KNOWLEDGE REPAIR

by

Satyajit Rao

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

Massachusetts Institute of Technology

May 1991

Author .....
Department of Electrical Engineering and Computer Science
May 22, 1991

Certified by .....
Patrick Henry Winston
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

KNOWLEDGE REPAIR

by

Satyajit Rao

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 1991, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

The Explanation Based Learning paradigm generates explanations in the course of problem solving. The explanations are retained as simple rule-like knowledge for future use. These rules provide speedup but no new knowledge because the explanations come from a preexisting unchanging domain theory. When the explanations are based on experience however, the domain theory at any moment is partial and there is no guarantee that it is sound. Consequently learned knowledge may prove to be faulty. This thesis makes the following contributions:

- It presents a knowledge repair mechanism for faulty knowledge acquired by the Explanation Based Analogical Learning paradigm. The repair mechanism first *isolates* the causes of failure, then it *explains* how they are responsible for the failure, finally it uses the explanations to *repair* the faulty knowledge. The key feature of the repair mechanism is its ability to use prior experience to perform the repair.
- It discusses how a machine could acquire *nameless concepts*; concepts that do not have a name attached but are defined either perceptually or by virtue of their relation to other concepts.

Thesis Supervisor: Patrick Henry Winston
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would first like to thank my parents, their emphasis on education made it possible for me to be here at M.I.T. My thesis advisor Patrick Winston deserves special thanks for giving me the opportunity to work with him since my undergraduate days. Many ideas in this thesis were a result of direct collaboration with him.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research is provided in part by the Office of Naval Research University Research Initiative Program under Office of Naval Research contract N00014-86-K-0685 and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

Contents

List of Figures

Chapter 1

Introduction

1.1 The problem

The Explanation Based Learning paradigm generates explanations in the course of problem solving. The explanations are retained as simple rule-like knowledge for future use. These rules provide speedup but no new knowledge because the explanations come from a preexisting unchanging domain theory. When the explanations are based on experience however, the domain theory at any moment is partial and there is no guarantee that it is sound. Consequently learned knowledge may prove to be faulty. For example the following cup classification rule learned via experience is not quite adequate.

If:
- The object has a bottom
- The bottom is flat
- The object has a concavity
- The object is light weight
- The object has a handle

Then:
- The object is a cup.

Provided:
- The object is stable
- The object enables drinking
- The object carries liquids
- The object is liftable

The rule correctly classifies the cups in figure 1.1 but it also mistakenly classifies the bucket and the trough as cups because it does not capture the function of the cups handle. Furthermore it does not classify the cups in figure 1.2 because it insists that cups have handles.

The cup classification rule is clearly faulty, it needs to be repaired.

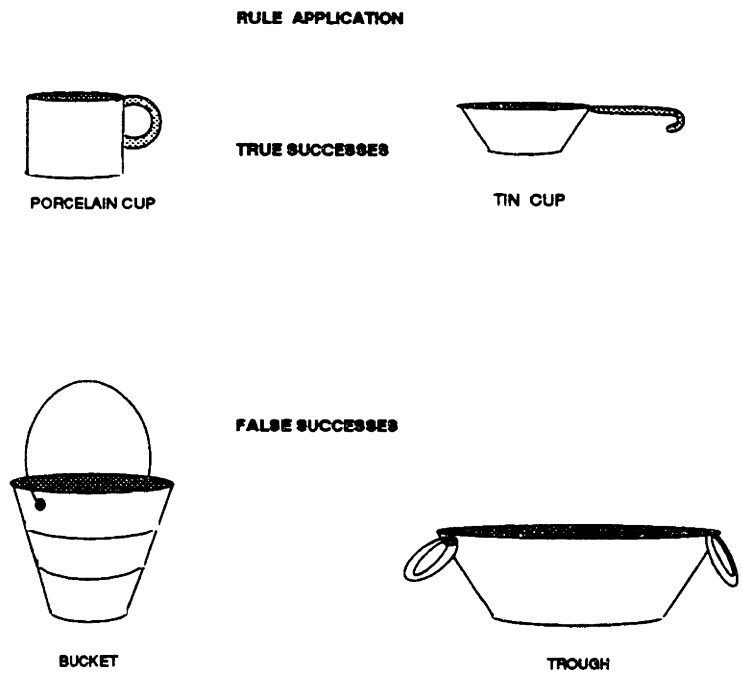

Figure 1.1: The cup rule classifies all these objects as cups. The cups are called *true-successes* and the bucket and the trough are called *false-successes.*

Figure 1.2: The cup rule does not consider these as cups because it requires cups to have fixed handles. The rule is too specific.

1.2 The solution

The following repaired cup rule correctly classifies the cups in figures 1.1 and 1.2. It no longer classifies the bucket and trough as cups.

If:
The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight
nameless-node-497
Then:
The object is a cup.
Provided:
The object is stable
The object enables drinking
The object carries liquids
The object is liftable
The object is orientable
The object is manipulable

The *nameless-node-497* relation represents a concept that does not have an english name but is defined by its function in the new cup rule and the following two auxiliary rules:

If:
The object has a handle
Then:
nameless-node-497

If:
The object is handsized
Then:
nameless-node-497

The following chapters explain how the faulty cup rule was repaired to obtain the new rule.

1.3 What this thesis is about

This thesis makes two contributions:

- It presents a knowledge repair mechanism for faulty knowledge acquired by the Explanation Based Analogical Learning paradigm. The repair mechanism first *isolates* the causes of failure, then it *explains* how they are responsible for the failure, finally it uses the explanations to repair the faulty knowledge. **The key feature of the repair mechanism is its ability to use prior experience to perform the repair.**

- It discusses how a machine could acquire **nameless concepts**. These concepts that do not have a name attached but are defined either perceptually or by virtue of their relation to other concepts.

1.4 The importance of Machine Learning

Learning has long been the holy grail of Artificial Intelligence. The possibility of not having to handcode large amounts of knowledge but have the system learn it is an alluring one. Even more seductive is the idea that once a system has learned something, the knowledge can be almost instantaneously transfered to other systems. Unlike humans each AI system will not have to start learning from scratch.

The *utility* value of having a machine learn to recognize faces, or use natural language or identify disease causing sections of DNA is obvious. However, as Rivest [9] points out there are reasons other than utility for studying machine learning. One goal is simply *self-knowledge*. To gain an understanding of how we humans learn by trying to model the process on a computer. Another viewpoint is that learning is a fascinating phenomenon in its own right and therefore worthy of study.

1.5 Some background about research in Machine Learning

Traditionally researchers have distinguished between two forms of learning: symbolic concept acquisition and skill acquisition, to distinguish between the tasks of say learning to recognize arches and and learning to ride a bicycle. Within the Artificial Intelligence community most of the efforts have been devoted to symbolic concept acquisition.

Research in symbolic concept acquisition can be classified along three different directions [2] : the learning paradigm used, the representation used for the learned knowledge and the training examples, and the domain of application.

Some of the learning paradigms are: rote learning, learning by instruction, learning by analogy and learning by examples. The examples come from either the teacher, the system itself or the environment. These paradigms can be further classified according to the amount of initial information provided to the system, and the extent of help provided by the teacher. This thesis uses the Explanation Based Analogical Learning (EBAL) paradigm which is described in the next chapter.

Typically the objects in the instance space of the concept are represented by a finite set of Attributes (e.g. color, weight) where each attribute can take on a finite set of values (e.g blue, green or red for the color attribute). The representation of the concept must be able to map an object of the instance space to either *true*—if it is an instance of the concept, or *false*—if it is not an instance of the concept, or to *unknown* in some cases when there is not enough information. Some concept representations in use are:

- Numerical weights and thresholds in neural nets that form an implicit representation of the concept.
- Decision trees like the ones generated by Quinlan's ID3 algorithm [8] classify an instance into one of several categories.
- Formal grammars are a compact representation for a possibly infinite set of strings. The target instance is checked to see if it could be generated by the grammar.
- Production rules consist of a pair $\langle CA \rangle$ of conditions and actions. If the set of conditions C are all found to be true then the rule fires and the set of actions A are executed. They are also sometimes called IF-THEN rules. For example here is an IF-THEN rule that attempts to capture the concept of an *arch*:

If

A is a rectangular block
B is a rectangular block
C is a rectangular block
C is on A and B
A and B are not touching

Then

The object is an arch

The above IF-THEN rule for an arch while sufficient for arches of the type shown in figure 1.3 clearly does not capture the concept of arch. For example the arches in figure 1.4 will not be recognized as arches by the rule. The point being made here is that the first attempt to capture a concept can rarely hope to be successful. It will usually be only an approximation to the desired concept. When presented with the arches in figure 1.4 the learner is forced to revise the now obsolete and incomplete IF-THEN rule derived from figure 1.3. Learning therefore has to be a *continuous process.*

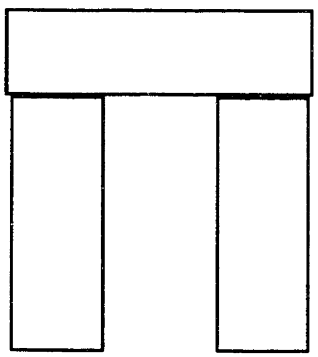

Figure 1.3: An arch

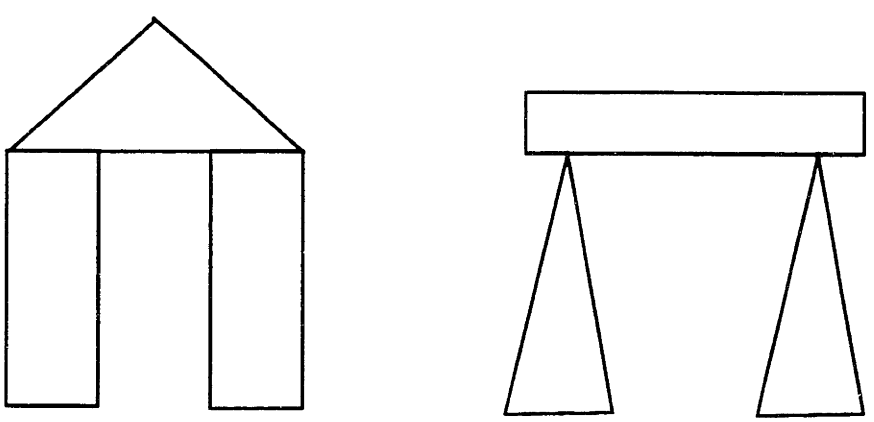

Figure 1.4: More arches

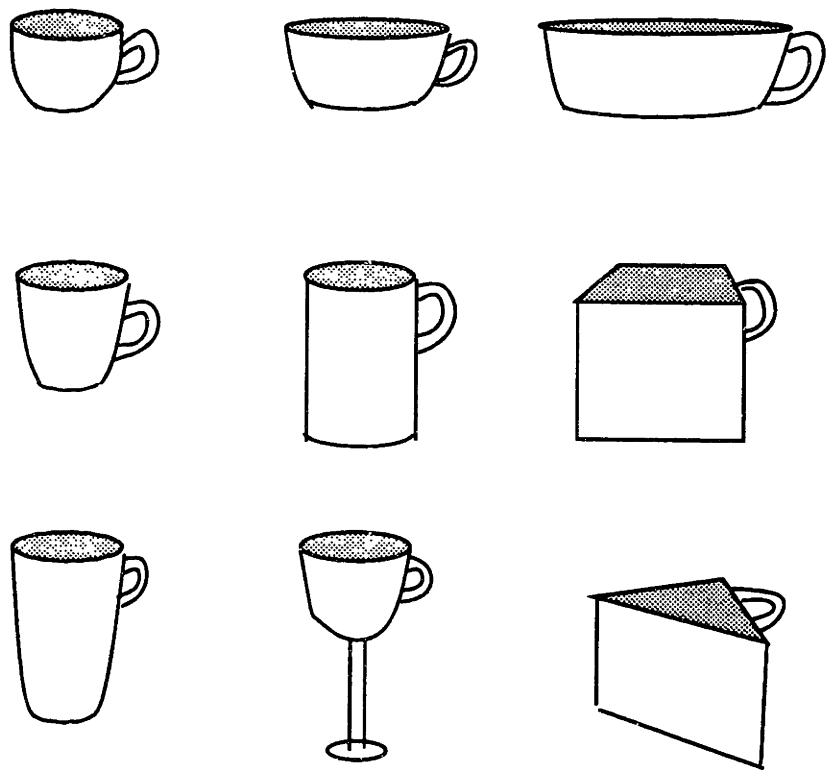

Figure 1.5: Subjects were asked to imagine the objects in various circumstances; in someones hand, or containing food or containing flowers. The structure and function of these objects influenced the names given to them. An object was named as a certain type of container if it had *either* the right perceptual representation, *or* the right functional properties.

