimental Results

The statistics presented in the previous section attempt to characterize Protos as a learner and problem solver in the domain of clinical audiology. There are many encouraging results:

- Protos learned to perform expertly. Its classification performance on a set of test cases was comparable to that of experienced human clinicians.\(^\text{15}\)

- The rate of growth of Protos’ category structure slowed over time, suggesting that the size of Protos’ required knowledge base should not be of concern in domains with the same general characteristics as audiology.\(^\text{16}\)

- Although, Protos’ classification effort increased as the size of the category structure increased, its performance did not degrade unacceptably. Most of its effort was independent of the teacher.

- The teacher’s training effort diminished as Protos acquired knowledge. The nature of his interaction with Protos shifted from providing assistance in integrating new exemplars into the category structure to providing key bits of domain knowledge that allowed Protos to recognize similarities between cases.

---

\(^{15}\)The comparative results must be evaluated cautiously because the experimental task was somewhat unfamiliar to the human subjects. They have been trained to perform diagnosis by direct observation of a patient or from a set of "raw" test results accompanied by a complete patient history. There were complaints that some cases were inadequately described by the data presented.

\(^{16}\)The size of Protos’ category structure is determined by the number of exemplar-containing categories and their polymorphism. As characterized by our expert, the domain of audiology was covered by a reasonable number of categories (approximately 35 of which Protos saw instances of 24). The expert was able to provide relatively simple domain knowledge that enabled Protos to recognize the similarities of cases and hence to retain them selectively as exemplars.
It must be stressed that the evaluation of Protos' performance in the
domain of audiology does not constitute a complete validation of the Protos
approach. The results are relevant to other heuristic classification tasks only
in so far as audiology is representative of this class of tasks. For successful ap-
plication, Protos requires an expert teacher who can present the system with
featural descriptions of cases\(^{17}\) and explain their classifications. Further exper-
imentation in other domains and with other teachers is necessary to determine
when this approach can best be applied.

This evaluation assessed the performance of the Protos system as a
whole. No attempt was made to analyze the performance of its components
selectively to determine the system's sources of power and hence to provide
justification for all of Protos' heuristic mechanisms. Further experimentation
is necessary to determine which design decisions (if any) were not justified.

Protos did, however, prove to be a practical tool. It successfully
learned to perform an expert classification task without requiring unreasonable
training or expending unreasonable effort. By doing so, Protos made effective
contributions to the fields of machine learning and automated knowledge ac-
quisition.

\(^{17}\)Issues of diagnostic strategy (i.e., the acquisition of features) are handled by the teacher.
In audiological diagnosis, a complete patient history is routinely elicited, and a large number
of routine tests are routinely performed. Protos was initially provided with all of the features
that it needed to classify a case. In other domains, however, the cost (or risk) of obtaining
information may increase the importance of strategy in the diagnostic process, placing a
greater burden on users of Protos.
Chapter 5

Related Research

This chapter compares and contrasts Protos with the work of others. Given the broad range of issues addressed by the Protos project, many topics could be included. For example, this chapter could survey human and artificial knowledge representation, heuristic classification, as well as machine learning for concept acquisition. Such a broad survey would, of necessity, be superficial. Instead, this chapter presents detailed discussions of case-based reasoning and automated knowledge acquisition. Work in these areas is most closely related to Protos' approach and application and is therefore most relevant to assessing the contributions made by the Protos project.

A chapter on related research is often simply a catalog of other projects. Rather than following this approach, Chapter 5 identifies relevant characteristics for comparing systems and focuses subsequent discussion on them. Every system surveyed is not discussed with respect to each of the issues; instead, systems are discussed only when the researchers addressed the issue being considered.

5.1 Case-Based Reasoning

Case-based reasoning is the usual name given to problem solving methods which make use of specific past experiences rather than a corpus of
general knowledge. The past experiences (cases) used in the reasoning process may be acquired by the system itself, or they may be provided by the system's implementor. The key distinction between case-based methods and other forms of automated reasoning is that in the former, a new problem is solved by recognizing its similarity to a specific known problem, then transferring the solution of the known problem to the new one. In contrast, other methods of problem solving derive a solution either from a general characterization of a group of problems or by search through a still more general body of knowledge.

Research has demonstrated that many types of tasks are amenable to case-based reasoning. Protos embodies a case-based technique for classification. In addition to classification (see also, [KD87,KK85]), case-based reasoning has been applied to planning [Kol87,Sim85,Ham86], legal reasoning [RA87,Bai86], question answering [Kol83a,Kol83b], speech recognition [Bra87], and word pronunciation [SW86,Leh87].

This section discusses representative research in case-based reasoning and contrasts it with Protos. The major research efforts in the field are introduced. Then, six fundamental issues of case-based reasoning are identified, and the research which has addressed each issue is discussed.

5.1.1 An Overview of Case-Based Reasoning Research

Kolodner was one of the first researchers to implement a case-based reasoning system. Her CYRUS system [Kol83a,Kol83b] was a question answering system based on Schank's model of Dynamic Memory [Sch82]. When a question was posed to CYRUS, it would extract (or infer) indices which would allow it to access a knowledge base of past events to retrieve the appropriate case for answering the question. CYRUS could learn incrementally as new
cases were presented. Learning and question answering in CYRUS employed a uniform mechanism for traversing an indexing structure and finding an appropriate location for adding a new case or retrieving a previous one.

This case-based model of memory was next applied to psychiatric diagnosis in Kolodner's SHRINK system [KK85]. SHRINK assigned a patient to a diagnostic category by retrieving the most similar previous case and applying the associated classification. This work introduced the idea of indexing cases by previous diagnostic failures as well as by observable symptoms. Unfortunately, the interesting issues raised by this work were not explored in the context of a substantial implementation.

Kolodner next applied case-based reasoning to meal planning in the JULIA system [Kol87]. Given the nature of the guests and their dietary preferences, JULIA used descriptions of previous meals to satisfy constraints in planning a new meal. Here, case-based reasoning functioned as a component of a more complex, integrated problem solving system. The case-based reasoner ran in conjunction with a problem reduction problem solver and constraint satisfier which provided goals to direct the retrieval of cases.

Simpson extended Kolodner's earlier research on memory organization and case-based reasoning by applying the techniques in the domain of dispute mediation. His MEDIATOR system [Sim85] formulated plans for settling disputes by recalling past disputes and adapting their solutions to the new situations.

Simpson's work is notable in that it identified the multiple roles that case-based reasoning can play in problem solving. The first role is problem understanding. A problem description is categorized by recalling a similar past case. If the description of the new problem is incomplete, additional infor-
ination can be inferred from the recalled case and the norms of its associated category. The second role is the formulation of a solution to the new problem. The solution to the recalled case can be modified and proposed as a solution to the new problem. The third role is failure recovery. If the proposed solution fails, a similar previous failure can be recalled to suggest a remedy for the failure. The fourth role is learning. The result of solving the new problem can be stored with appropriate indices so that it is available for solving future problems, potentially saving substantial problem solving effort.

Rissland and Ashley applied case-based reasoning to legal argument in the area of trade secrets law (e.g., [RA87]). Given a factual situation, their HYPO system identified the "most on point" (i.e., most similar on relevant dimensions) cases for a legal argument. Once a case was retrieved, its facts could be perturbed to create hypothetical cases for further argument by systematically varying aspects of the case description. The HYPO system differs from previously discussed systems in that it employed a fixed set of indices and cases and did not learn in the course of problem solving.

Kibler and Aha [KD87] compared the effectiveness of various strategies for acquiring a knowledge base of cases for performing classification. They identified three general approaches, which were applied to a database of thyroid disease patients for comparison. This work only addressed the issue of selecting which cases to retain in order to maximize classification accuracy; indexing issues were not explored.

Bradshaw's NEXUS system [Bra87] took a case-based approach to learning speech recognition. He identified three learning mechanisms: one for adding new cases in response to recognition failures, one for generalizing existing cases in response to partial successes, and one for removing cases which
did not prove to be useful in the recognition process.

Stanfill and Waltz [SW86] studied a case-based approach to determining word pronunciation implemented on the Connection Machine. This is the only work to date to exploit parallelism in the processes of case retrieval and matching. However, it did not address learning issues.

5.1.2 Issues in Case-Based Reasoning

Although these applications of case-based reasoning are quite different, all share a common set of issues, which can be divided into two broad categories — those concerned with finding a matching case and those involving acquisition of the necessary knowledge base of cases. The process of finding a matching case provides two characteristics for comparing case-based reasoning systems:

- how cases are indexed for efficient retrieval.
- how the similarity between a new problem and a retrieved case is assessed.

Acquiring a knowledge base of cases provides the following additional characteristics for comparison:

- how cases are selected for retention.
- how indexing information is learned.
- how any additional domain knowledge required for the assessment of similarity is acquired.
- how generalization (if any) occurs during learning.
Not every case-based reasoning system introduced in the previous section addressed all of these issues. Consequently, the following sections will only discuss a system with respect to an issue if the issue was addressed as a fundamental part of the research. It is argued that Protos is the only system which integrates viable solutions to all of these fundamental problems of case-based reasoning.

5.1.3 Indexing Cases for Efficient Retrieval

When presented with a new case, a case-based reasoner must locate a case which is similar to the new one. Some researchers (e.g., [KD87, SW86]) have assumed the knowledge base is small enough to attempt matching the new problem against all stored cases. However, if there are many cases or if the matching process is computationally expensive, this approach may be intractable. In general, a case-based reasoner needs indexing mechanisms to reduce search, and researchers have explored a variety of approaches.

In all of Kolodner's systems, knowledge is organized as a hierarchy of generalized norms (maximal conjunctive generalizations). The search for a matching case begins at a very general norm (called an E-MOP in CYRUS [Kol83a]). Indices record featural differences of past cases from the norm. One of these indices, which matches a feature of the new problem, is selected for traversal (by an unspecified process). The index selection and traversal process continues recursively until a leaf of the memory structure (i.e., a case or a norm with no successors) is reached or until no features of the new problem match possible indices. In the latter situation, a static body of background knowledge may be used to transform the problem's features or to derive new features which allow the process to continue [Kol83b]. If the traversal process fails to retrieve
an appropriate case, presumably the system backtracks and makes another indexing choice at a higher level.

Every featural difference is a potential index; each known case can be retrieved via many different paths. Although this indexing scheme is presented as a way to maintain efficient case retrieval as the size of the case knowledge base grows, it can also impose severe search and memory burdens on the system. The potential exists for explosive growth of indices as the size of the knowledge base increases, and search through all of the possible combinations of indices may prove intractable. In CYRUS, this problem is avoided by allowing only a small vocabulary of features [Kol83b].

Kolodner’s indexing scheme is designed to locate cases within a single general category (represented by a top-level E-MOP). In some domains, however, several top-level hypotheses may be possible. The process of choosing a general hypothesis based upon features of a new case is still an open research issue [KK85].

Simpson's MEDIATOR system builds upon Kolodner's implementation. Like Kolodner, he avoids the search problems which result from a large number of possible indices by greatly restricting the featural vocabulary for case descriptions. This restriction also allows him to circumvent the problem of choosing a top-level E-MOP at which to begin traversal. In addition to the hierarchy of norms and cases, MEDIATOR has a set of orthogonal indexing structures corresponding to each type of feature. When a new case is presented, MEDIATOR traverses each of these structures as deeply as possible and returns all of the most specific cases (or norms) encountered. If multiple

\footnote{This issue is discussed further in Section 5.1.6 on learning indexing.}
cases are retrieved, it uses a heuristic procedure to select the most similar case. In the event that the retrieval process returns a norm instead of a case (i.e., the traversal of the indexing structures does not lead to a unique case because inadequate featural information is available), a precedent case, preassociated with the norm, is retrieved.