1.6 The motivation for continuous learning

One can never afford to stop learning because:

- *Learning is inaccurate* Assuming that the concept to be learned is well defined (as a collection of some finite set of attributes) the learned concept may be inaccurate simply because of insufficient data or a bad sample.
- *Learning is never complete*
  - Concepts are imprecise: A more realistic situation in concept acquisition is when the concept to be acquired is not well defined. Figure 1.5 shows a collection of cup-like objects. Labov tried to find out how shape variations influenced the naming of an object and whether the function of the objects also influenced their naming. The subjects were asked to imagine the objects in various circumstances - in someones hand, or containing food or containing flowers. Labov [3] found that the names given to the objects (cup, mug, bowl, vase) differed not only with the shape but also with the

function performed by the objects. Labov's work showed that an object can be named as a certain type of container if it *either* has the right perceptual representation, *or* the right functional properties. Also no single part of a concept need *always* be present to make the object a member of a particular category, i.e a rule for a concept need not have any necessary conditions. This example while introducing the roles of structure and function in categorization also demonstrates that there may be no absolute definition for many human concepts.

- A changing world makes old knowledge obsolete: A medical diagnosis program which is trying to learn the symptoms of a disease, must keep changing its concept of the disease as the bacteria or virus mutates.

Continuous learning is therefore essential for any system that is going to deal with the real world. Given the need for continuous learning, how does one realize it? When learned knowledge is inaccurate or incomplete, we could either relearn it from scratch or try to repair the faulty chunk of knowledge. This thesis advocates the later approach.

1.7 Continous learning via knowledge repair

The knowledge repair approach is preferred to re-learning everything from scratch because it is incremental. An incremental solution is preferable because:

- Computational common sense says that throwing away faulty knowledge and starting over is expensive and wasteful.
- A subtler reason is that other knowledge may depend on the faulty chunk of knowledge. Gross changes to the faulty knowledge may unintentionally break these dependencies. We want to avoid throwing the baby out with the bathwater.
- Sharp discontinuities in the behavior of the program are undesirable. This would happen if the new solution was radically different from the old one.

1.8 Traditional approaches to repair of IF-THEN rules

When the rule is found to be too special it is *generalized.* Generalization is usually performed by:

1. Dropping the overly-specific IF conditions from the rule, assuming of course that the overly-specific IF conditions can be identified somehow.

2. Replacing some of the categories used in the rule with more general categories. This assumes that the knowledge of category hierarchies exists and is available. For example the faulty IF-THEN arch rule could be generalized by replacing the *rectangular-block* category with the *polyhedron* category. This is also known as the *climb-tree* heuristic in the literature because it involves climbing the category hierarchy tree.

Rules are *specialized* in a symmetric manner by adding IF conditions and by descending the category hierarchy tree.

1.9 This thesis proposes a new repair mechanism

The learning paradigm used in this thesis is a mixture of Explanation based Learning and Learning by Analogy. The learned concept is represented as a production rule. The problem that this thesis addresses is:

Given:

- A set of precedents (prior experience).
- A benign teacher.
- A faulty concept.

Is it possible to repair the faulty rule using the precedents?

1.10 Repair Biases

The repair mechanism that is proposed has the following biases:

- It is *incremental* for the reasons stated in section 1.7.
- It is *driven by failure.* Don't fix something unless it's broken.
- It is *Guided by experience.* In the absence of a sound domain theory, experience can be used to guide the repair process.

The repair mechanism proposed in this thesis differs from the traditional repair methods in that it heavily emphasizes the use of *prior experience.*

1.11 Outline

- Chapter 2 describes the explanation based analogical reasoning system, previously described in [7].
- Chapter 3 describes the specialization mechanism of rule-repair. The material in this chapter is described in [14].
- Chapter 4 describes the generalization mechanism of rule-repair.
- Chapter 5 describes the formation and utility of nameless concepts.
- Chapter 6 is an assessment of the repair process.

1.12 Summary

Machine learning is an important area of research in A.I. Most of the work has focused on *concept learning*. Learning however is never complete. Once a concept has been learned it is continuously refined. Concept repair is at least as important as concept acquisition. The traditional repair methods make changes to the superficial representation of the concept. Assuming that concepts can be represented by simple rule-like knowledge, this thesis proposes a repair mechanism which effectively exploits prior experience. The thesis also demonstrates how a nameless concept can be formed by virtue of its relation to other named concepts.

Chapter 2

The Explanation Based Analogical Learning system (EBAL)

2.1 The domain

The domain of descriptions that relate the structures of objects to their functions has been chosen to illustrate the workings of the EBAL system and the repair process. The interaction between structure and function in categorization has already been introduced in section 1.6 in the discussion of the Labov experiment which showed that an object can be a named as an instance of a particular category if it *either* has the right perceptual representation, *or* the right functional properties.

Figure 2.1 shows different kinds of arches. No single definition that is purely *structural* could capture these arches. Structural definitions are too limiting. They are too specific.

The arches have a common *functional* property in that you could push a toy car through them, however you would have to change your grip of the car from one hand to the other at some point. The problem with a purely functional definition is that firstly it would apply to too many structures that are not arches (a hoop for example), i.e it is too general. Secondly it is too abstract to use, because it does not specify *how* a structure could satisfy the functional property, i.e it is not *operational.*

The solution then is that we need an arch concept that is a *bridge-definition* [5]. Bridge-definitions explain how certain structures enable certain functions. Bridge-defintions *bridge* the gap between structures and the functions that they perform. This domain will be particularly important in the future as we make progress in Vision and Natural Language Understanding. Object recognition programs when given the task of picking an object from the environment that can be used for sitting on, will need to know the relationship between structure and function of an object in order to perform the task. To illustrate the importance of this domain in the area of understanding natural language, here is an example from Marr [4]. When we read

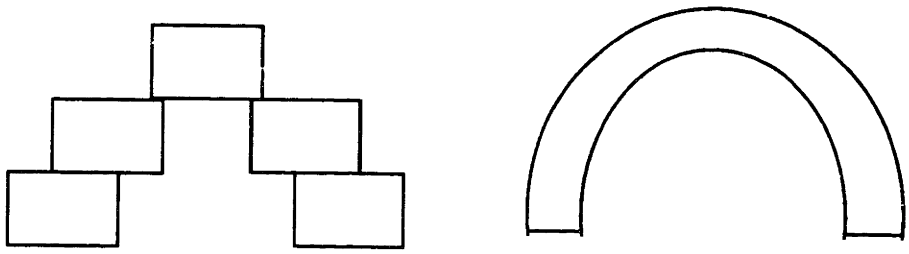

Figure 2.1: Arches that have a functional similarity

> The fly buzzed irritatingly on the windowpane. John picked up the newspaper.

We may infer that the newspaper is about to be used to swat the fly. To place a similar interpretation on John's actions a program that understands written text would have to know the relationship between the structure of a newspaper and its function as a fly swatter. In this situation the other functions of the newspaper such as its news content or its combustible nature are irrelevant.

2.2 Precedents

Precedents represent the prior experience of the system. Each precedent contains some information about how the structural aspects of a particular object are related to its functions. Figure 2.2 shows some precedents and the contents of one of them; the glass precedent.

The representation that links the structure of the glass to its function is called an *and-tree*. An and-tree is therefore a kind of bridge-definition. Each arrow in the and-tree represents a causal relation and each node is a property of the particular glass, caused by the nodes immediately under it. Note that some relations like **`Glass is transparent`** appear by themselves, these will be called *free-floating relations*. The glass precedent is shown with only one and-tree, however a precedent may contain more than one and-tree. Different precedents may contain different and-trees for the same relation. For example in the glass precedent the glass is stable because it has a bottom and because the bottom is flat. However, in another precedent an object may be stable for very different reasons. For instance a block of wood is stable when firmly clamped in the jaws of a vice. This idea of a particular functional property having different structural explanations in different precedents is essential to the workings of the EBAL paradigm and the repair process as you shall soon see.

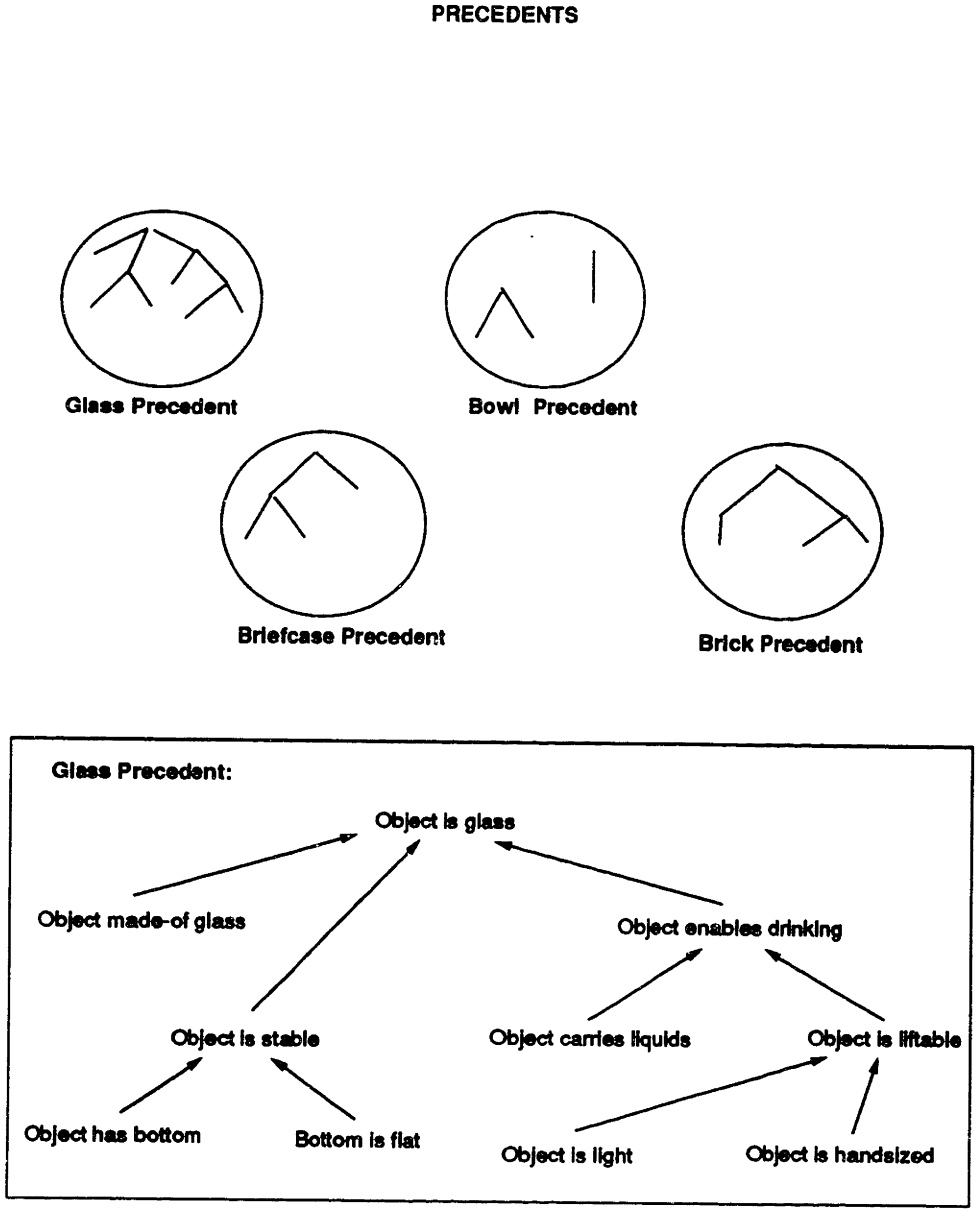

Figure 2.2: Some examples of precedents and the contents of one of them; the glass precedent.

2.3 Problem solving

The teacher selects an exercise for the system. An exercise consists of a group of free-floating relations and a query. The query is simply a relation which the system must try to derive from the free-floating relations. Problem solving consists of bridging the gap between the query relation and the free-floating relations by gluing together and-tree structures from various precedents. The resulting and-tree; an explanation of how the query relation follows from the free-floating relations of the exercise can be viewed as a *recollection* of knowledge because that is precisely what it is—a recollection of knowledge from many precedents.

Figure 2.3 shows a problem and the result of problem solving. The problem is to show that an object is a cup by connecting the relation `Object is cup?` to the facts known about the object by gluing together and-tree fragments from other precedents. The resulting explanation, a gap filling and-tree, is shown in the bottom half of the figure along with the names of all the precedents that contributed to the solution. Note that the leaves of the gap filling and-tree are the previously free-floating relations of the exercise.

2.3.1 The analogical and explanation based nature of problem solving

There is no domain theory which allows us to conclude that if a glass is stable because it has a flat bottom, then a different object, a cup, must also be stable if it has a flat bottom. Nevertheless the system makes the assumption. Thus the system is using *analogy* to make inductive assumptions. The problem solving is *explanation based* because the result of problem solving *explains* why the object in the exercise is a cup. However this paradigm differs from traditional Explanation based Learning in that the explanations are constructed by analogy from prior experience which certainly can not be considered as sound domain theory.

2.3.2 The choice of the exercise is important

The system has no way of knowing which recollections of knowledge in the precedents are useful. By choosing the exercises carefully the teacher guides the system towards useful recollections of knowledge. The tacit assumption made is that the teacher will choose exercises that result in recollections that capture some commonly occurring regularities of the world.

2.4 Rule formation

The retained recollection is also maintained in the form of an IF-THEN-PROVIDED rule for the purpose of future application.

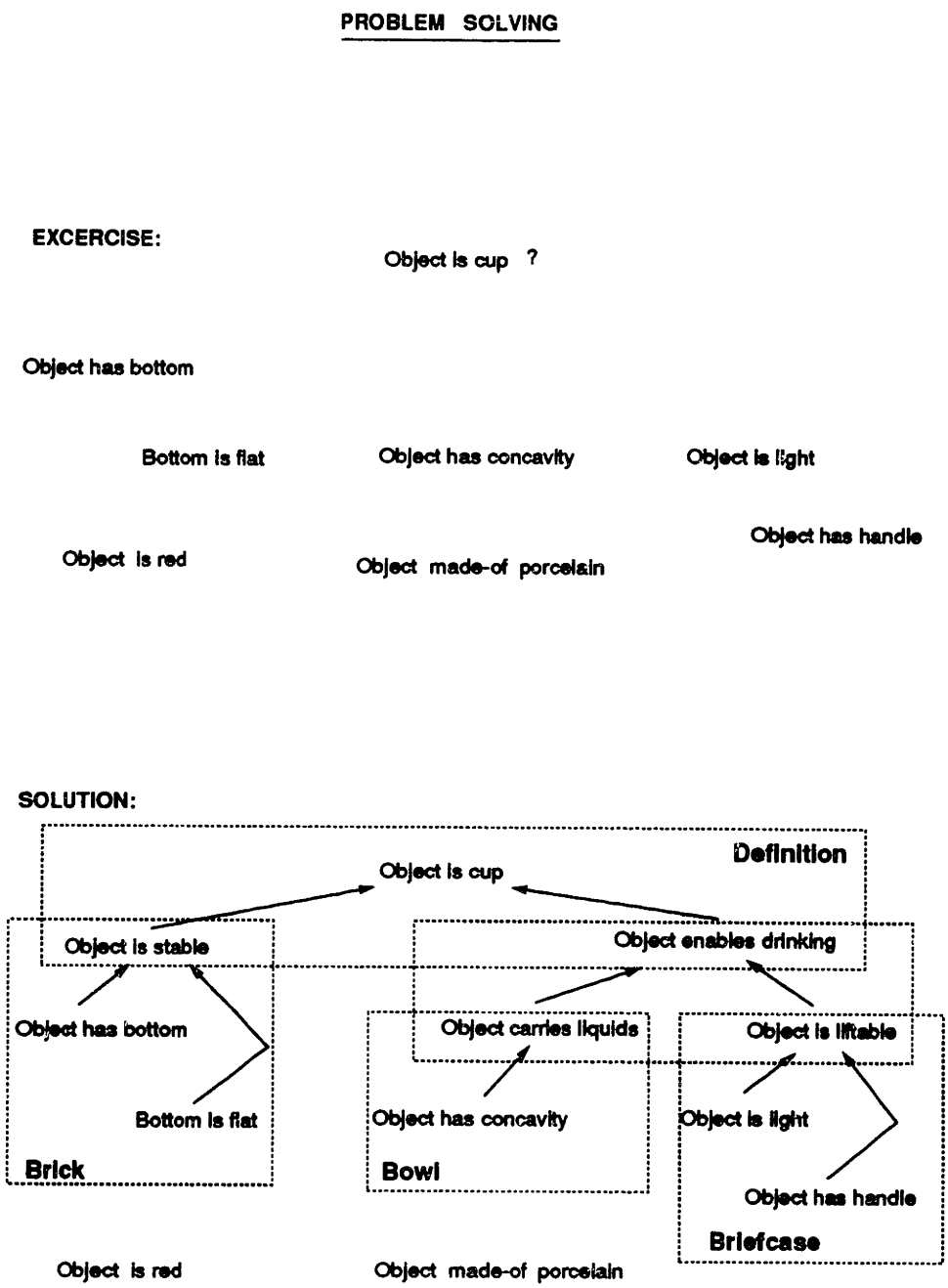

Figure 2.3: The EBAL system solves a problem by pooling together chunks of knowledge from several precedents.

If:

The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight
The object has a handle

Then:

The object is a cup.

Provided:

The object is stable
The object enables drinking
The object carries liquids
The object is liftable

The leaves of the cup recollection become IF conditions, the root of the recollection becomes the THEN condition and all the intermediate nodes become the PROVIDED conditions. There is no logical difference between the IF and PROVIDED conditions with respect to making the THEN condition true, however there is an operational difference. This operational difference is relevant only to rule application and will be discussed in the next section. It should always be kept in mind that the IF-THEN-PROVIDED rule representation is just the superficial structure of the rule and that the deeper structure, the rule and-tree, is also maintained with the rule. From now on whenever we talk of rules it will be assumed that the rule and-tree is also accessible.

In summary, the problem solving recollection (the rule and-tree) is retained because it represents *new knowledge*. It is also maintained in the form of a rule so that the new knowledge can be *applied* as a one step deduction.

2.5 The new rule embodies new-knowledge - not just speedup

The recollection obtained as a result of problem solving is retained because it represents new knowledge. To drive home this point consider the nonsense recollection show in figure 2.4. Every individual piece of this recollection when considered in isolation makes sense but the whole recollection when taken together is nonsense because there is no known cup that is a cup because of the reasons in the recollection. The point is that arbitrary recollections of knowledge do not make sense because the corresponding situations do not occur in the world. Given a situation that does occur in the real world, the recollection of knowledge that is produced by problem solving is meaningful because it represents one way in which chunks of knowledge can come together and explain a real situation, as opposed to all the other nonsensical ways that knowledge can be combined. The solution captures a regularity of the world and hence it bears remembering.

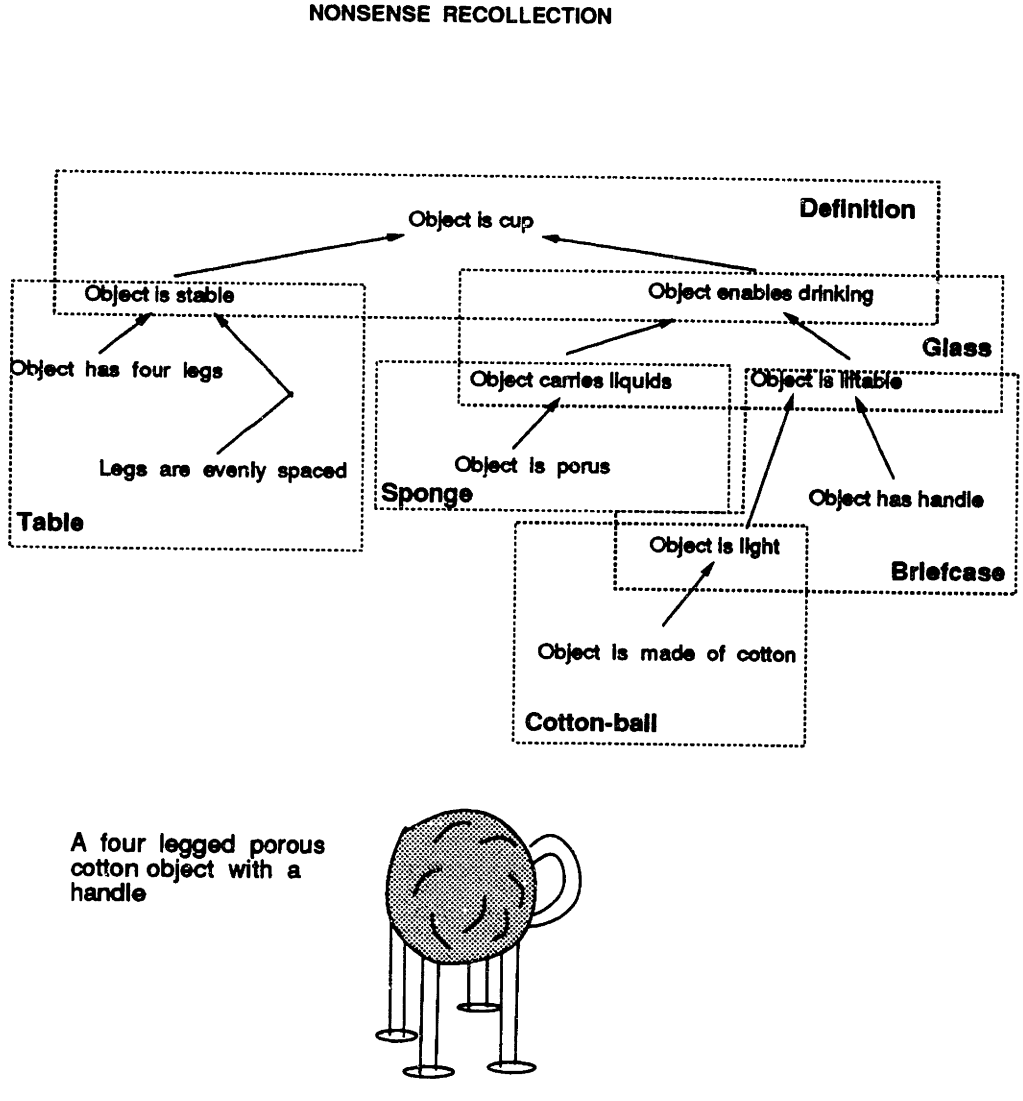

Figure 2.4: Individual pieces of this recollection make sense but the recollection as a whole is nonsense. There is no object which is a cup because of the reasons given above.