Indexing in MEDIATOR (and in Kolodner’s SHRINK and JULIA) uses more than just features of the new case. Cases in the knowledge base may also be indexed by past problem solving failures in which they were involved. These failure indices are chosen from a fixed enumeration of types of failures (e.g., “wrong goals inferred for disputants”) for installation between a case that was incorrectly retrieved and the case that is truly most relevant to the problem. Initially, case retrieval is based solely on the features of the new case. If the plan proposed by the retrieved case is rejected by a human oracle, the reason for the failure is determined by a combination of question asking and heuristic inference. After the failure is classified, its classification is used in a secondary indexing process to retrieve a case which rectified a similar failure during past problem solving.

In Rissland and Ashley’s HYPO system, cases are indexed by a set of dimensions, which “are compiled knowledge from the case law; they relate legally operative facts to decisions [RA87].” These dimensions were drawn from scholarly analyses of legal domains rather than identified by the HYPO’s implementors. A dimension is a complex structure consisting of:

---

2 This procedure is discussed further in Section 5.1.4 on the assessment of similarity between cases.

3 The acceptability of a case is determined by feedback from MEDIATOR’s user. If the user rejects a case, MEDIATOR interacts to determine the cause of the failure and then examines failure indices associated with the case to attempt to locate a more appropriate one.
• a list of factual predicates which must or might be true of the case if the dimension is applicable.

• a list of the claims for which the dimension is significant.

• a list of cases indexed by the dimension.

• specifications of how to perturb the values of the factual predicates to create hypothetical cases. This captures knowledge of what makes a legal situation better or worse along a given dimension.

When a new case is presented to HYPO, domain knowledge is used to determine which factual predicates hold. The predicates are used to identify applicable and near-miss dimensions. An applicable dimension is one for which all required factual predicates hold; a near miss is one in which only some hold. The selected dimensions, in turn, index cases. All cases indexed by the identified dimensions are retrieved. Unlike other systems, the emphasis in HYPO is not on retrieving a single “best” case. Rather HYPO retrieves all cases which may be relevant to the legal argument at hand.

HYPO’s ability to retrieve relevant cases is largely dependent on the system’s ability to determine which factual predicates are relevant to a new case. As the authors point out, this can be a difficult process. The applicability of a legal predicate to a case may, itself, be central to the legal debate about a case. Determining the factual predicates relevant to a case description may be the most complex part of legal reasoning.

The applicability of HYPO’s indexing mechanism to a domain depends on whether that domain can be characterized in terms of a small set of pre-enumerated dimensions. Before the mechanism can be applied, this set of
dimensions must be determined, and mappings from case facts to dimensions and from dimensions to known cases must be specified.

Protos employs a richer indexing scheme than these systems. It makes use of multiple indexing mechanisms which correspond to distinct stages in the search for a matching case (i.e., exemplar). The different search goals of each stage necessitate the use of different types of indices. First, the system must hypothesize a classification based on the features of a new case. Second, it must choose the exemplar of the hypothesized classification that is most likely to match a new instance. Finally, it must identify and avoid reocurrences of previous classification failures. Protos' remindings, prototypicality ratings, and difference links, respectively, correspond to these three stages in the search for a matching exemplar.

Remindings are compiled associations between case features and classifications. Unlike Kolodner's approach, this addresses the problem of initial hypothesis formation by associating hypotheses directly with some case features. Furthermore, there is no need to restrict the featural vocabulary to avoid combinatorial explosion of indices; remindings result only when case features have explainable relevance to classifications. Indexing via remindings compiles the two-step indexing process that occurs in HYPO. Compilation makes the indexing process more efficient.4

Although Protos' remindings may point to specific cases, they most often indicate categories from which the most similar case may be chosen.

---

4It should be noted, however, that HYPO uses intermediate information (that maps features to factual predicates and factual predicates to dimensions) for other purposes, such as the generation of hypothetical cases, which were not considered relevant to the classification task performed by Protos.
Within a category, Protos relies on prototypicality to provide a partial ordering of cases. The most prototypical exemplars of a category are considered most similar to other category members, so they are tried first in the subsequent matching process. Prototypicality provides a less structured mechanism for selecting cases within a category than Kolodner or Simpson's hierarchical indexing structures. Such hierarchical structures may provide a more efficient means of finding the best match for an outlying case. However, because such cases are expected to be rare, the additional indexing overhead may be unnecessary.

Protos' difference links record past problem solving failures like the failure indices used in Kolodner and Simpson's systems. A difference link records case features which are relevant in determining which of two stored exemplars is the best match for a new case. This differs from their failure indices which index cases in terms of the types of problem solving failures in which they were involved. A Protos difference link is used directly during the search for a matching case without the intermediate step of determining if a failure of a given type occurred. Such a determination can involve much inferential effort. Maintaining this information in uncompiled form enables more efficient failure-based indexing in Protos.

5.1.4 The Assessment of Similarity

After a case has been retrieved from the knowledge base, its similarity to the new case must be assessed. This is especially important given that most systems retrieve several cases which must be ranked in order of similarity to the new case or relevance to the problem solving task at hand.

In Kolodner's systems, the assessment of similarity is combined with
the indexing process. These systems retrieve only the first case encountered during traversal of the index structure. This case is known to be similar to the new case because the cases match on the indexing features. These featural matches, however, may not be direct because features of the new case may have been transformed (using a static body of background knowledge) to determine indices. The occurrence of featural transformations does not adversely affect the assessment of similarity; matches involving transformations have the same status as direct matches.

Should the retrieved case prove inadequate for problem solving, the SHRINK and JULIA systems make use of secondary indexing based on failures (as discussed earlier). When a case is retrieved via such a link, it is assumed to have greater similarity to the new case being processed because of its past involvement in a similar failure.

Since the MEDIATOR system pursues all possible indexing paths, it often retrieves multiple cases. In such situations, a heuristic procedure ranks the retrieved cases according to their assessed similarity to the new case. This procedure first eliminates all cases in which the most important features (e.g., the disputants' goals) are not identical to the goals in the new case. The remaining cases are rated according to number of matches on a subset of (pre-enumerated) features believed to be most important. If a choice cannot be made at this point, a case is selected according to a fixed preference scheme. Like Kolodner's systems, MEDIATOR sometimes employs transformational matching of case features to indices. During the process of assessing featural matches, the number and types of such transformations affect the similarity rankings. Like SHRINK and JULIA, MEDIATOR also makes use of a secondary indexing procedure based on problem solving failures.
Unlike the other systems discussed here, the aim of Rissland and Ashley's HYPO system is not to select a single closest match case but rather to collect all similar cases. HYPO retrieves all cases that match or nearly match the new case on any underlying dimensions. The cases that have dimensions in support of the position being argued and none in support of the opposite position are considered to be "most on point."

The goal of Stanfill and Waltz's system is to assign a classification rather than to select the case that most closely matches a new case. Classification is not generally determined by similarity to one case. Their system assesses the similarity of every case in the knowledge base to the new case using a complex distance metric. All cases within a threshold distance (i.e., dissimilarity) are retrieved. If all of their classifications agree, the new case is similarly classified. Otherwise, a total dissimilarity score is computed for the retrieved cases of each category (with respect to the new case), and the classification with the least overall dissimilarity is chosen. Similarity is assessed on the basis of syntactic matching of features, which necessitates a uniform representation of cases.

Protos assesses the similarity of a retrieved exemplar to a new case by producing an explanation of their similarity. The explanation process attempts to identify a case feature which corresponds to each exemplar feature and to construct a chain of inferences which explains their equivalence. The importances of unmatched exemplar features are determined by evaluating explanations of their relevance to the exemplar's category. The overall quality of the explanation of case and exemplar similarity is heuristically assessed. If the similarity is judged adequate, the explanation is presented to the user in symbolic terms.
Kedar-Cabelli's work on purpose-directed analogy [Ked85] also makes use of explanations to assess the similarity of objects. If a functional explanation of how the features of a known object allow it to satisfy a known purpose can be transferred to the features of a new object, the objects are considered to be equivalent. This work is similar in spirit to Protos' assessment of similarity but differs in two important ways. First, Kedar-Cabelli's system regards explanation as formal proof whereas Protos simply regards it as plausible inference. Second, her system is constrained to produce only homogeneous, functional explanations. Protos, on the other hand, can produce heterogeneous explanations which combine many types of inferences into a single explanation. The former approach is potentially overconstrained, while the latter approach has sometimes proven to be underconstrained.

\subsection{The Retention of Cases}

The primary learning mechanism of case-based systems is selective retention of cases. If a case-based system is in an environment which provides a large variety of cases, it must decide which ones to retain for future problem solving.

Kibler and Aha compared the effectiveness of three strategies for case retention: the proximity, growth, and shrink algorithms. The proximity algorithm simply retains all training cases. The growth algorithm retains all cases which cannot be correctly classified by matching to previously retained cases. The shrink algorithm collects all cases and then discards those that would be correctly classified by other cases. The three algorithms were compared empirically using cases drawn from a database of thyroid disease patients. All three of these simple algorithms performed well, producing correct classifications with
better than 70% accuracy. The authors' preliminary analysis suggested that the growth algorithm maximized classification accuracy while minimizing storage requirements.

Kolodner and Simpson's systems are based on a memory model intended to retain all unique cases. A new case is retained unless it matches an existing case on all features (not just the current indexing features). Their work has focused on problem solving using small knowledge bases of cases. For example, Simpson noted that, at its largest, MEDIATOR's knowledge base contained nine cases [Sim85]. They have not worked at a scale which has forced consideration of the issue of selective retention of cases.

Bradshaw's NEXUS system has two mechanisms for determining which cases are retained. One is the seeding mechanism which learns from failures to correctly classify speech patterns. When a pattern is misclassified, it is retained ("seeded into the concept space"). This corresponds to Kibler and Aha's growth algorithm. The second mechanism is pruning. Periodically, NEXUS examines its knowledge base and eliminates cases which have not been used in matching. This corresponds to the shrink algorithm.

Like NEXUS, Protons employs a variant of the growth algorithm. A new case is retained as an exemplar only when its similarity to an existing exemplar cannot be adequately explained. Because unmatched features which are believed to be of low importance do not adversely affect the assessment of similarity, only the presumed relevant features of a case are involved in the

---

It should be noted that their data involved only two classifications. Cases were distributed 90%:10% in one experiment and 95%:5% in another. Overall accuracy could have been improved by always guessing the dominate classification, however, this is not a reasonable strategy when the classifications differ in importance (e.g., normal versus serious illness).
decision to retain or discard it. However, since Protos takes a conservative approach to merging cases, it is likely to retain a higher number of exemplars than Kibler and Aha's approach. Protos' performance is also dependent upon domain knowledge (in the form of explanations) provided by its teacher.

5.1.6 Learning Indexing

As the size of a system's knowledge base of cases grows, the cost of unguided search for a matching case quickly becomes prohibitive. Therefore, as new cases are added, the acquisition of indexing information must be included as an integral part of the learning process.

Kolodner's approach to retaining a new case locates the place that a new case should be installed in the system's indexing structure based on the case's features. The case is added at the deepest point reached by the index traversal process. If that point is a generalized norm, the new case is indexed from the norm by all of its featural differences from the norm. If the deepest point is a case, a new norm is created, which is a similarity-based generalization of the two cases, and both cases are indexed from the norm by their unique features.