Another reason why retaining the recollection from problem solving does not provide just speedup is that given that there are practical limitations on the amount of effort that can be expended in solving a problem, using previous solutions brings some problems over the computational horizon. Now some problems can be solved that could not previously be solved because of computational limitations.

Yet another reason for retaining recollections is communication. In many domains it is very important for the teacher to know what it is that the system has learned so far about a particular concept, the concept of a cup for instance. If the system fragments all its knowledge into little pieces then it would just hand back the cup definition that it was initially provided by the teacher. The knowledge of how this definition grounds out into structure would have been lost by the fragmentation.

2.6 Rule Application, Operationality, and Censors

Knowledge is useless unless it can be applied. It follows therefore that it should be represented in a way that makes application easy. This is the purpose of the IF-THEN-PROVIDED representation of the rule and-tree. It permits the application of the rule as a one-step deduction.

Censors are rules that are just like any other rule except that they have negative THEN conditions. Their sole purpose is to prevent rules from firing by falsifying one of the provided conditions. The justification for maintaining censors as separate rules instead of incorporating them into the body of the main rule is given by what Minsky [6] calls *the exception principle*:

> It rarely pays to tamper with a rule that nearly always works. It's better just to complement it with an accumulation of specific exceptions.

and

> Unless we treat exceptions separately, they'll wreck all the generalizations we may try to make.

Rule application works in following manner: Given an exercise, the IF conditions of a rule are checked to see if they are true. If all the IF conditions of a rule are true, and it is a censor, then it fires, asserting it's negative THEN condition. If it is not a censor but a normal rule and all the IF conditions are true then we check to see if any of the PROVIDED conditions have been falsified by any censor, if not the rule fires and asserts the THEN condition. Note that we do not check for censors of censors. This is an arbitrary decision motivated by the feeling that exceptions of exceptions are too rare to bother about. The key idea in rule application is operationality, so that the rule can be easily applied as a one step deduction.

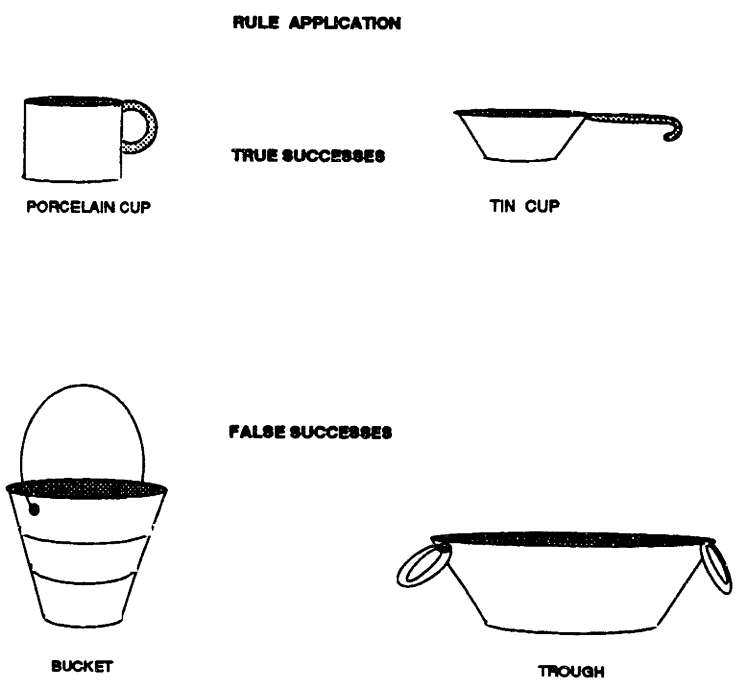

Figure 2.5: The cup rule classifies all these objects as cups. The cups are called *true-successes* and the pail and the trough are called *false-successes.*

Figure 2.5 shows some examples on which the cup rule is applied. The rule fires on the cups as it should but it also classifies the pail and the trough as cups. This brings us to the main topic of this thesis.

2.7 The general problem of repair

Given

- A set of precedents.
- A faulty rule—A rule that fails on a particular example.
- A benign teacher who can help the system isolate possible causes of failure.

Is it possible to repair the faulty rule *using the precedents*?

The emphasis will be on finding a repair mechanism that can use precedents instead of resorting to ad-hoc measures like adding and deleting arbitrary IF conditions.

2.8 The overall approach to repair

- Correlation **isolates** suspicious relations.
- Precedents **explain** how suspicious relations should specialize the rule.
- Explanations are used to **repair** the rule and-tree.

The three stages of isolation, explanation, and repair characterize the repair process that is described in this thesis. Both specialization and generalization repair processes have these stages in common. They only differ in the specifics of what is done at each stage.

We now consider each of these stages for the specialization mechanism of rule repair.

2.9 Summary

The domain of descriptions that relate the structure of objects to their functions was chosen to illustrate the Explanation Based Analogical Learning paradigm. Precedents represent the prior experience of the system and are used to solve problems. The result of problem solving—a re-collection of knowledge is retained because it embodies new knowledge and because its rule representation provides speedup. Learning is never complete and the new rule might prove to be inadequate when it is applied. This brings us to the issue of repair. The overall approach to repair will be to first *isolate* suspicious relations, then explain why they may be related to the failure of the rule, finally the explanations are used to repair the rule.

Chapter 3

Specialization

3.1 Correlation

The purpose of correlation is to isolate possible causes of failure of the rule. Figure 3.1 Shows the intersection of the relations of the two true-successes and the two false-success instances. The center region of the Venn diagram must contain the IF conditions of the faulty rule because it classified all four objects as cups. The top shaded region represents relations common only to the true-success instances. They will henceforth be called *true-success suspicious relations.* These relations may somehow be essential to the "cupness" of a cup and the rule might have failed by not incorporating some of these relations, thus leading to an overly general rule which classifies pails and troughs as cups. Correspondingly the bottom shaded region represents relations common only to false success instances. They will henceforth be called *false-success suspicious relations.* These are relations which may be essential to the the "non-cupness" of the pail and the trough. The rule misfired on the pail and trough because of not being censored by another rule which incorporates the knowledge of *how* the false-success suspicious relations contribute to "non-cupness" of the pail and trough.

As will be seen later, the *type* of a suspicious relation, i.e whether it is a true-success or a false-success suspicious relation determines the repair strategy for that relation. Before we go to the next stage of repair it is instructive to consider an alternative to the repair mechanism that is about to be proposed.

3.2 Quick fix - a zero-knowledge solution

A simple way to repair the rule once the suspicious relations have been isolated by correlation would be add all the true-success suspicious relations to the IF conditions of the rule, and make censors out of all the false success suspicious relations. The following new rules would be the result of such a repair.

Specialized rule:

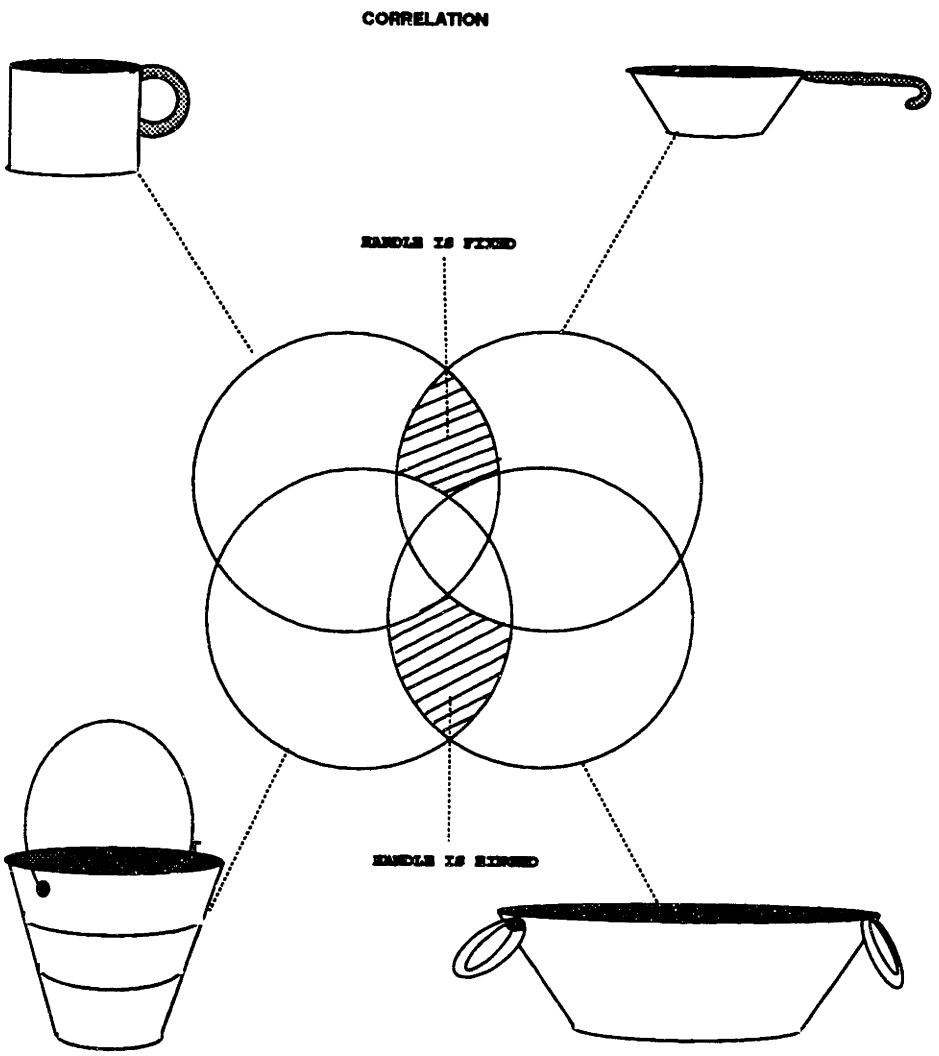

Figure 3.1: Isolation of suspicious relations by correlation.

If:

The object has a handle
The handle is fixed
The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight

Then:

The object is a cup.

Provided:

The object is stable
The object enables drinking
The object carries liquids
The object is liftable

New censor:

If:

The handle is hinged

Then:

The object is *not* a cup.

The new rule now is specialized by the **Handle is fixed** relation in its IF conditions, hence it will not fire on the pail and the trough. The rule is further specialized by being censored by the **Handle is hinged** relation. So, for the moment it seems to have been repaired. However this repair is not robust, the changes to it are cosmetic and the rule will soon run into new problems. The Quick Fix is a **zero-knowledge** solution that does not work for two reasons

- The Quick Fix is altering only the superficial structure of the rule, it is not considering the rule and-tree which is the real structure in the rule.
- It is also not making use of any knowledge in the precedents.

We *can do better* than this.

3.3 A preview of the result of repair

The actual repair mechanism proposed uses the and-tree structure of the rule and uses knowledge from the precedents. Figure 3.2 compares the rule and-tree before and after repair. Note that

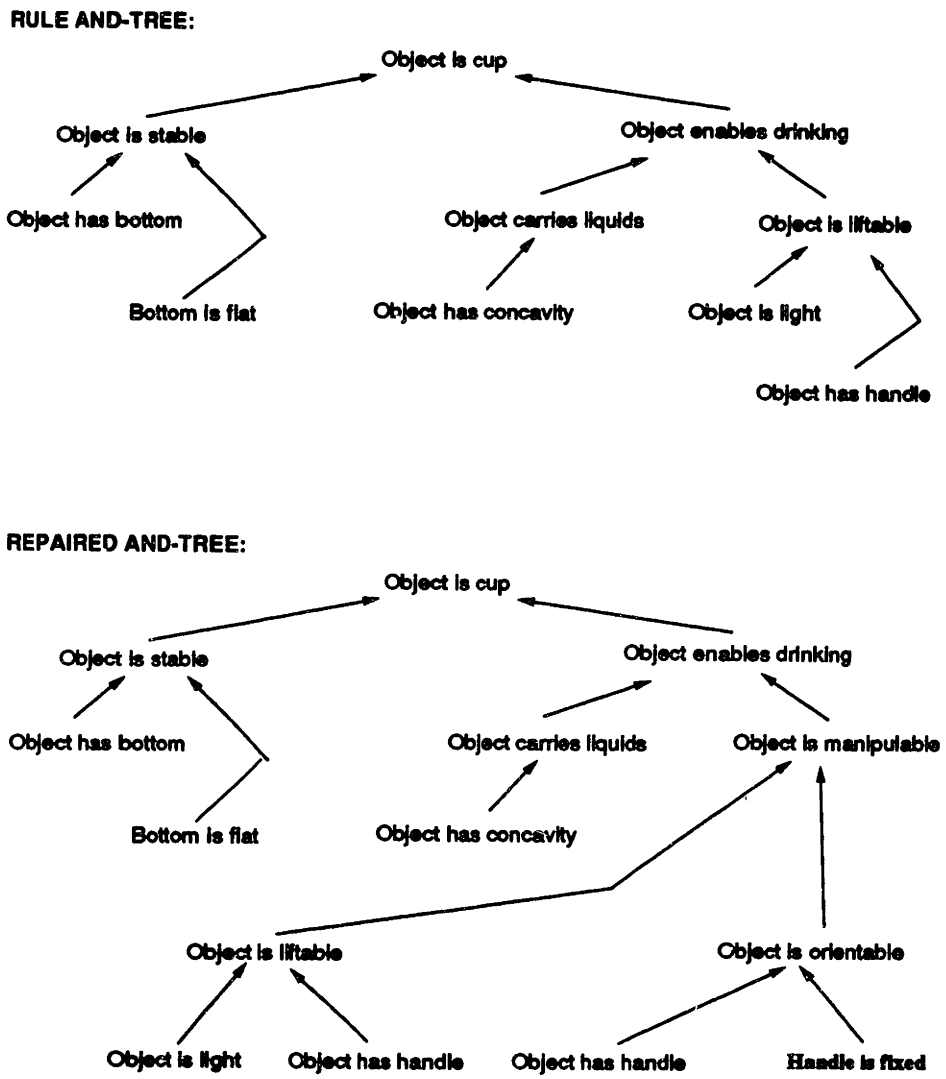

Figure 3.2: The old and new rule recollections.

- Most of the old-tree has been preserved, therefore the change is incremental as desired.
- The recollection for the relation **Object enables drinking** is different in the new rule and-tree. The new recollection includes The true success suspicious relation **Handle is fixed** as a leaf, and there are two entirely new intermediate relations **Object is manipulable** and **Object is orientable**.
- The old recollection was too general because one cannot be sure that one can drink from an object just because it carries liquids and is liftable. It has to be orientable by virtue of having a fixed handle. The new recollection still recognizes cups but its increased specificity prevents it from misclassifying the pail and trough as cups.

3.4 Specialization with true success suspicious relations

As mentioned before the type of the suspicious relation determines the repair strategy.

3.4.1 Explaining the cause of failure

The rule may have failed because it does not incorporate the relation **Handle is fixed** which may somehow be essential to the "cupness" of the cup. In order to explain why **Handle is fixed** may be essential to the "cupness" of the porcelain and tin cups, we need to find a recollection from the precedents that can bridge the gap between the **Handle is fixed** relation and any relation in the rule. Such a recollection could be spliced into the rule replacing the old one, thus incorporating the **Handle is fixed** relation into the rule.

The task of the repair algorithm is to find a new recollection for a rule relation that includes the true-success suspicious relation **Handle is fixed**. A priori, the algorithm doesn't know which node of the rule and-tree has to be repaired so it considers all of them in a breadth first bottom up manner.

Figure 3.3 shows the algorithm trying to ground the **Object enables drinking**, **Object has bottom**, and **Object carries liquids** relations in the set of all relations under them in the and-tree and the relation **Handle is fixed**. Only recollections that contain **Handle is fixed** will be accepted. Finding such recollections may require a considerable amount of search in the precedents. The algorithm limits the re-explanation process to a small subset of the total set of precedents. The initial subset of precedents considered is indicated in the circle in figure 3.3. The initial precedent set is the union of two sets of precedents called the *top* and *bottom* sets, which are formed in the following manner:

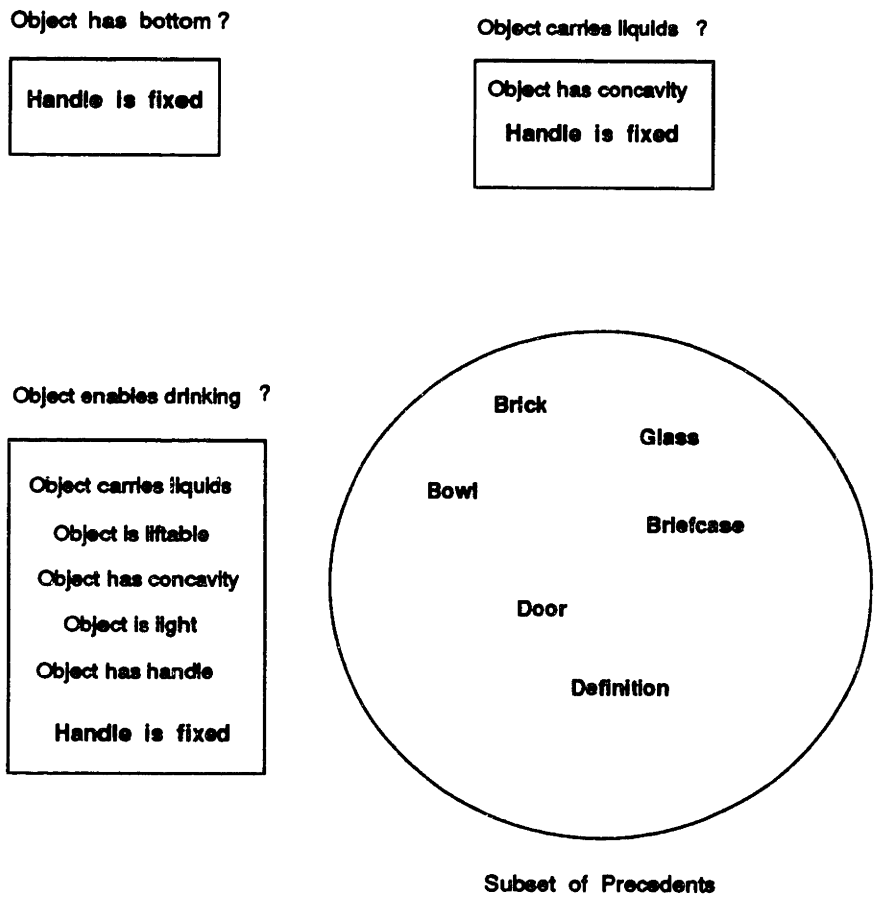

Figure 3.3: Algorithm tries to reexplain rule relations. Only explanations containing the `handle is fixed` relation will be accepted

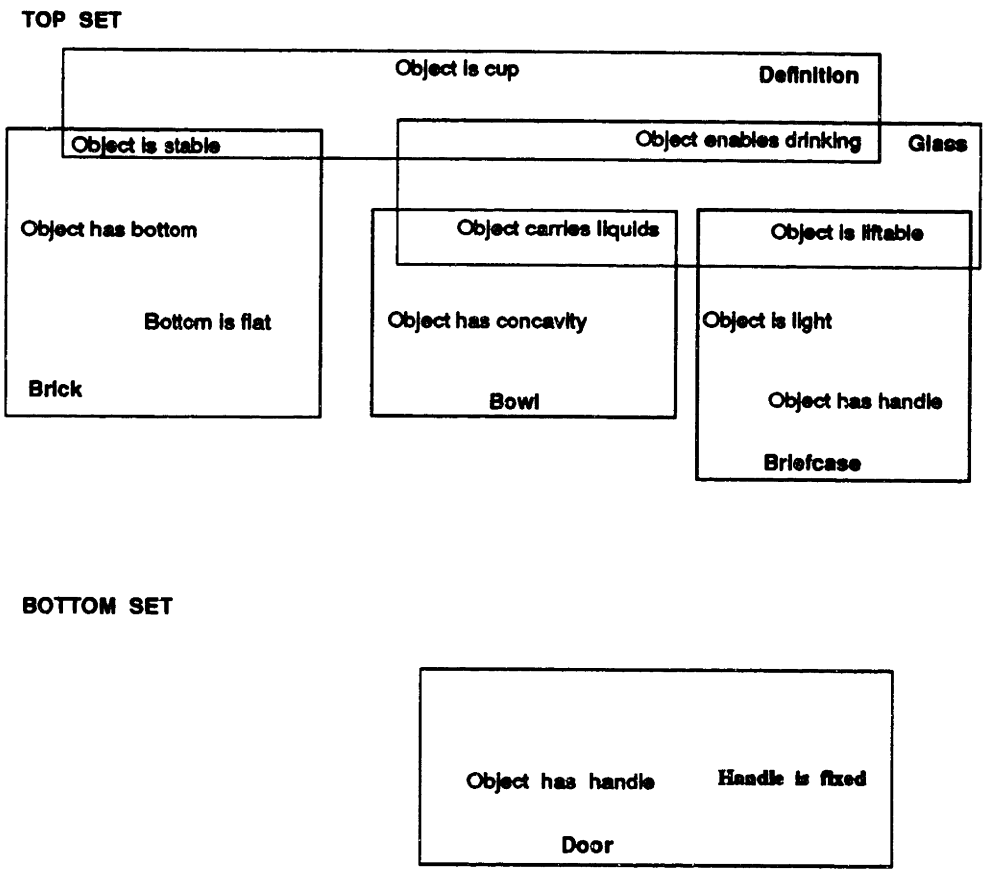

Figure 3.4: The sources of the precedents in the initial top and bottom sets.