Kolodner notes the potential for explosive growth of indices and states that only predictive case features (features which are correlated with and hence predict other case features) should serve as indices [Kol83a]. When a new case is retained, only features which are believed to be predictive because of past experience (or initial information encoded into the system) are chosen as indices. CYRUS uses an empirical approach to assess the predictiveness of its indices. The success of such an approach in avoiding explosive growth of indices is dependent upon processing an adequate number of cases (or sufficient initial
knowledge) to make reasonable empirical assessments of predictiveness. In learning tasks characterized by many possible categories relative to the number of cases (e.g., the audiology domain), there will not be enough data available to assess the system's indices, and hence this approach is not suitable.

Kolodner also notes that multiple indexing paths can lead to a case in memory. This suggests that when a new case is added, every possible indexing path must be traversed and revised to add the new case.

MEDIATOR employs the same indexing mechanisms as Kolodner's systems. When a new case is added to MEDIATOR's knowledge base, every indexing structure is traversed, and the case is added to each one. This approach is computationally possible because of the small featural vocabulary used to describe cases (i.e., there are very few possible indices). Simpson notes, however, that the process of adding a case to memory is more than two orders of magnitude slower than case retrieval [Sim85]. Unlike Kolodner, he makes no attempt to remove indices by assessing their predictiveness over time.

When problem solving failures occur, MEDIATOR learns failure indices, which index cases by the types of problem solving failures in which they are involved. When the system retrieves a case that fails to solve the problem posed by a new case, heuristic rules are used to classify the failure (e.g., "wrong goal inferred for disputants"), and this failure classification is installed as an index between the two cases.\(^6\) SHRINK and JULIA acquire failure-based indices in a similar manner.

Protos learns reminders by analyzing teacher-provided explanations of the relevance of case features to the category to which a new case is being

\(^6\)This process is not clearly described in [Sim85].
added. Reminders are only learned for features which have strong, explainable relationships to the classification of a case. When reminders suggest an incorrect classification during problem solving, they are analytically reassessed. Protos has knowledge-based criteria (described in Section 3.3.5) for choosing and revising reminders. It does not rely on the weaker mechanisms using syntactic featural differences or empirical reassessment. However, this limits Protos' applicability to domains in which the appropriate domain knowledge can be acquired.

Protos empirically learns prototypicality, which is used for intracategory indexing. When an exemplar is involved in a close match to a new case, its prototypicality rating is increased.

Protos also learns failure indices in response to problem solving failures. When a retrieved exemplar is rejected as incorrect by its teacher, Protos installs difference links which record the important featural differences between the new case and the retrieved exemplar. Unlike Simpson's approach to deriving these indices heuristically, Protos obtains them directly by asking its teacher. The heuristic approach was rejected because it was not believed that the appropriate knowledge for "understanding" failures could be made generally available to the system a priori. Likewise, an empirically-based approach to learning difference links was rejected because it was not believed that an adequate amount of training would be available to allow accurate empirical assessment of their relevance.

5.1.7 Learning Additional Domain Knowledge

Among the case-based reasoning systems discussed in this chapter, only Kolodner's systems, Simpson's MEDIATOR, and Protos employ domain
knowledge in addition to a set of cases. This additional knowledge enables these systems to make inferences during problem solving. Koldner's systems have employed domain knowledge to infer missing indices. MEDIATOR has used knowledge to predict missing features and to recognize the similarity of differing case descriptions. Protos employed knowledge to compile indices and to generate explanations of featural equivalence. Case-based reasoning systems which do not have the capabilities to make such inferences must rely upon a uniform representation that completely describes cases and represents their commonalities explicitly.

Only Protos acquires this domain knowledge in the course of problem solving. Kolodner's systems and MEDIATOR rely on static sets of inference rules which are preprogrammed. In contrast, Protos provides an input channel through which its teacher can provide domain knowledge during discussion of problem solving failures. When a case is incorrectly classified, Protos elicits explanations relating its features to its correct classification. When the similarity of two cases is inadequately explained, it elicits explanations of featural equivalence. Because its domain knowledge grows, Protos' abilities to index cases accurately and to explain the similarity of cases improves in the course of its use.

5.1.8 Generalization During Learning

The idea of generalization may seem incompatible with case-based reasoning, but without some form of generalization, the training received by a system would not be transferable to new problems. Case-based reasoning systems must therefore generalize their training to solve new problems.

In Kolodner and Simpson's systems, the process of learning indices is
a generalization process. Not all features of a new case become indices. The features chosen as indices are the result of an (empirically-based) assessment of the predictiveness of case features. During future problem solving a case will be retrieved whenever a new problem shares this subset of its features.

The transformational matching of indices that these systems perform is also a form of generalization. The process, however, relies on a static body of domain knowledge, which fixes the generality of cases.

The indexing structure common to these systems includes a hierarchy of norms, which are similarity-based generalizations covering the cases below them in the indexing structure. If the features of a new problem provide insufficient information to allow the system to locate a case, the most specific, indexed norm is used for problem solving. In such situations, the system attempts to reason from a general problem characterization rather than a specific past case. Recent work in machine learning, however, has suggested that useful generalizations do not always exist in the language in which cases are described [FD86]. Consequently, reasoning from similarity-based norms may be of limited utility.\footnote{It should be noted that this portion of the discussion does not apply to MEDIATOR. Each norm in MEDIATOR's indexing structure has an associated \textit{precedent} case. When index traversal fails to retrieve a specific case, the precedent case of the most specific norm indexed is returned. The associations between norms and precedent cases are provided \textit{a priori}; it is unclear if MEDIATOR can learn additional ones.}

In one set of experiments, Kibler and Aha generalized the cases in their knowledge base by assigning different importance values to features. Rather than learning this information, their system relied on an external oracle to provide the needed information. Features selected as indices by the ID3 learning program [Qui83] were assigned high importance, and other features were con-
- checking the information for consistency and completeness.
- designing an architecture for the system.
- transforming the information provided by the expert into a knowledge base suitable for automated reasoning.
- testing, debugging, and refining the resulting system.

Research in automated knowledge acquisition is aimed at automating the knowledge engineer's role in these tasks.

Because expert systems most frequently perform classification, most research in automated knowledge acquisition has focused on acquiring knowledge for various aspects of classification tasks. Systems in this category have included TEIRESIAS [Dav79], ROGET [Ben85], MORE [KNM85], MOLE [EEMTS87], ETS/AQUINAS [Boo84, BB87], ASK [Gru87], ID3 [Qui83, Qui86], AQ11 [MC80], AQ15 [MMJL86], Meta-Dendral [BF78], and Protos.

As applications of expert systems expanded to nonclassification tasks, research in automated knowledge acquisition has correspondingly addressed a variety of other problem solving tasks, including design (e.g., LEAP [MMS85]), planning (e.g., SALT [MM86], OPAL [MFCS87], and Knac [LL87]), and evaluation (e.g., KNACK [KBGM87,KGM87]).

5.2.1 Issues in Automated Knowledge Acquisition

A unified discussion of automated knowledge acquisition is difficult because research in the area has pursued a variety of goals. To structure this discussion, four general characteristics were selected to provide a framework for identifying and comparing the contributions of each system:
1. the type of assistance that the system provides to its users.

2. the information that the user must provide as input to the system.

3. the results of using the system.

4. the sources of power upon which the system draws.

The topic of automated knowledge acquisition is much broader than the topic of case-based reasoning discussed in the previous section. Because numerous systems have been built, a subset of the systems enumerated earlier was selected which exemplifies the variety of approaches that researchers have taken in addressing the problems of automated knowledge acquisition. The systems chosen are ROGET, ETS, TEIRESIAS, ASK, ID3, and LEAP. Each system is characterized with respect to every characteristic, and brief summaries of Protos' approaches are also provided to facilitate comparison.

5.2.2 The Type of Assistance Provided

Although automated assistance could be provided for any of the knowledge engineering tasks identified earlier, research has focused on providing only a few types of assistance:

- acquisition of the conceptual structure of the problem domain.
- knowledge refinement.
- strategy acquisition.
- autonomous concept acquisition.
- knowledge compilation.

This section will discuss each type of assistance in turn.
Acquisition of Conceptual Structure: When confronted with a knowledge acquisition problem, one of the first tasks is determination of the conceptual structure of the domain. For classification tasks, the expert's knowledge must be characterized in terms of features, categories, and the relationships among them. The acquisition of appropriate vocabulary is an essential first step in constructing the explicit representation of the expert's knowledge that the automated reasoning process requires.

ROGET [Ben85] is designed for this task. It makes use of a set of models abstracted from well-known expert systems that perform heuristic classification. When confronted with a new problem, ROGET interviews the user to select an appropriate general model for guiding the acquisition of the vocabulary of the problem domain. The model suggests types of categories and evidence relevant to reasoning about the domain. ROGET also elicits general relationships between evidence and categories which can subsequently be used to assist in the acquisition of inference rules. However, it does not acquire these rules itself.

ETS [Boo84] (and its successor AQUINAS [BB87]) also acquires the vocabulary for a problem domain. ETS makes use of a general, psychologically-based technique to structure the interview process. It begins by asking the user to enumerate possible classifications and then identifies important domain features by asking the user to discriminate among groups of classifications. After it has obtained these categories and features, the relationship between each category and feature is systematically elicited. Unlike ROGET, ETS processes the information it acquires to create an executable rule base.
Knowledge Refinement: Most work in automated knowledge acquisition has focused on the refinement of existing knowledge bases. The premise of research in knowledge refinement is that it is not difficult to interview an expert to capture his expertise in rough form. The difficult task is refining the resulting rough system so that it performs at a truly expert level. Knowledge elicited from the expert during interviewing is incomplete, inconsistent, and partially erroneous. To produce a system which performs acceptably, such shortcomings must be identified and repaired.

TEIRESIAS, the grandfather of automated knowledge acquisition systems, is a representative knowledge refinement system. TEIRESIAS focuses discussion of a knowledge base on particular problem solving episodes. A test case is presented to the expert system undergoing refinement, and the system produces a classification. If the user questions the result, TEIRESIAS is invoked to determine the reasons for the failure. It guides the user through the chain of inferences which led to the answer. When a problem is found, the user enters a new inference rule or modifies an existing one. The system validates the rule's form by comparing it to an appropriate member of a set of rule models\(^8\) then validates its effect by reprocessing the test case.

The important lesson of TEIRESIAS is that focusing interaction with the user on discussion of a particular domain problem is a powerful technique for knowledge acquisition. Rather than asking an expert simply to regurgitate his knowledge, he is asked to provide information that rectifies a specific problem.

\(^8\)A rule model is a general model representing the structure of rules which infer a particular conclusion. A rule model estimates the likelihood of various antecedents occurring in rules with a particular conclusion, based on empirical analysis of the rule base undergoing refinement. For example, a rule model might note that 80% of the rules concluding the type of a bacterium include shape as an antecedent.
solving failure.

**Strategy Acquisition:** Problem solving expertise consists of factual knowledge of a domain and strategic knowledge of how to apply it to particular problems. Research has concentrated on the acquisition and refinement of factual domain knowledge, and the problem of acquiring strategic domain knowledge has been largely ignored. Problem solving strategy is presumed to be implicit in the (standard) inference mechanism which reasons with the acquired factual knowledge.

An exception is the ASK system [Gru87] which concentrates on eliciting strategy from an expert. The user enters cases represented as sequences of problem solving actions. Each action has an associated justification chosen from a predefined set. ASK interacts with the user to ensure that each action’s justification clearly differentiates it from other possible actions and then generalizes each justification/action pair to produce a strategic rule.

**Autonomous Concept Acquisition:** Alternatives to interactive approaches are provided by machine learning research. A premise of machine learning research is that eliciting knowledge from an expert in appropriate form is prohibitively difficult. The expert cannot directly communicate his expertise; he can only provide examples of its use. A learning program can then process this training set to formulate a knowledge base for expert reasoning. The learning programs most notably applied to knowledge acquisition have been ID3 [Qui86], AQ11 [MC80], AQ15 [MMJL86], and MetaDendral [BF78].