- The precedents originally used to form the rule and-tree constitute the initial *top set.*
- The precedents in which one of the true-success suspicious relations causes something constitute the *bottom set.* Among the precedents the door precedent is the only one where **Handle is hinged** causes something. Therefore it is the only precedent in the bottom set.

The attempt to find a new explanation that includes **Handle is hinged** in the initial set of precedents is unsuccessful because the top and bottom sets are not connected in figure 3.4, i.e they do not share any common precedents. The conclusion is that more precedents have to be considered. Either the top or bottom set, whichever is smaller is expanded.

The top set is expanded by pulling in new precedents in which there is a relation with two properties: The relation must cause something in an existing top set precedent, and the relation must be caused by something in the new precedent. Thus the new top set precedents extend the causal chain of the top set downwards.

The bottom set is expanded by pulling in precedents in which there is a relation with two properties: The relation must be caused, in part, by something in an existing bottom set precedent; and the relation must help cause something in the new precedent. Thus the bottom set extends the existing causal chain in the bottom precedents upwards. The set with the fewer number of precedents is augmented to limit the number of precedents in the top and bottom sets. Figure 3.5 shows that the straw precedent is added to the bottom set. The straw precedent is added because it contains the relation **Straw is orientable** which causes **Straw is manipulable** and because in the door precedent **Handle is fixed** causes **Door is orientable.**

The re-explanation process starts again with the augmented bottom set. This time the algorithm succeeds in finding a new recollection for the rule relation **Object enables drinking** that includes the true-success suspicious relation **Handle is fixed.** The new recollection is shown in figure 3.7. Having successfully incorporated the only true-success suspicious relation into a new recollection, the explanation must now be used to repair the rule and-tree. We come to the final stage of repair with true-success suspicious relations; using the explanation (the new recollection) to repair the rule-and tree.

3.4.2 Repair with the new recollection containing the true-success suspicious relation

The repair is very simple, the new recollection for **Object enables drinking** shown in figure 3.7 is simply spliced into the rule and-tree in place of the old recollection for **Object enables drinking**, yielding the repaired and-tree shown

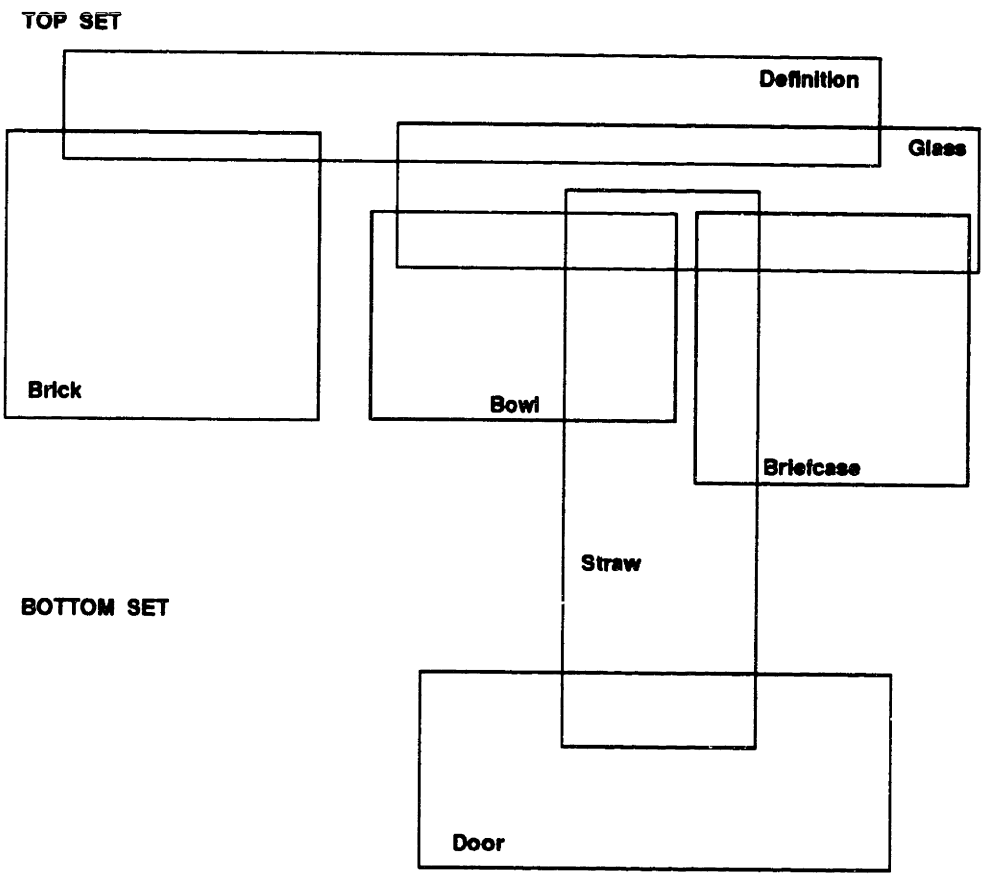

Figure 3.5: The bottom set is augmented by the straw precedent.

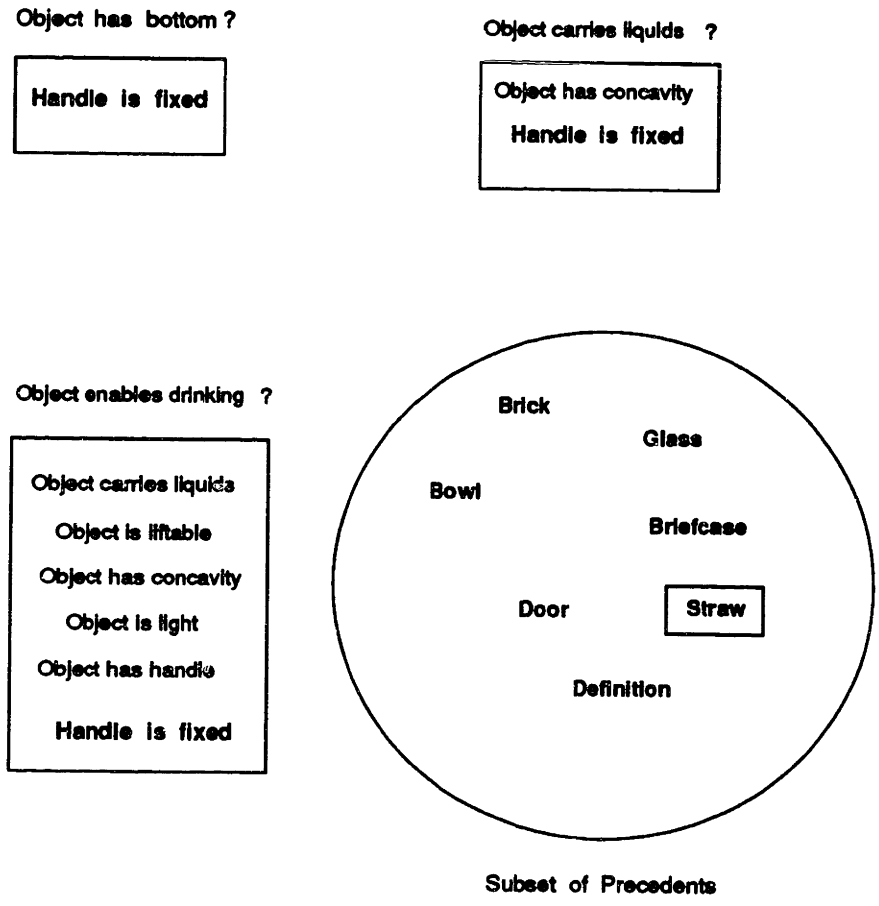

Figure 3.6: A second attempt at reexplanation, with the precedent set augmented by the straw precedent.

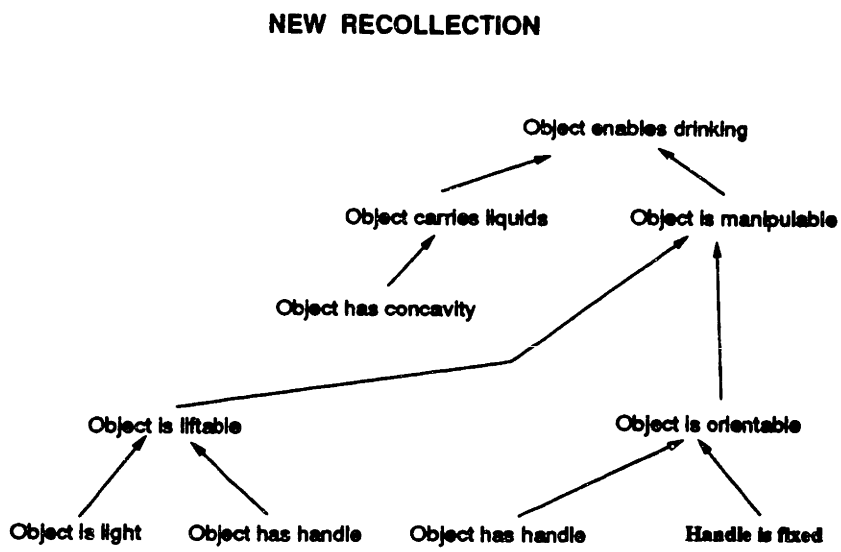

Figure 3.7: The new recollection that explains why the **Handle is fixed** relation is important for the "cupness" of a cup.

in figure 3.8. The true-success suspicious relation **Handle is fixed** has successfully been used to repair the rule; however we can further specialize the rule and learn more in the process by using the false-success suspicious relations also to repair the rule.

3.5 Specialization with false-success suspicious relations

In order to explain why **Handle is hinged** may cause an object not to be a cup, we need to find a recollection from the precedents that can bridge the gap between the **Handle is hinged** relation and the *negation* of any relation in the rule recollection. Such a recollection would explain why the **handle is hinged** relation should prevent the cup rule from firing, and would thus provide the material for a censor to the cup rule.

The task of the repair algorithm is to find a recollection for the negation of a rule relation, so that the recollection includes the false-success suspicious relation **Handle is hinged**. Once again, a priori, the algorithm doesn't know which node of the rule and-tree can be censored by the **Handle is hinged** relation. So the algorithm tries all of them in a breadth first bottom up manner.