ID3 has been the most widely studied and applied program for learning to perform classification from examples. Given a set of classified examples,
ID3 constructs a decision tree based on the features which describe them. The tree is constructed by recursively partitioning the training set until homogeneous subsets are obtained. The features which effect the partition can be chosen heuristically or by information-theoretic criteria. ID3's basic method has been enhanced to construct probabilistic decision trees in which the cases corresponding to a leaf have only a probability of belonging to the associated category [Qui86].

**Knowledge Compilation:** A different machine learning approach, explanation based learning [MKK86,GR86], has also been applied to knowledge acquisition. Explanation-based learning relies on a pre-existing theory of the problem domain during the process of concept acquisition. LEAP [MMS85] is an explanation-based learning apprentice for acquiring design rules for VLSI circuits.

LEAP combines learning with providing design assistance to its user. Given a design problem (of the form "design a circuit which when given input \( X \), produces output \( Y \)"), LEAP recommends a solution from its library of design rules. The user either accepts the solution or provides one of his own. In the latter case, LEAP can learn. It uses its domain theory to prove that the user's design meets the circuit's specification and to determine important constraints on the design, which guide the system in generalizing the design. The result is a design rule whose left-hand side is a generalized input-output specification and whose right-hand side is a description of a class of circuits which satisfies the specification. Strategic knowledge about the likely applicability of rules is not acquired. The roles of the user in the learning process are to provide a specification, to reject LEAP's proposed design, and to present an alternative; the user does not determine the form of the rule that the system acquires.
Protos: Like most other concept acquisition systems, Protos acquires a knowledge base from examples. Protos employs a combination of autonomous learning and focused interaction with its teacher. The teacher provides Protos with a sequence of cases to be classified. Protos decides which cases to retain and learns an indexing structure for accessing them. However, Protos' classification mechanism requires other types of information which are not always feasible to learn autonomously. Protos questions the teacher in the context of specific problem solving failures to acquire domain knowledge, some types of indexing information, and reassessments of the knowledge it has previously acquired.

Protos strikes a balance between autonomous learning and learning by asking questions. In contrast to traditional automated knowledge acquisition systems, Protos is more autonomous; in contrast with machine learning systems, Protos has a much wider bandwidth of communication with its teacher.

5.2.3 Input Provided to the System

In general terms, the role of an automated knowledge acquisition system is to capture the expertise of the user. For this to happen, the user must be able to communicate his expertise to the system in a form which is compatible with the system's requirements. Many of the systems being discussed require substantial pre-existing knowledge, which provides the system with leverage in performing its designated task. The assumption is always made that the initial body of knowledge is relatively easy to provide and that the system provides assistance with the truly difficult part of knowledge acquisition. Consequently, no assistance is provided for collecting, organizing, and encoding this initial input. These tasks are left to the expert or (more likely) to a human knowledge engineer who is assisting him.
When using an automated knowledge acquisition system, the user must also be able to provide appropriate input when queried. The role of most systems is to provide a context which facilitates the user's recall of bits of expertise. The burden of identifying and expressing the appropriate knowledge is the user's.

ROGET provides assistance with the initial phase of knowledge acquisition. It does not require a large body of pre-existing, specific domain knowledge to perform its function. To begin, the user must provide ROGET with a characterization of the nature of the problem to be solved, which allows it to select an appropriate general framework (i.e., model) for vocabulary acquisition. The user must be able to describe the nature of his task in terms known to the system such as "determine causes" or "recommend actions." Providing this input, which is essentially a classification, can force the user to reconceptualize his task, switching from his existing view to the heuristic classification view required by ROGET.

The bulk of the user's input to ROGET is information that instantiates the general model. This model identifies the kinds of domain terms which are expected to occur in the problem solving process. The user is asked to provide specific classifications along with features of the expected types which allow the classifications to be determined. This information is acquired outside of the context of actual problem solving; the user is asked to enumerate the required information in response to general queries (e.g., "What kinds of evidence would be used to determine X?").

ETS requires the user to provide three kinds of input. First, the user

---

9 Most other research in automated knowledge acquisition is predicated on the belief that such information is difficult to elicit via general queries.
must provide an exhaustive list of possible classifications. Second, the system presents the classifications in groups of three and asks the user to identify a feature which discriminates each member of the group from the other two. Third, the categories and features are combined in a rating grid, and the user is asked to rate every feature relative to each classification. The task of identifying discriminating features provides a natural problem solving context to facilitate appropriate input. The task of providing ratings, however, may have no natural analogue in the user's problem solving, and hence such input may be difficult to provide.

The most critical input to a knowledge base refinement system, such as TEIRESIAS, is an initial knowledge base on which to operate. If this knowledge base is not relatively complete and largely error free, the system's heuristic methods do not have the information they need to provide strong guidance in refining the knowledge base. The focused interaction of TEIRESIAS is dependent on the system's ability to identify existing rules which almost or erroneously apply to a particular problem. If the knowledge base is so sparse that it cannot identify such rules, there is no focus for its interaction with the user; he must answer general questions (such as "How do you conclude X?") which are believed to be much more difficult.

Even when TEIRESIAS has an adequate existing knowledge base, it is the user's responsibility to input a rule which repairs an identified problem solving failure. TEIRESIAS assists the user by checking his input against appropriate rule models, which are derived from regularities in the existing rule base. Because accuracy in empirical learning is dependent upon a large

---

10 TEIRESIAS has a limited ability to formulate rules itself in restricted contexts. However, the reliability of this feature has not been adequately demonstrated.
and representative set of training instances the rule models cannot be learned if the initial input to TEIRESIAS is sparse and inconsistent.

Another important input to TEIRESIAS is the set of test cases that exercises a rule base to expose its shortcomings. Experience in software engineering suggests that the selection of such a test set is a difficult task, which requires detailed knowledge of the system being tested. No automated assistance is provided for this problem.\footnote{Some knowledge refinement systems, such as MORE, rely on a static technique of structural analysis rather than on dynamic testing. In principle, this obviates the need for test cases. However, Eshelman's work on MOLE, the successor to MORE, led him to conclude that static analysis is of limited utility [EEMT87].}

The use of ASK is dependent upon the pre-existence of a knowledge base which completely captures the factual knowledge of a domain. This knowledge base allows the system to identify possible alternatives to the user's problem solving actions for discussion with the user to refine his justifications.

The user's interactive input to ASK is a set of problem solving histories represented as sequences of justified actions. Justifications are selected from a pre-enumerated (but user-extensible) set of alternatives. When ASK determines that an action is not a clear choice over known alternatives, the user must provide additional justifications until the system is satisfied.

The user must also provide ASK with criticism during the learning process. A justified action is generalized to formulate strategic rules by producing all of the syntactic generalizations that ASK's heuristic rules enable it to generate. These are presented to the user who must identify the ones which constitute generally valid problem solving heuristics (such as "prefer the least costly alternative").
Systems that learn autonomously by the comparison of examples, such as ID3, must be provided with appropriate training sets of examples. The cases in a training set must be chosen to be representative of the cases the resulting performance system is intended to classify. To be sure of accurate learning, large numbers of examples are necessary. It is the responsibility of the user to acquire such a training set and to represent its cases in appropriate featural terms for processing.

The determination of an appropriate language for featural description of cases may not be a simple task. Quinlan reported that the selection of appropriate features for describing chess positions took almost two man-months [Qui83]. Furthermore, the language that is natural for describing observed properties of cases may not be an appropriate language for representing generalizations [FD86]. Concept learning systems provide no assistance in choosing a representation. The user must be sophisticated enough to choose an appropriate one and to encode the training set accordingly.

Explanation-based learning systems, such as LEAP, place a still greater burden on the user. Before such a system can provide automated assistance, it must be pre-programmed with an appropriate theory of the domain in which a knowledge base is to be acquired. This domain theory is required to be complete and consistent with respect to the problems being solved; these properties enable the generation of formal proofs to guide the system's learning. The domain theory implicitly contains all of the knowledge which the system is expected to acquire. The knowledge acquisition process consists of compiling this knowledge into an efficient form for problem solving.

The required domain theory may be difficult to provide. It is perhaps possible for a domain such as the VLSI design where the theory simply con-
sists of a set of logical transformations (e.g., DeMorgan's Laws) and a library of primitive building blocks for logic circuits. Most expert domains, unfortunately, are less amenable to formalization and hence are much more difficult to characterize in complete and consistent terms. Explanation-based learning systems provide no assistance in the task of acquiring a domain theory.

Protos follows the machine learning tradition of acquiring a knowledge base from examples. The user's primary input to Protos is a sequence of cases for the system to classify. Although experimentation has demonstrated that a satisfactory knowledge base can be acquired from a random set of cases, it is believed that careful case selection will result in a better knowledge base. Cases presented to Protos are described by sets of features. As in other systems, these features must be selected by the user. However, since Protos does not attempt to induce general definitions in terms of these features, uniform description of cases is less important; because Protos learns from explanations of featural equivalence, heterogeneous descriptions can be related as necessary.

A crucial input to Protos is explanations. Protos requires the user to supply two types of explanations. The first type is explanations of featural equivalence. During problem solving, these explanations allow transformational matching of cases which are described differently. The second type is explanations of the relevance of case features to a classification. These explanations are used to derive reminders and to estimate the importance of features. Successful use of Protos is dependent upon the user's ability to formulate such explanations and to present them to the system in its predefined explanation language. Experience indicates that providing explanations can be a source of difficulty for some users and may limit the domains to which Protos can be applied.
5.2.4 The Results of Using the System

The knowledge acquisition systems discussed herein were designed to assist their users in a variety of ways. Consequently, the results produced by these systems (their "output") varies greatly.

ROGET instantiates a general model, capturing the vocabulary of the domain and general relationships between evidence and conclusions. The instantiated model is transformed into EMYCIN format. Its vocabulary translates into the goals and parameters integral to the EMYCIN reasoning process. Because a general question and answer process is used in knowledge acquisition, the user cannot be sure that all important concepts in the domain have been captured; he can only be sure that he has been asked about all types of concepts that ROGET expects to be present. ROGET provides only limited assistance in the task of formulating inference rules. The general relationships captured in the model are translated into rule schemata which are available to the user for copying and editing within EMYCIN, thereby easing the task of entering specific inference rules. There is no guarantee that these rule schemata are correct or that they will provide any leverage in the acquisition of specific inference rules.

ETS also systematically attempts to capture the vocabulary of a domain. However, unlike ROGET, ETS produces an executable rule base as its output. The rule base is created by applying a factor analysis technique to the rating grid to produce a graph of logical entailments, which is then translated into inference rules. The rule base produced by ETS tends to be shallow. The features most predictive of particular classifications are those acquired in the category discrimination task; most of the rules that ETS generates link these features directly with classifications. Intermediate inferences in the rule base
relate features to one another. These rules are not guaranteed to be useful in
the reasoning process. Furthermore, the resulting system is not guaranteed to
adequately cover the domain. Possible classifications may be omitted from the
initial enumeration, and the technique for eliciting features is not guaranteed
to identify all features necessary for unambiguous classification. Most impor-
tantly, Boothe and Bradshaw have noted that the ETS techniques do not scale
up to the construction of large systems [BB87]. The techniques for eliciting dis-
criminating features and constructing a rating grid quickly become unwieldy
as the complexity of the domain increases.\footnote{A successor to ETS, AQUINAS, has been designed to address many of these shortcomings [BB87].}

The use of a knowledge base refinement system such as TEIRESIAS,
produces a knowledge base that is more extensive and correct than the one to
which it was applied. New rules have been added, and erroneous ones have
been repaired. The degree of improvement, however, is largely dependent on
the quality of the user's set of test cases and his ability to follow the performance
system's inference process, formulating appropriate repairs when asked to do
so.