The mechanics of the explanation stage of specialization with false-success relations is exactly the same as the explanation stage of repair with true-success

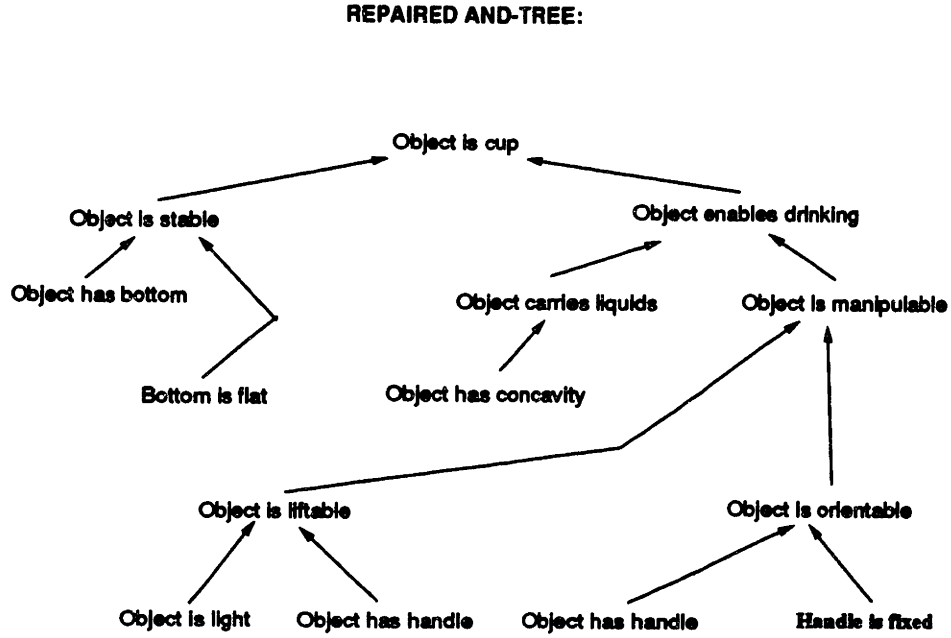

Figure 3.8: The repaired rule recollection.

relations except that we are trying to find a recollection connecting the suspicious relation to the *negation* of the rule recollection nodes instead of the nodes themselves. In this case the initial precedent set is sufficient to find a recollection for `Object is not orientable` in terms of `Handle is hinged`. Figure 3.9 shows the recollection.

Now that we have the necessary recollection, it must be used to repair the rule somehow. Remember that a concept of "cupness" is captured by not just the cup rule but also the host of surrounding censors of rule relations also.

3.6 Repair with the new recollection containing the false-success suspicious relation

Repairing the rule simply involves making a censor out of the recollection obtained as a result of the explanation stage of repair with false-success suspicious relations.

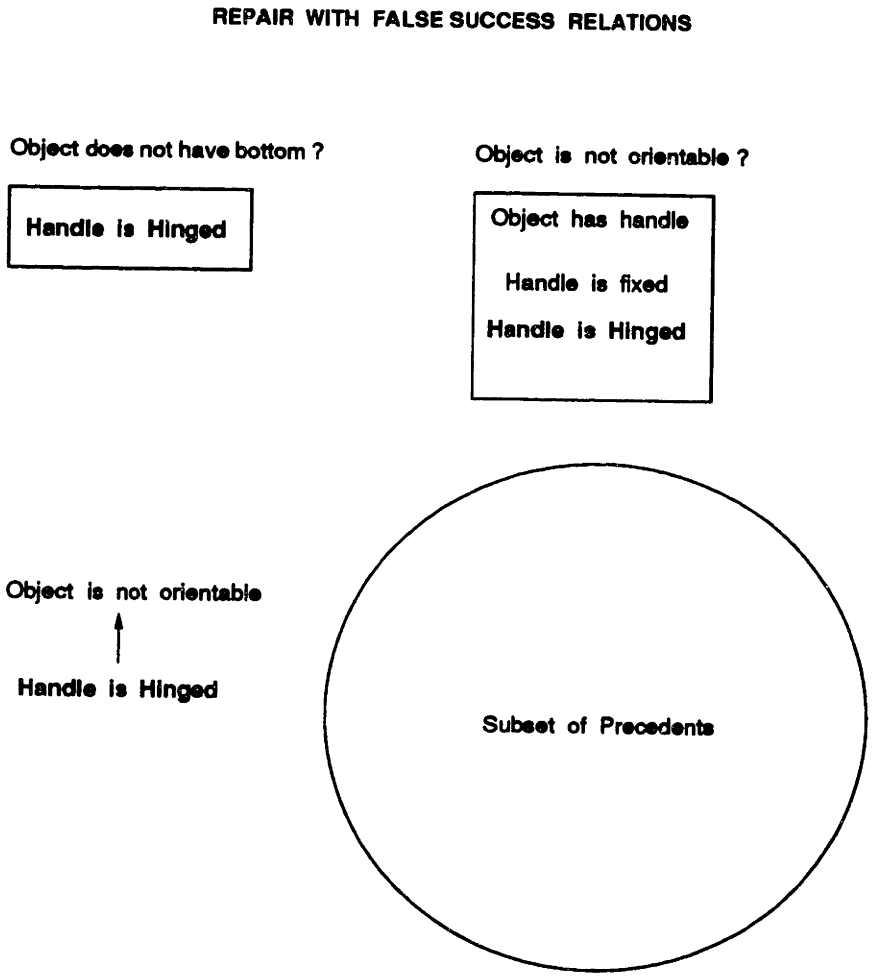

Figure 3.9: The program tries to find an explanation for the negation of a rule-relation such that the explanation includes the false-success suspicious relation **Handle is hinged**.

3.7 The final result of specialization

The specialized rule and the censor are shown below:

The repaired rule:

If:
The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight
The object has a handle
The handle is fixed
Then:
The object is a cup.
Provided:
The object is stable
The object enables drinking
The object carries liquids
The object is liftable
The object is orientable
The object is manipulable

The new Censor:

If:
The handle is hinged
Then:
The object is NOT orientable

To summarize, the key features of the repair mechanism are:

- **Isolation** of suspicious relations via correlation.
- Finding **explanations** that connect the suspicious relations to the rule.
- Using those explanations to **repair** the rule.

It is instructive to compare this result to the quick-fix solution proposed earlier. The repair mechanism of the algorithm has

- Repaired the AND-tree instead of the surface structure.
- Brought knowledge from many precedents to bear on the problem.

- Contributed new-knowledge to the rule, not just speedup. The inclusion of the new relations `Object is orientable` and `Object is manipulable` in the cup rule is not at all obvious from the quick-fix perspective but follows naturally from trying to repair the rule and-tree with knowledge from the precedents.

3.8 There is a panoply of heuristic choices

In implementing the program, a number of choices were decided by heuristic arguments and Ocam's principle. Here are some examples:

- The program is content with the first explanation it finds that connects the suspicious relations to the rule relations (or their negation). It could instead do more work and look for the simplest explanation based on the number of precedents used or the length of the longest explanation chain. However the algorithm excepts the first explanation found on the grounds of simplicity (less work).
- The program could terminate after just one suspicious relation has been incorporated into a recollection for a rule relation (or its negation). However the program tries to include all suspicious relations in recollections to learn as much as possible.
- The program gives up after two rounds of precedent set expansion on the grounds that the more precedents involved, the flimsier the argument.
- The attempt to find an explanation connecting the suspicious relations to the rule and-tree could start from either the root or the leaves of the rule and-tree. The advantage of starting from the leaves is that the repair preserves as much of the original tree as possible. The disadvantage with this approach is that repair of a node high up in the tree can erase the effect of repair lower down in the tree, thus making all that work wasteful. This could be remedied by starting the repair from the root, however the bottom up leaf to root approach is adopted because we are willing to tolerate some wasteful work for a minimal change to the rule.

3.9 The role of the teacher in repair

Once the EBAL system fails on a particular example, it is the job of the teacher to provide it with more examples which are similar to each other in that the system will fail on them for the *same reasons* but are different enough from each other on other properties. The teacher thus plays an important part in the isolation of the suspicious relations.

3.10 The repair may fail due to lack of knowledge

The repair process will fail if no suspicious relations can be isolated, this would indicate that the descriptions of the true and false success examples is not rich enough. The system could then request a richer description. This, however has not been implemnted yet. Another reason why the repair could fail is by not being able to find recollections that connect the suspicious relations to the rule, i.e there is a gap between the suspicious relations and the rule that cannot be bridged by the precedents. This too indicates a paucity of knowledge but in this case the knowledge involves the causal structure between relations.

3.11 Summary

This chapter demonstrated the specialization of the cup rule. Correlation was used to isolate true and false-success suspicious relations. The repair mechanism then tried to explain how the true success relations should be in the rule and how the the false-success suspicious relations should prevent the rule from firing. The search for these explanations is performed in a subset of the entire precedent space. Once found the explanations are used to repair the rule in different ways. The explanation connecting a true-success suspicious relation to a rule relation is simply spliced into the rule and-tree. The explanation connecting a false-success suspicious relation to a negation of a rule-relation is used to make a censor. The advantage of this repair over a quick-fix of the rule is that the deep structure of the rule, namely the and-tree structure is being repaired, and that the repair extensively uses knowledge from the precedents.

Chapter 4

Generalization

4.1 Repair due to specific IF conditions

Continuing with the cup example, the rule after specialization was :

If:
The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight
The object has a handle
The handle is fixed
Then:
The object is a cup.
Provided:
The object is stable
The object enables drinking
The object carries liquids
The object is liftable
The object is orientable
The object is manipulable

However when presented with the objects in figure 4.1 the rule fails to classify them as cups because the IF conditions of the rule require a cup to have a handle and that the handle be fixed. The cup rule is too specific; it needs to be generalized.

4.2 False failures motivate repair

Instances on which a rule should fire but doesn't are known as false failures. The occurrence of a false failure indicates that the rule must be generalized so

Figure 4.1: The cup rule does not consider these as cups because it requires cups to have fixed handles. The rule is too specific.

that the false failures become true successes. There are two possible reasons for a rule not firing on a false failure instance.

1. Some of the IF conditions of the rule do not fire because they are too specific.
2. A censor could have falsified a PROVIDED condition thus preventing the rule from firing.

In the second case the problem is that the censor is too general. It must be specialized. The instances of false failure of the rule become instances of false success of the censor and the specialization mechanism of the previous chapter repairs the censor.

This chapter describes the repair mechanism for the first case, i.e overly specific IF conditions.

4.3 A preview of the result of generalization

The boxed relations are the overly specific IF conditions. Note their absence in the repaired tree. There is a new nameless concept relation and the `Object is manipulable` relation which was previously an internal node of the tree, is now a leaf.

4.4 The general approach

The repair mechanism for generalization follows the same overall pattern as specialization;

- Correlation helps to **isolate** suspicious relations.
- Precedents **explain** how suspicious relations are connected to the rule.
- The explanations **repair** the rule And-tree.