ASK does not extend a performance system's factual domain knowl-
dge. The system's pre-existing knowledge base is augmented with a collection
of strategic rules which guide the system in applying its knowledge during prob-
lem solving. The quality of these strategic rules is dependent upon four factors.
First, the user must select an appropriate set of training cases to exemplify pos-
sible problem solving strategies. Second, he must be able to adequately justify
his problem solving actions in the system's vocabulary. Third the system must
be able to produce useful generalizations by syntactic modification of justifi-
cations. Fourth, the user must be able to act as a critic to discriminate valid strategic rules from invalid ones.

The result of using a concept acquisition program, such as ID3, is a rule base for classifying cases described in the same featural terms as the system's training instances. The decision tree generated by ID3 corresponds to a set of classification rules. This rule base is shallow: Conjunctions of features are linked directly to classifications, and no intermediate concepts are induced. Also, the nature of the rules tends to be foreign to human experts [MC80,Qui83], so it is sometimes difficult for a user to accept a rule firing as justification for the conclusion reached by the system.

The result of using an explanation-based learning system, such as LEAP, is a compilation of the knowledge present in its domain theory. The system does not actually acquire any new knowledge during training. It merely transforms the knowledge present in its domain theory into a form more suited to efficient problem solving. In doing so, it produces the maximal generalizations of its training instances which are guaranteed to be sound. Its success depends on the quality of its domain theory, the amenability of the theory to formal proof, and the tractability of the proof process.

The results of using Protos are a set of exemplars which cover the categories of the domain, indexing information which allows efficient exemplar retrieval during problem solving, and an accompanying body of domain knowledge which allows inferential matching of the exemplars to new cases. As in other systems which learn from examples, the quality of the knowledge base is dependent upon the user's ability to assemble an appropriate training set. However, Protos' abilities to acquire and use domain knowledge for inferential matching make it less susceptible to the effects of training set choice than sys-
tems which attempt to abstract the training set into general rules. However, Protos' success is dependent upon the user's ability to provide adequate explanations during training. Experimentation has shown that the Protos approach is robust; it has been demonstrated to attain expert performance from a reasonably sized set of randomly chosen cases and a body of simple explanations.

5.2.5 The Systems' Sources of Power

To understand automated knowledge acquisition systems, it is important to recognize their sources of power — how they are able to provide the assistance that they do. Each system possesses heuristic knowledge to facilitate the transfer of the type of domain knowledge it is intended to acquire.

The primary source of power of ROGET (and other model-based systems, such as KNACK [KGM87]) is a general model which identifies important characteristics of the problem-solving domain. ROGET has a set of such models which correspond to variants of the task of heuristic classification. Its heuristic knowledge of the applicability of these models allows it to interact with the user to select an appropriate one or (presumably) to recognize when no model adequately applies. The selected model provides strong expectations about the structure of the domain which allow ROGET to question the user systematically to elicit important conceptual elements of the domain and the general relationships between these elements. The model also provides term-specific expectations about descriptive information (such as the reliability of a test result) which is likely to be important in the problem-solving process. However, ROGET interviews the user in a context much different than the one in which the user solves problems. He is asked to reflect upon his knowledge and to characterize it given guidance only in abstract terms. The lack of an
actual problem solving context may impair the user's ability to recall important concepts and relationships.

Another source of power in ROGET is analogy to other expert systems. ROGET has knowledge of the expert systems from which its models were abstracted. Whenever a question is posed to the user, it is able to provide specific examples to illustrate appropriate responses. It should be noted, however, that the process of analogical reasoning takes place in the mind of the user, who must map the example from its original domain to an appropriate analogue in his own domain.

A final source of power possessed by ROGET is knowledge of how the information it captures is represented in EMYCIN. This enables ROGET to provide automatic translation of an instantiated model into a set of EMYCIN goals and parameters.

ETS derives power from knowledge of the problem solving method employed by MYCIN-like, rule-based systems. It "knows" that a domain must be characterized in terms of classifications and the features which allow an expert system to discriminate among them. To assist the user in providing this information, ETS makes use of the two artificial tasks discussed earlier: discriminating among subsets of classifications and constructing a rating grid. These tasks provide a simplified problem solving context for the elicitation of features and their relationships to classifications. The value of these tasks is heuristic; ETS is not guaranteed to identify a sufficient set of goals and features to adequately capture the information needed for problem solving.

ETS also draws power from the factor analysis algorithm which generates a graph of logical entailments for the rating grid filled in by the user. The primary function of this algorithm is to identify dependency relationships
that allow some features of a case to be derived from other features during problem solving. The quality of the results are dependent upon the quality of the ratings input by the user.

Finally, ETS has knowledge about the form of rules comprising a MYCIN-like expert system and can generate them automatically from the entailment graph. It also has heuristics which allow it to infer certainty factors for the rules based on ratings in the rating grid.

Rule refinement systems, such as TEIRESIAS, draw power from knowledge of the problem solving method used in the expert system under construction. TEIRESIAS has considerable heuristic knowledge of the inference process of MYCIN-like systems. It is an implementation of heuristic methods of debugging such systems gathered from experience in manual debugging.

TEIRESIAS also draws power from the rule models which it abstracts from the initial rule base. These models allow the system to check rules as they are input by the user; they provide probabilistic guidance in the identification of missing premises and conclusions. However, the rule models are reliable only in situations where the rules present in the initial rule base are representative of those to be acquired.

ASK possesses general heuristic knowledge of problem solving strategy and heuristics for analyzing the body of factual domain knowledge with which it is provided. When the user provides ASK with a justification for a problem solving action, heuristics allow the system to determine if the user's preference is adequately differentiated from other possibilities and to interact with the user to capture an adequate justification.

ASK also uses heuristics to generalize justifications based on their syntactic form (see [Mic83] for an enumeration of such heuristics). The suc-
cess of the generalization process is dependent upon the justification language, which ensures that justifications will be simple and will be represented in terms amenable to syntactic generalization. The most important source of power in this process, however, is the user, who serves as a critic, identifying valid generalizations. Without such powerful guidance, ASK has no means of identifying which generalizations have validity in a larger problem solving context.

Much of the research interest in machine learning systems such as ID3 comes from a desire to identify their sources of power. Inductive generalization is an unsound process, yet these systems have been demonstrated to produce useful results. ID3 has especially interested researchers because of the simplicity of its algorithm.

Much interest has focused on the information theoretic methods used to select features during learning, based on the belief that these formally characterizable techniques provide a general alternative to unsound, heuristic ones. It is the belief of this author, however, that ID3's main source of power is the featural language defined by the user to characterize the training set. Appropriate choice of features enables the system to use its simple algorithms to detect important commonalities and differences among its training instances.

Explanation-based learning systems, such as LEAP, draw power from their pre-existing domain theories. LEAP's domain theory implicitly contains all of the knowledge that can be learned by the system. The completeness of the domain theory determines the relationships which can be identified and compiled by LEAP.

LEAP also draws power from characteristics of its domain of application: VLSI design. This domain has a foundation in formal logic. An adequate domain theory consists of a small number of rules expressing logical transforma-
tions and a correspondingly small library of primitive building-blocks for logic circuits. Hence the formal proof process that underlies learning is tractable.

Like many of the other systems discussed, Protos draws power from knowledge of its problem solving method. It "knows" that it must retain an appropriate set of exemplars for matching new cases, compile an indexing structure, and acquire domain knowledge to enable its explanations relating features to categories and equivalent features. Heuristic evaluation of explanations guides the retention of exemplars and acquisition of indices. Like TEIRESIAS, Protos relies on focused interaction with the user during problem solving to acquire missing domain knowledge and to repair erroneous relations in its knowledge base. Heuristic knowledge of problem solving focuses discussion on specific reasons for failures to classify a case or to explain a classification adequately.

In this section, four characteristics were identified to compare automated knowledge acquisition systems and a representative subset of existing systems were compared accordingly. Automated knowledge acquisition systems can be grouped according to the types of assistance that they provide to the user, the information that the user must provide as input to the process, the results of using the system, and the sources of power that enable the system to function. Because the systems discussed in this section were motivated by different research goals and provide assistance with different phases of the knowledge acquisition process, it is difficult to say conclusively that one approach is better than another. However, Protos is unique in that it combines autonomous learning with focused interaction to construct effectively a knowledge-based system sufficiently powerful to solve real-world classification problems.
Chapter 6

Contributions and Future Work

We all have occasional failures. Fortunately, Dr. Watson never writes about mine.

— attributed to Sherlock Holmes

The result of this research is Protos, a system that learns to perform classification in the course of classifying instances of natural and goal-directed concepts (i.e., equivalence classes with respect to human problem solving goals). A human teacher presents Protos with cases to be classified and provides explanations of cases that the system fails to classify correctly. Through this training, the system acquires concepts, builds an indexing structure, and acquires a general body of domain knowledge which enables the system to explain its classifications. Protos was demonstrated to learn to classify accurately and efficiently.

This chapter summarizes Protos' contributions in the areas of concept representation, classification, and learning and then provides suggestions regarding further experimentation with the current implementation of Protos and new research to refine the Protos approach and to extend its applicability.
6.1 Contributions

Natural and goal-directed concepts, which are ubiquitous in real-world problem solving, impose requirements on concept representation, classification, and learning which must be satisfied by a real-world classifier and learner. Concept representation must capture the polymorphism of these concepts. Classification must involve reasoning under uncertainty because of variations in case descriptions, noise, and missing information. Learning must acquire a representation that enables accurate and efficient classification.

The overarching contribution of the Protos project is a unified approach to concept representation, classification, and learning that satisfies these requirements. This approach and its implementation in Protos are summarized in the next three subsections. The fourth subsection summarizes experimental results that support the validity of the approach.

6.1.1 Representation

Protos' knowledge representation is a category structure which captures the polymorphism of natural and goal-directed concepts while representing the essential similarity of their instances that makes concepts cohesive. The category structure captures the polymorphism of concepts by representing them extensionally. A category (the extension of a concept) is represented by a set of retained cases. These exemplars have been chosen by the system to represent the variability of significant features found in the entire range of category members.

Category cohesiveness is provided by domain knowledge that which makes the underlying commonalities of category members explicit in the representation. This knowledge captures the relationships between features and
classifications in the form of explanations of the relevance of features to category membership. It also captures the similarity of case descriptions as explanations of featural equivalence.

6.1.2 Classification

In Protos, classification is a two-step process. The first step is to hypothesize a classification based on the features of a new case. The second step is to confirm the classification by retrieving an appropriate exemplar of the hypothesized category and constructing an explanation of its similarity to the case.

Two indexing mechanisms contribute to the process of hypothesis formation. Remindings associate features with categories (or particular exemplars). A reminding suggests that a new case is an exemplar of a category because it possesses features associated with known category members. Censors are negative associations between features and categories. A censor suggests that the presence of a feature in a new case disconfirms its membership in a category.

The remindings and censors associated with the features of a new case are combined heuristically to yield an ordered list of possible classifications. Multiple remindings to the same category increase Protos' belief in its relevance. Also, remindings are inherited taxonomically so that more specific categories are tried first during the hypothesis confirmation process. If an absolute censor is associated with a case feature, however, the hypothesized classification is excluded from further consideration. A weaker censor serves only to weaken a hypothesis; such censors affect the order in which hypotheses are tried but do not preclude their consideration.
Protos attempts to confirm its hypotheses in order of preference. Hypothesis confirmation consists of locating an exemplar of the hypothesized category that is likely to be similar to the new case, explaining the equivalence of their features, and checking to ensure that the match between them is not the reoccurrence of an incorrect match. When an adequately similar exemplar is found, it is reported to the teacher. Additional hypotheses are considered only if the teacher requests further search.

When Protos considers a hypothesized category, exemplars are selected for comparison to the new case on the basis of their prototypicality ratings. An exemplar's prototypicality rating is a heuristic estimate of its family resemblance to other category members. Hence, if a new case is a member of the category, it is a priori more likely to be matched by an exemplar with a high prototypicality rating. However, if such an exemplar is not adequately similar to the new case, Protos is willing to consider other, less prototypical, exemplars.

An exemplar serves as a specific model for interpreting the new case. It provides expectations of features the new case should possess and suggests the importance of confirming their presence. The similarity of the new case to the exemplar is gauged by a process of knowledge-based pattern matching which uses domain knowledge within the category structure to explain the equivalence of case features to exemplar features based on direct (e.g., correlational) association or knowledge of underlying commonality (e.g., function). The quality of explanations of featural equivalence and the importance of unmatched exemplar features are considered when assessing similarity.

When knowledge-based pattern matching results in an imperfect match, difference links associated with the matched exemplar may suggest better
matches. These difference links point to exemplars which the current exemplar imperfectly or incorrectly matched during past classification attempts. By considering these exemplars as alternative matches for the new case, Protos is able to locate the closest match and to avoid the reoccurrence of past classification failures.

When an adequately explained match is found, it is presented to the teacher. In addition to reporting the classification, Protos explains in symbolic terms why it believes that the new case should be so classified. Discussion of the match provides opportunities for learning.

6.1.3 Learning

Learning is driven by the needs of classification. Protos' learning objectives are to acquire a representation that enables accurate and efficient classification and to acquire domain knowledge that allows it to explain its classifications. Failure to classify a case correctly or to explain a classification adequately identifies missing knowledge that must be acquired.

Protos learns a category by selectively retaining the cases necessary to represent its polymorphism. A new case is retained as an exemplar when Protos cannot classify it correctly or when its similarity to an existing exemplar cannot be adequately explained.

Protos learns from teacher-provided explanations in addition to cases. It acquires explanations in two situations. The first is when a new exemplar is added to the category structure; Protos acquires explanations of the relevance of its features to its classification. These explanations are analyzed to compile reminders and to estimate the importances of the case's features to its classification. The second situation in which Protos acquires explanations
is when a correct, but imperfect, match is discussed with the teacher; Protos learns explanations of featural equivalence. These explanations enable the knowledge-based pattern matching process by which new cases are classified.

In some learning situations, the teacher cannot convey the necessary information to Protos via concise explanations. Consequently, Protos combines empirical learning and question-asking with learning from explanations. The system learns prototypicality ratings empirically by crediting exemplars for participation in strong, correct matches. Other types of information, however, are inappropriate to be learned autonomously because of the small amount of training available or the complexity of the learning problem. Protos learns censors and difference links by asking questions in situations in which the teacher is likely to give correct answers.

6.1.4 Experimental Evaluation of Protos

Protos was evaluated experimentally by applying it to the problem of knowledge acquisition for expert, heuristic classification. An expert task provides objective standards for evaluating a system's performance. Learning to perform heuristic classification was an appropriate test of Protos because heuristic classification embodies the real-world issues of concept representation, classification, and learning that were fundamental to this research.

Protos was applied in the domain of clinical audiology. Its task, learning to classify hearing disorders, was representative of learning heuristic classification: Protos learned a set of diagnostic categories by processing a sequence of patient descriptions. The Department of Speech and Hearing at the University of Texas provided an interested expert to serve as Protos' teacher and a population of students and clinicians to serve as a standard for evaluating
Protos' performance.

Protos learned to classify hearing disorders expertly. Its average accuracy was 82% on 200 training cases (excluding the first instance of each diagnostic category) and 100% on 26 test cases which were presented after training. This is comparable to the performance of experienced human clinicians.

Protos' classification effort increased as the size of its category structure grew, but its performance did not degrade unacceptably. On average, it considered fewer than 18% of possible classifications for a new case (this percentage was fairly constant across the training and test cases). The majority of Protos' effort was independent of the teacher. It was able to reject most incorrect classifications independently and only presented an average of 1.5 matches per case to the teacher for discussion.

The teacher's training effort diminished as Protos acquired knowledge. The nature of his interaction with the system shifted from providing the explanations necessary to integrate new cases into the category structure to providing key bits of knowledge that allowed Protos to recognize similar case descriptions.

The rate of growth of Protos' category structure slowed over time, suggesting that the amount of knowledge required by Protos should not be of concern in domains with the same general characteristics as audiology (as discussed in Section 4.4). The relative rates of exemplar retention for various diagnostic categories mirrored their relative polymorphism. Diagnoses with simple underlying causes (e.g., biomechanical problems) required few exemplars while less cohesive, "catchall" categories required many.

Significantly, Protos proved to be a practical tool. It learned to perform an expert classification task successfully from a reasonable amount of
training. By doing so, Protos demonstrated the effectiveness of its approach to concept representation, classification, and learning.

6.2 Future Work

The Protos paradigm is rich with opportunities for additional research. These opportunities fall into two categories: First, further experimentation is desirable to determine when Protos can best be applied and the system's sources of power. Second, new research can improve the mechanisms comprising Protos and broaden the applicability of the Protos approach.

6.2.1 Further Experimentation

The experimental application of Protos in the domain of clinical audiology does not constitute a total validation of the Protos approach. The results are demonstrably relevant to learning other classification tasks only in so far as audiology is representative of the domains of interest. It is possible that properties of this domain such as its reasonably small number of categories and concise diagnostic explanations make it unusually suitable for the application of Protos. Protos should be applied to a broad range of expert and common sense classification tasks to understand the bounds of its applicability. Dan Dvorak has reimplemented Protos and is applying it to learning to select qualitative models of cardiac function.¹

In addition to studying Protos in a variety of domains, it should be compared to other approaches to learning classification. Machine learning re-

¹All of the specific research projects mentioned in this section are being pursued by students in the Department of Computer Sciences at the University of Texas.
search has been largely concerned with classification. A variety of empirical, similarity-based, and explanation-based techniques have been demonstrated to have utility. A comparative study should be undertaken, involving several task domains, to determine the situations in which Protos' unique capabilities offer advantages over alternative techniques. Rita Duran and Rich Mallory are currently performing studies which compare the performance of some other learning algorithms to Protos' performance at learning to classify hearing disorders.

Finally, further experimentation is necessary to determine Protos' sources of power. The experiment of learning audiology evaluated the performance of Protos as a whole. However, Protos is a collection of heuristic mechanisms which are based on a large number of underlying assumptions. Systematic experimentation should be performed to assess the contributions of various mechanisms and the system's sensitivity to variations in its many internal parameters. Such experimentation is desirable to determine which underlying assumptions were justified and hence which aspects of Protos can be incorporated most profitably into the work of other researchers.

6.2.2 New Research

Additional study of the heuristic processing of explanations is necessary. Protos relies on explanations for acquiring knowledge from its teacher, reasoning about classifications, and justifying its conclusions. Its simple mechanisms are only a first step towards solving the problems involved. Additional research is needed on the representation of explanations, mechanisms for assimilating new explanations into existing knowledge, and mechanisms for the generation of explanations. James Lester, Ken Murray, Karen Pittman, Bruce
Porter, and Art Souther are currently studying these issues in the context of a large-scale knowledge base.

An integrated approach that combines learning from experience with learning from explanations needs to be developed. Currently Protos' knowledge tends to increase monotonically, and the organization of that knowledge reflects the contents of teacher-provided explanations. Experience in using the acquired knowledge may suggest useful reorganizations. Conversely, revisions of the system's knowledge resulting from problem solving experience may suggest corresponding revisions of explanations. The roles of experience and explanation in learning need to be better understood, and an integrated learning mechanism should be devised which embodies this understanding.

The Protos paradigm of learning and reasoning from exemplars and explanations is not fundamentally limited to classification. It could be extended to apply to learning other types of problem solving such as planning. Such an extension raises a number of research issues including the representation of exemplars, the nature and roles of explanation, and the modification of solutions to known problems to apply to similar new problems. Karl Branting is currently extending the Protos paradigm to produce a case-based system for determining and explaining legal relationships. In particular, he is addressing the issue of transforming the explanation of a known case to apply to a new case.

6.3 A Final Word

The discovery of unifying principles is a fundamental goal of research in artificial intelligence. When feasible, the joint study of related issues of cognition is a means of pursuing this goal. Such a method was employed in
the Protos project. Protos began as machine learning research on real-world concept learning. However, it quickly became apparent that concept learning could not be studied in isolation. Consideration of learning issues raised issues of concept representation and classification, which took on equal importance. The success of Protos in both learning and classification resulted from its unique category structure composed of exemplars augmented by explanations. Protos’ unified approach to representation, classification, and learning, built on this foundation, produced a system that has been demonstrated to perform effectively.
Appendix A

Psychological Research on Concept Representation

The psychological literature contains a large body of research on human concept representation. In general terms, researchers have embraced three competing views [SM81]:

1. the Classical View
2. the Probabilistic View
3. the Exemplar View.

This appendix defines each of these views in turn, discusses its strengths and weaknesses, and explores variations which are aimed at rectifying its weaknesses.

A.1 The Classical View

The Classical View [SM81] holds that a concept is defined by a set of features which are singly necessary and jointly sufficient for classifying concept instances. Although a concept instance must have all of the defining features, it is not precluded from having additional features. However, the defining features are the only ones relevant to the concept.

Classical concept definitions offer three theoretical advantages. First, they are easy to learn. A concept is induced from training instances simply by
collecting the features that all instances have in common. Second, classifying a concept instance is correspondingly easy. If all of the defining features are present, classification succeeds; otherwise it fails. Third, the Classical View reflects a common sense meta-theory of concept representation [MS84]. Even in the face of contradictory evidence, people believe that classically defined concepts are the norm.

Unfortunately, problems with the Classical View far outweigh its advantages. Six fundamental problems have been identified [CSdFM80, SM81]:

1. People are unable to specify defining features for most concepts.
2. Unclear cases frequently arise during classification.
3. Featural descriptions of instances of a single concept can vary greatly.
4. Concept instances vary in typicality.
5. The nesting of the features of a superordinate concept within those of its subordinates is often imperfect.
6. People frequently use nonnecessary features during classification.

Some theorists have attempted to explain these problems by differentiating the definition of a concept from its classification procedure. They have posited that concept instances are classified by heuristic procedures, and a concept’s (classical) definition merely serves as a validity check on the classification of instances. This theory, however, suggests that classical definitions alone are insufficient as a concept representation.

A more radical attempt to salvage the Classical View has led to the Transformational Model [SM81]. Concepts are still required to have classical definitions, but the notion of matching features has changed. Features of an
object are not required to match a concept's definition directly. Rather, they are required to be transformable, in some sense, into the features comprising the definition. Both learning and classification are highly dependent upon the total state of the entity's knowledge. Nelson's theory of functional and relational concept definitions [Nel74,Nel77] is the most developed example of the Transformational Model.

A.2 The Probabilistic View

The Probabilistic View [SM81] overcomes the problems of the Classical View by weakening the notion of a concept definition. A probabilistic concept definition consists of a set of features which have appeared in previously seen instances of the concept. The definition is a list of features which might appear in an instance of the concept; it is not directly realizable as an exemplar.

Concepts are learned by adding features to an evolving concept definition as they are encountered in new instances. Weights may be learned reflecting the relative importance of various features. The weight assigned to a feature may be a function of its frequency of occurrence or predictiveness.

Classification is more complex than in the Classical View. It is performed by applying a similarity function which evaluates featural matches between a concept definition and an object to be classified. The most commonly discussed function is a weighted sum of features. If the function's value exceeds a threshold, the classification is made.

A serious problem results from the relaxed concept definition. The classifier lacks information about which combinations of features comprise realizable instances of a concept. Neither knowledge of correlated groups of features
nor knowledge about the acceptable ranges of values for individual features is represented. Several studies (e.g., [HH77, DS78, MAF82, FH84]) have demonstrated the importance of such information in the classification process.

A.3 The Exemplar View

The Exemplar View [SM81, Bro78] also overcomes the problems of the Classical View. In the Exemplar View, a concept is represented extensionally; it is synonymous with its corresponding category. The definition of the concept is implicit in its instances; no explicit definition is abstracted. Consequently, information about featural correlations, acceptable feature values, and realizable concept instances is preserved in the representation.

A concept is learned simply by storing examples of its category members. Two possible strategies have been suggested for determining which exemplars to store. One strategy is simply to store every unique instance of a concept which is encountered. Another, attributed to Rosch [SM81], is to store the $N$ best exemplars of the concept, using the criterion of family resemblance.

Objects are classified by matching their features to the features of stored exemplars. There are two basic classification strategies. One strategy is to assign the classification when an object is very similar to any exemplar of a concept. Another is to look for a high degree of match between the object and several exemplars of the concept. The simplest way to evaluate the match is to compute the percentage of matching features. Simple exemplar models (e.g., [Bro78]) assume that the features of an exemplar are uniformly weighted in the matching process. An alternative is to assign different weights to different featural matches, depending on their relative importances. However, the Exemplar View does not address the acquisition of such information.
The significant shortcoming of the Exemplar View is that it does not address the issue of category cohesiveness. Not every collection of exemplars constitutes a reasonable concept. However, the underlying commonality of category members is not explicit in an exemplar representation. Additional knowledge is required to explain cohesiveness [MM85].

An exemplar-based concept representation comes closest to sufficiency for classification and many related tasks which are of interest in AI research. Because exemplar representations have not been widely applied in AI, the design and construction of such systems should be fruitful research. However, because the emphasis of psychology is on explanatory modeling rather than computation, exemplar models from the psychological literature are inadequately specified for implementation in AI systems. In particular, the problems of category cohesiveness, exemplar matching, restricting search, and exemplar retention must be addressed. Our solutions to these problems are embodied in Protos' concept representation and algorithms.
Appendix B

Internal Representation of the Category Structure

This appendix contains the Prolog representation of the sample category structure illustrated in Figures 2.1 and 2.2.

/* exemplar chair1 of the category chairs: */
has_exemplar(chairs, chair1).
prototypicality(chair1, 2.25).

has_feature(chair1, backrest).
has_feature(chair1, seat).
has_feature(chair1, pedestal).
has_feature(chair1, metal).
has_feature(chair1, armrests).
has_feature(chair1, wheels).

/* exemplar chair2: */
has_exemplar(chairs, chair2).
prototypicality(chair2, 1.0).

has_feature(chair2, backrest).
has_feature(chair2, seat).
has_feature(chair2,legs(four)).
has_feature(chair2,wood)

/* featural importances -
   importance(Feature,Category,Importance). */
importance(backrest,chairs,0.5).
importance(seat,chairs,0.75).
importance(metal,chairs,0).
importance(wheels,chairs,0).
importance(wood,chairs,0).
importance(pedestal,chairs,0.25).
importance(armrests,chairs,0.25).
importance(legs(four),chairs,0.25).

/* explanation links -
   link(From,Qualifiers,Relation,To). */
link(seatSupport,[],hasTypicalSpec,legs(four)).
link(legs(four),[],hasTypicalGen,seatSupport).
link(holds(person),[sometimes],isEnabledBy,lateralSupport).
link(lateralSupport,[],enables,holds(person)).
link(lateralSupport,[sometimes],isFunctionOf,armrests).
link(armrests,[],hasFunction,lateralSupport).
link(holds(person),[sometimes],isEnabledBy,seatSupport).
link(seatSupport,[],enables,holds(person)).
link(seatSupport,[],hasTypicalSpec,pedestal).
link(pedestal,[],hasTypicalGen,seatSupport).
link(holds(person),[sometimes],isEnabledBy,seat).
link(seat, [], enables, holds(person)).
link(chairs, [sometimes], hasFunction, holds(person)).
link(holds(person), [], isEnabledOf, chairs).
link(holds(person), [sometimes], isEnabledBy, backrest).
link(backrest, [], enables, holds(person)).

/* reminders -
   reminding(Feature, Category, Strength). */
reminding(backrest, chairs, 0.9).
reminding(seat, chairs, 0.9).
reminding(pedestal, chairs, 0.6).
reminding(armrests, chairs, 0.6).
reminding(legs(four), chairs, 0.6).

/* difference links -
   difference(Exemplar, Category, Neighbor,
              Neighbor's Category, Neighbor's unmatched features). */
difference(chair2, chairs, chair1, chairs, [pedestal, armrests]).
difference(chair1, chairs, chair2, chairs, [legs(four)]).
Appendix C

Protos’ Explanation Language

This appendix lists the predefined relational links and qualifiers of Protos’ explanation language along with the a priori, default strength of each. When a relational link is used in an explanation, its actual strength is heuristically determined as a function of its default strength, any associated qualifiers, and the context in which it is used (as discussed in Chapter 3).

C.1 Relational Links

Unidirectional relational links are presented in the format:

\[ \text{link (strength) / inverse link (strength)} \]

Bidirectional links are presented in the format:

\[ \text{link (strength)} \]

- Correlational:
  - co-occurs (0.8)
  - requires (0.9) / is required by (0.9)
  - is consistent with (0.8) / exhibits (0.8)
  - implies (0.8) / is implied by (0.8)
  - spurious (0.0)

- Circumstantial-to-Inferred:
- if and only if (0.9)
- suggests (0.8) / is inferred from (0.8)

- **Definitional:**
  - is equivalent to (1.0)
  - definition implies (0.9) / *no inverse*

- **Generalization / Specialization:**
  - has typical generalization (0.9) / has typical specialization (1/bf)\(^1\)

- **Part-to-Whole:**
  - has part (0.7) / is part of (1/bf)

- **Causal:**
  - causes (1.0) / is caused by (1/bf)

- **Functional:**
  - has function (1.0) / is function of (1/bf)
  - enables (0.9) / is enabled by (1/bf)

- **Predicate-to-Argument:**
  - acts on (1.0) / is acted on by (1/bf)
  - affects (1.0) / is affected by (1/bf)

---
\(^1\)This link has variable strength. When it occurs in an explanation, its strength is 1.0 divided by the branching factor of links of this type connected to the current node in the search graph.
- Conjunctive:
  - is conjunct (1.0) / has conjunct (1.0)\(^2\)

- Mutual Exclusion:
  - is mutually exclusive with (1.0)

C.2 Qualifiers for Explanation Links:

When a qualifier is associated with a relational link in an explanation, the link’s qualified strength is determined by multiplying its default strength by the strength of the qualifier. The qualifiers “always” and “certainly” are exceptions; if one of these is used in an explanation, its strength (1.0) is substituted for the default strength of the relational link.

- Strength of Relationship:
  - strongly (1.0)
  - moderately (0.7)
  - weakly (0.4)

- Breadth of Context:
  - always (1.0)
  - usually (0.8)
  - sometimes (0.6)
  - occasionally (0.4)

\(^2\)These are Protos' internal links. The user specifies a conjunction by using the connective "and" in an explanation.
- Certainty of Belief:
  - certainly (1.0)
  - probably (0.7)
  - possibly (0.4)
Appendix D

Heuristics Used in Formulating Explanations

Protos' knowledge-based pattern matching algorithm performs uniform cost search to find sequences of relations (i.e., explanations) relating nodes in Protos' category structure. At every step in the search procedure, the algorithm must determine which relation is the best continuation of the current search path. In the absence of heuristic guidance, its only option would be to choose the relation which is, in isolation, most certain. In a broader context, however, a sequence of such steps often does not constitute a reasonable explanation. The heuristics listed in this appendix enable Protos to make better choices during search by considering more contextual information, such as the current classification hypothesis and the other relations involved in the emerging explanation.

The majority of the enumerated heuristics block poor choices instead of promoting good ones. The current scope of the Protos project does not include a general theory of explanation which is sufficiently powerful to provide only positive guidance in the formulation of explanations. The acquisition of heuristics was a pragmatic process during the initial use of Protos. Whenever Protos produced a bad explanation, an attempt was made to formulate a general rule to prevent its reoccurrence (and similar ones). Explanation-promoting heuristics were formulated whenever good explanations were evaluated to be weaker than Protos' user believed they should be.
D.1 Feature-To-Feature Heuristics

Protos uses the following heuristics when formulating explanations of the equivalence of features:

1. Two features are not equivalent by virtue of their common relationships with the category of the exemplar being matched. For example, "carnivorous is consistent with dog which usually exhibits barking" does not imply the equivalence of carnivorous and barking.

2. Two features are not equivalent by virtue of their common relationships with a sibling category of the category of the exemplar being matched. For example if the hypothesized category is dog, "carnivorous is consistent with coyote which usually exhibits barking" does not imply the equivalence of carnivorous and barking.

3. Two features (or categories) are not related just because they were involved in a previous conjunctive explanation. For example, "(legs(4) and big paws) is consistent with Great Dane" does not imply the equivalence of legs(4) and big paws.

4. A multi-step correlational explanation is weak, especially if it involves more than two inference steps. For example, "odor is sometimes consistent with dog sometimes co-occurs with boy usually co-occurs with toys" does not imply the equivalence of odor and toys.

---

An explanation of featural equivalence is an explanation that, because of underlying domain knowledge, two features provide a similar type of evidence for a classification. Such an explanation can be absolute, or its certainty can be qualified (as described in Section 3.1.2). Like a formally defined equivalence relationship, featural equivalence is reflexive. However, because the relationship can be qualified in the explanation, it is not guaranteed to be symmetric or transitive.
5. A path through a common generalization is acceptable but not continu-
able. For example, "chair legs have typical generalization seat support
has typical specialization pedestal usually has part wheels" does not
equate chair legs and wheels.

6. A path from a generalization to a specialization is not continuible to
another generalization. For example, "hunts in pack is consistent with
wolf has typical generalization canine has typical specialization dog has
typical generalization pet is sometimes consistent with lives indoors"
does not equate hunts in pack with lives indoors.

7. The function of an assembly cannot be ascribed to one of its parts. For
example, "volume control is part of tv set has function communication
..." does not relate volume control to terms in the continuation of this
explanation.

8. If a composite system is the cause of something, the causality cannot be
ascribed to one of its parts. For example, "water is part of acid rain
causes ecological damage ..." does not relate water to terms in the
continuation of this explanation.

9. When selecting an inference step, prefer one which leads to the goal
feature over one which is not known to lead to the goal. For example
if the goal feature is probocis, prefer "nose is sometimes equivalent to
probocis" to "nose is part of face ..."

10. Mixing causal and correlational links within a chain of inferences is sus-
picious but not prohibited (i.e., the certainty of a link that creates such
a situation is reduced).

11. Mixing functional and correlational links within a chain of inferences is
suspicious but not prohibited.
12. Two features are not equivalent by virtue of being parts of the same assembly. For example, "wheels are part of car which has part engine" does not imply that wheels are equivalent to engine.

13. Two features which share a common function are likely to be equivalent. For example, "wings enable lift which is enabled by helicopter rotor" is a strong explanation of the equivalence of wings to helicopter rotor.

14. Two categories which cause the same thing are likely to be somewhat equivalent. For example, "bacterial infection sometimes causes fever which is sometimes caused by viral infection" is a weak, but valid explanation of the equivalence of bacterial infection to viral infection.

15. The length of a sequence of causal relationships should not diminish belief in its quality (i.e., a sequence of causal inferences can be considered to be equivalent to a single inference step). For an example, see heuristic 15 of the next section.

16. The length of a sequence of functional relationships should not diminish belief in its quality. For an example, see heuristic 16 of the next section.

17. The length of a sequence of generalization relationships should not diminish belief in its quality. For an example, see heuristic 17 of the next section.

18. In the absence of more specific guidance to the contrary, the longer an explanation is, the weaker it is.

### D.2 Feature-To-Category Explanations

Protos uses the following heuristics in formulating explanations of the relevance of features to category membership:
1. An explanation cannot involve a sibling of the category to which the explanation is attempting to relate the feature. For example, given that coyote and dog have a common generalization, "carnivorous" is consistent with coyote which usually exhibits barking which is consistent with dog does not imply a relationship between carnivorous and dog.

2. Two features (or categories) are not related just because they were involved in a previous conjunctive explanation. For an example, see heuristic 3 of the previous section.

3. A path through a common generalization is acceptable but not continuable. For an example, see heuristic 5 of the previous section.

4. A path from a generalization to a specialization is not continuable to another generalization. For an example, see heuristic 6 of the previous section.

5. When selecting an inference step, one leading to the target category is preferred over one that is not known to do so. For an example, see heuristic 9 of the previous section.

6. A multistep correlational explanation is weak, especially if it involves more than two steps. For an example, see heuristic 4 of the previous section.

7. When relating a feature to a category, traversing a specialization link from a general category is a reasonable step in the explanation (i.e., a very common explanation form relates a feature to a generalization of the category of interest). For example, "body covering(hair)" is consistent with mammal has typical specialization dog is a reasonable explanation relating body covering(hair) to dog.
8. Mixing causal and correlational links within a chain of inferences is suspicious but not prohibited.

9. Mixing functional and correlational links within a chain of inferences is suspicious but not prohibited.

10. The function of an assembly cannot be ascribed to one of its parts. For an example, see heuristic 7 of the previous section.

11. If a composite system is the cause of something, the causality cannot be ascribed to one of its parts. For an example, see heuristic 8 of the previous section.

12. Two terms are not equivalent by virtue of being parts of the same assembly. For an example, see heuristic 12 of the previous section.

13. Two terms which share a common function are likely to be equivalent. For an example, see heuristic 13 of the previous section.

14. Two categories which cause the same thing are likely to be somewhat equivalent. For an example, see heuristic 14 of the previous section.

15. The length of a sequence of causal relationships should not diminish belief in its quality (i.e., a sequence of causal inferences can be considered to be equivalent to a single inference step). For example, "dizziness is sometimes caused by vertigo which is sometimes caused by vestibular problem which is sometimes caused by Menieres Disease" is equivalent to "dizziness is sometimes caused by Menieres Disease."

16. The length of a sequence of functional relationships should not diminish belief in its quality. For example, "wings enable lift enables flight" is equivalent to "wings enable flight."
17. The length of a sequence of generalization relationships should not diminish belief in its quality. For example, "body covering(hair) is consistent with mammal which has typical specialization canine which has typical specialization dog which has typical specialization collie" is equivalent to "body covering(hair) is consistent with mammal has typical specialization collie"

18. In the absence of more specific guidance to the contrary, the longer an explanation is, the weaker it is.
Appendix E

The Questionnaire Presented to the Human Subjects

E.1 Instructions

The twenty-six cases in this case set were drawn from successive adult hearing evals performed at the Methodist Hospital Speech & Hearing Laboratory in Houston. Your task is to select a diagnostic category based on the feature list at the top of the page. If you are confident in the diagnosis place a check-mark on the line in front of your choice of diagnostic category. If you are not confident in placing the case in a single diagnostic category, indicate your choice of "possibles" and rank them by putting a 1 on the line in front of the diagnosis you judge most probable, a 2 on the line in front of the diagnosis you judge next most likely, etc. Use as many alternative diagnostic categories as necessary for you to be comfortable with your evaluation.
E.2 Sample Page of Questionnaire

Case A
age greater than 60
mild-to-moderate sensorineural
bone-PTA(unmeasured)
air-PTA(mild)
speech intelligibility(good)
tympanogram type(A)
static pressure(normal)
acoustic reflex: uncrossed(normal)
acoustic reflex: crossed(normal)

opposite acoustic reflex: uncrossed(normal)
opposite acoustic reflex: crossed(normal)

Diagnostic Categories
  _acoustic_neuroma
  _bells_palsy
  _cochlear_age
  _cochlear_age_and_noise
  _cochlear_age_plus_poss_menieres
  _cochlear_noise_and_heredity
  _cochlear_poss_noise
  _cochlear_unknown
  _conductive_discontinuity
  _conductive_fixation
  _mixed_cochlear_age_fixation
  _mixed_cochlear_age_otitis_media
  _mixed_cochlear_age_ser om
  _mixed_cochlear_unk_discontinuity
  _mixed_cochlear_unk_fixation
  _mixed_cochlear_unk_ser om
  _mixed_poss_central om
  _mixed_poss_noise om
  _normal_ear
  _otitis_media
  _poss_central
  _possible_brainstem_disorder
  _possible_menieres
  _retrocochlear_unknown
Glossary of Protos Terminology

case – a featural description presented to Protos for classification.

category – the collection of the instances of a concept. It is the extensional representation of the concept.

category structure – Protos' network of domain knowledge including categories, their exemplars, and explanations relating them.

censor – a feature which, if present, tends to rule out a classification. Censors can be relative or absolute.

concept – the characterization of a set of physical or conceptual entities which are equivalent in some context (e.g., with respect to a goal).

conditional explanation – an explanation which is valid only in a specific context (e.g., an explanation of featural equivalence which is valid only in the context of a particular category).

constructive induction – an inductive learning process that uses domain knowledge to infer features not present in the initial description of a case.

criterial feature – a feature which plays an important role in determining category membership.

difference link – an index connecting two exemplars (in the same or different categories) which records important featural differences between them. The presence of a difference link between an exemplar and a near-miss
exemplar allows Protos to avoid making a plausible, but incorrect (or nonoptimal), classification.

**exemplar** – a case which Protos has processed and retained. It is embedded in the category structure with an accompanying body of domain knowledge (explaining the relevance of its features to the classification).

**explanation** – a chain of known relations that allows Protos to conclude the equivalence of two features or that a feature is relevant to category membership.

**family resemblance** – a measure of the prototypicality of an exemplar. An exemplar has family resemblance within a category to the extent to which it is similar to other category members.

**featural exemplars** – a feature of a case may itself be a category containing exemplars when viewed at a level of greater detail.

**features** – the labels for the attributes of a case that comprise its description.

**feature-to-category explanation** – an explanation of how the presence of a feature in a case increases the belief that its classification is a particular category.

**feature-to-feature explanation** – an explanation of the equivalence of two features.

**idiosyncratic feature** – a feature that is a unique characteristic of a particular exemplar.

**implicit generalization** – the acquisition of domain knowledge that allows knowledge-based pattern matching to match an exemplar to cases which
are transformationally more distant within a category. Implicit generalization does not change the declarative description of an exemplar; it changes the exemplar's range of coverage within the category.

**importance** — a rating of the necessity of a feature being present for a given classification to be made. Importance is an estimate of the conditional probability of the feature being present in exemplars of a given category.

**knowledge-based pattern matching** — a heuristic search procedure that uses domain knowledge within the category structure to construct explanations of featural equivalence or of the relevance of case features to a classification (see also transformational matching).

**learning apprentice** — a system that provides interactive aid in problem solving and learns in the course of its use.

**near miss** — an exemplar that was strongly, but incorrectly, matched by a new case.

**node** — a feature or category in Protos' category structure.

**prototype** — the most representative exemplar(s) of a category. A category may have multiple prototypes.

**prototypicality** — a rating of the representativeness of exemplars with respect to a category. Prototypicality is determined by family resemblance: an exemplar is prototypical to the extent to which it is similar to other category members.

**range of coverage** — the space of possible category instances that an exemplar can match via knowledge-based pattern matching of its features (given a particular state of the system's domain knowledge).
relational links – single-step inferences which relate two nodes in the category structure.

reminding – an index into the category structure associating a feature with a category (or a particular exemplar) that may be relevant to the classification of a case possessing the feature. Remindings are compiled from explanations of the relevance of features to categories or exemplars. The strength of a reminding estimates the conditional probability \( p(\text{category}|\text{feature}) \) or \( p(\text{exemplar}|\text{feature}) \).

satisficing search – search characterized by willingness to accept any solution to a problem which satisfies some minimum criteria.

similarity function – a method of computing the similarity of two cases.

spurious feature – a feature which, although it is present in the description of a case, is irrelevant to its classification.

subordinate category – a category which is more specific than a given category.

superordinate category – a category which is more general than a given category.

transformational distance – the amount of inference that is needed to relate two features or categories.

transformational matching – the process of finding a chain of known relations (i.e., an explanation) which relates features or categories. In Protos, transformational matching is performed by the knowledge-based pattern matching algorithm.
Glossary of Audiological Terminology

air – the degree of hearing loss for air-conducted tonal signals in the frequency region important for speech perception.

ar_c – indicates the presence of an acoustic reflex in the stapedius muscle of the opposite ear when a signal is applied to the test ear.

ar_u – indicates the presence of an acoustic reflex in the stapedius muscle of the test ear when a signal is applied to the test ear.

bone – the degree of hearing loss for bone-conducted tonal signals in the frequency region important for speech perception.

history – indicates that the patient has a past history of a given symptom.

notch_at_4k – indicates a hearing threshold that is at least 20dB lower at 4KHz than at 2KHz or 8KHz.

o_ar_c – indicates the presence of an acoustic reflex in the stapedius muscle of the test ear when a signal is applied to the opposite ear.

o_ar_u – indicates the presence of an acoustic reflex in the stapedius muscle of the opposite ear when a signal is applied to the opposite ear.

severity_sn_frequency – indicates a diminished cochlear or retrocochlear function (i.e., sensorineural loss) in a given frequency range. For example, m_sn_gt_2k denotes a mild sensorineural loss at frequencies greater than 2KHz.

speech – indicates the relative ability of the patient to identify spoken words.
**static** – indicates the pressure in the middle-ear airspace.

**tymp** – indicates the transmission characteristics of the eardrum and middle-ear system.
BIBLIOGRAPHY


1982.


VITA

Ellis Raymond Bareiss Jr. was born in Houston, Texas on August 3, 1951, the son of Janette W. and Ellis R. Bareiss. After graduating from Pasadena High School, Pasadena, Texas, in 1969, he attended San Jacinto College in Pasadena, Texas for one semester before transferring to the University of Texas. He received a Bachelor of Science degree in Communications in 1974. He returned for graduate study in Computer Sciences from 1977 through 1979. During that period, he also worked for the University of Texas Center for Energy Studies as a computer programmer. In the following years, he was employed as a software engineer at Tektronix, Inc. then as Director of Software Engineering at IMPRES, Inc. In January 1984, he returned to the University of Texas to complete his studies in Computer Sciences.

Permanent address: 4215 Greystone
Austin, Texas
78731

This dissertation was typeset\textsuperscript{1} with \LaTeX{} by Audrey Bareiss and the author.

\textsuperscript{1}\LaTeX{} document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The \LaTeX{} macro package for The University of Texas at Austin dissertation format was written by Rhe-Sing The.