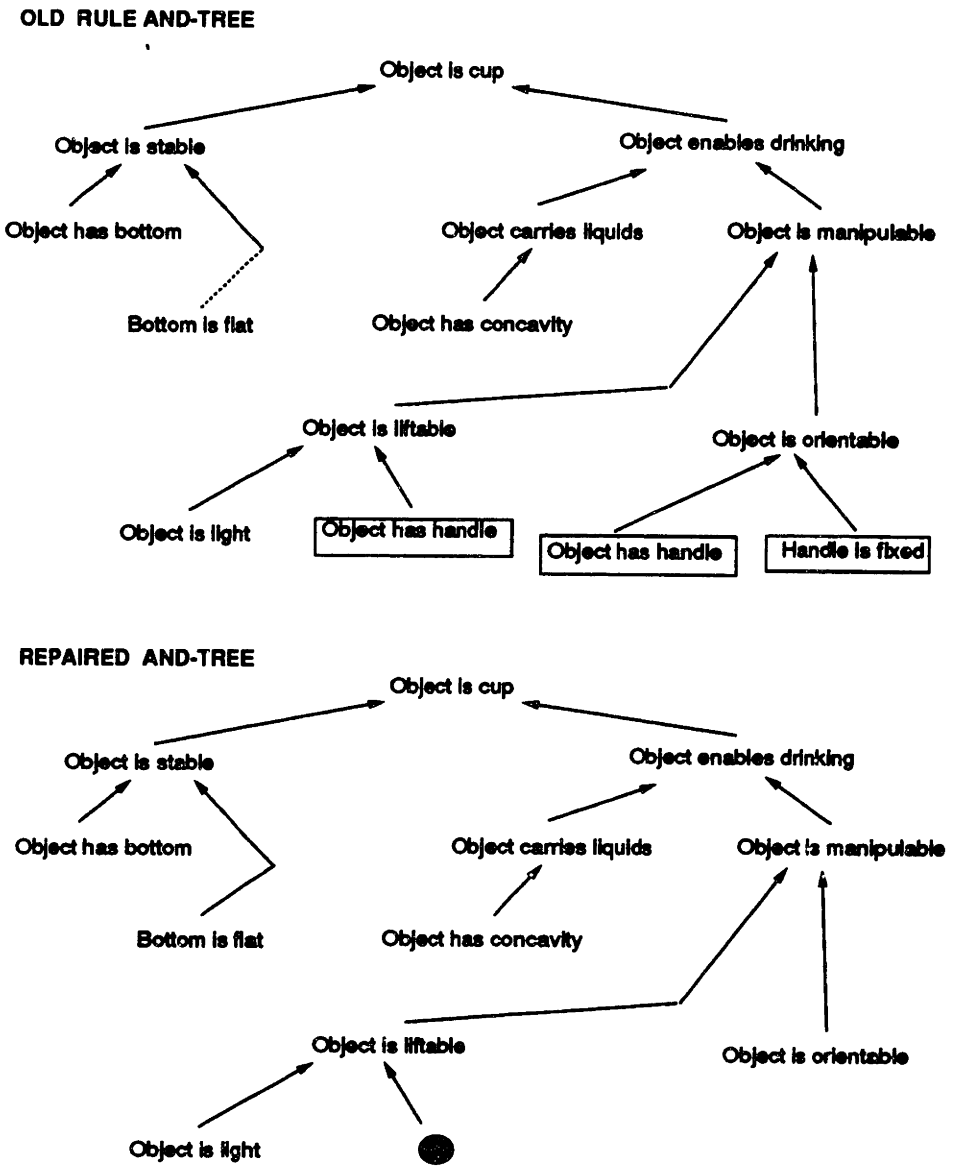

Figure 4.2: The rule recollection before and after generalization.

However the specifics of the repair are different. In the case of specialization the repair mechanism tried to repair all the nodes in the and-tree because there was no way to identify which sub-tree was faulty. In the case of generalization however the repair mechanism does have some information about which nodes of the rule and-tree need repairing. The IF relations that didn't fire and the nodes above them on the path to the root of the and-tree are the only ones that need explaining.

The mechanism tries to find recollections that connect the overly specific IF relations and the nodes above them to the suspicious relations. The new recollection would explain

- how the suspicious relations should make the rule fire.
- why the IF conditions are too specific.

Once we have the explanations, repairing the and-tree cannot be accomplished by simply splicing in the new recollections into the tree, as was done during specialization. The new recollection must be *merged* with the corresponding recollection in the rule and-tree to generalize the rule.

The following sections explore the generalization algorithm in more detail.

4.5 Correlation

Figure 4.3 shows the correlation operation. The rule relations have been intersected with the instances of false-failures. The top shaded region represents relations that are present in the rule but none of the false failures. Any IF conditions in this region represent the source of the problem. The repaired rule must certainly not have them. The relations common to the false failures only are potential sources for alternate explanations for why an object should be a cup.

A Quick-Fix solution here might be to drop all the overly specific IF conditions. Then the new rule would fire on the false-failure instances while still correctly classifying the instances that it used to. This Quick-Fix like the one before is superficial and does not consider the and-tree structure of the rule.

4.6 The specific IF conditions may be explained by the suspicious relations

The repair mechanism first tries to explain the overspecific IF relations directly in terms of the suspicious relations. Figure 4.4 shows the repair in progress. The program tries to explain the specific IF conditions **`Object has handle`** and **`Handle is fixed`** in terms of the suspicious relation **`Object is handsized`**.

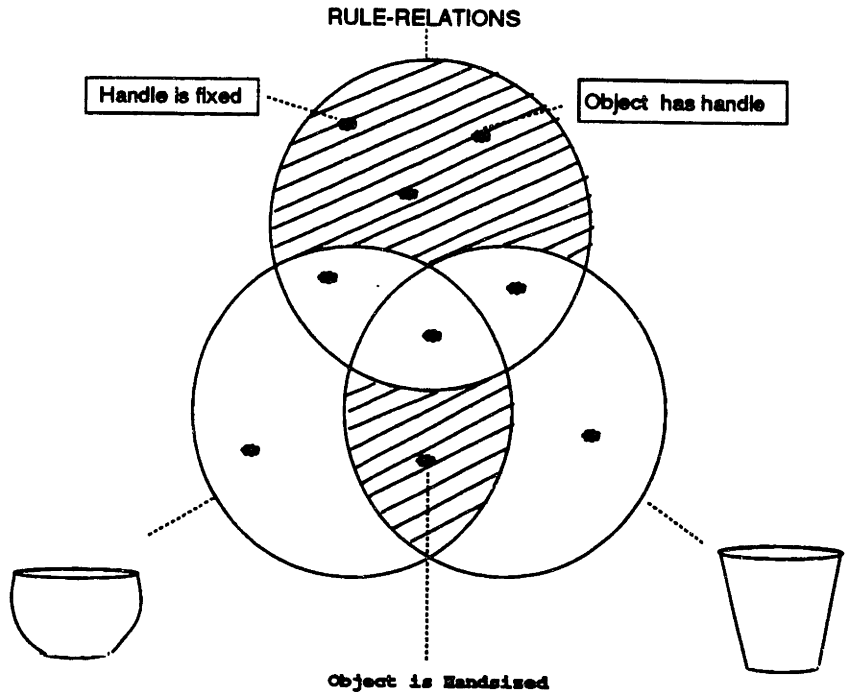

Figure 4.3: Correlation isolates suspicious relations which may help explain why the false-failures should be classified as cups.

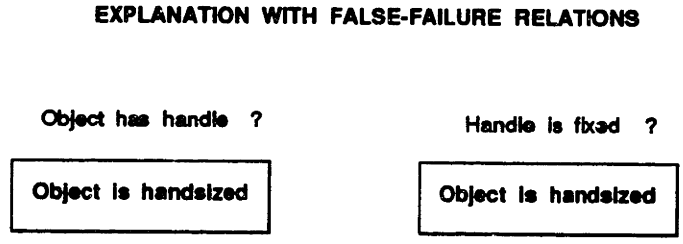

Figure 4.4: The program tries to explain how the relation `Object is handsized` could contribute to the "cupness" of an object.

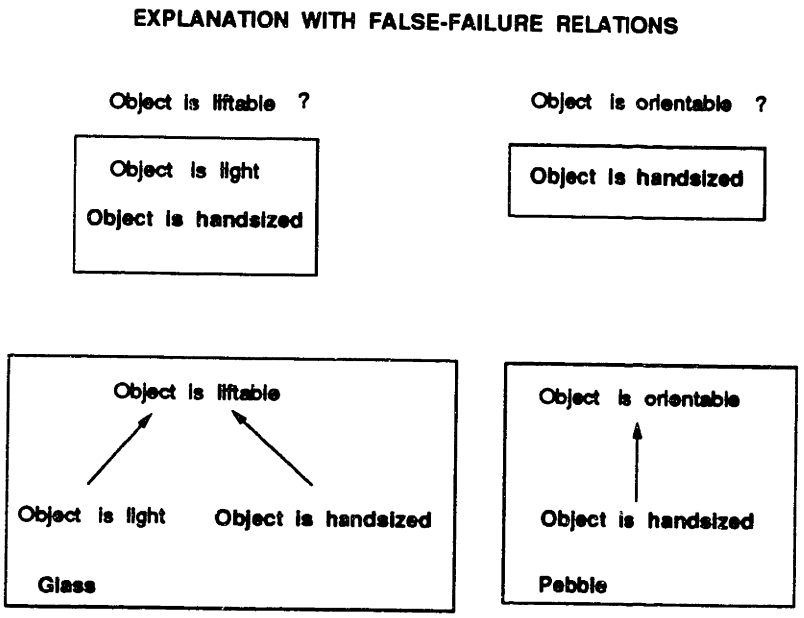

Figure 4.5: The program is able to explain the rule relations `Object is liftable` and `Object is orientable` with recollections that include the the suspicious relation Object is handsized.

When this attempt fails, the program next tries to explain the nodes immediately *above* the specific IF conditions, i.e the `Object is liftable` and `Object is orientable` relations. Figure 4.5 shows this stage of the repair.

4.7 The specific relation may be higher up in the and-tree

Figure 4.5 shows that the algorithm succeeds in finding explanations that contain suspicious relation for `Object is handsized`. These explanations must now be used to repair the rule and-tree.

4.8 The assumption of a common underlying explanation

We now have two different explanations for how a cup can be liftable; the one in the rule and-tree and the other just found by the repair. Figure 4.6 shows

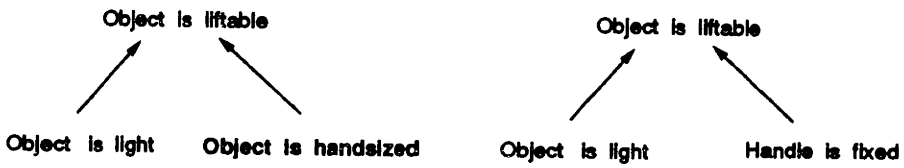

Figure 4.6: Two different explanations for the relation `Object is liftable`. One from the rule and-tree and the other found by the repair mechanism.

the two explanations. Omission of either explanation from the rule makes the rule too specific. Here we make the assumption that *two superficially different explanations for the same relation may have some common underlying explanation.* This guiding principle leads us to first look for the underlying explanation in the precedents. Failing such an explanation, we hypothesize one with the formation of nameless concepts.

The repair process is now explained in greater detail.

4.9 Precedents may help in constructing the underlying explanation

Figure 4.7 shows an attempt to ground `Object is liftable` in two different sets of relations formed from the two different recollections. Any explanation found in this manner would naturally subsume the two recollections, and it would be the common explanation that we are looking for. When this attempt fails, we merge the two recollections by hypothesizing a new nameless concept.

4.10 If the precedents don't contain a common explanation, hypothesize one.

Figure 4.8 shows the merging of the two recollections for the `Object is liftable` relation. The two recollections for `Object is liftable` have a common cause `Object is light`, and one differing relation each. The assumption that the underlying explanation is the same allows us to hypothesize that the two differing relations `Handle is fixed` and `Object is handsized` have the same effect in making an object liftable, given that the object is light weight.

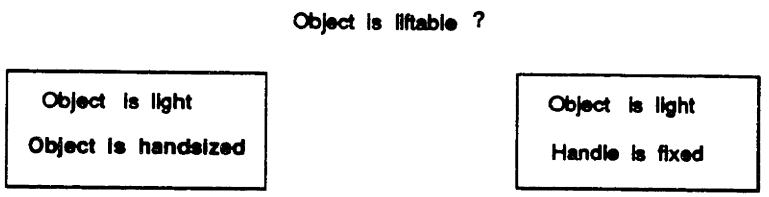

Figure 4.7: The program tries to find a third explanation for the relation `Object is liftable` that can be grounded in both the boxes, i.e an explanation that subsumes the previous two explanations for `Object is liftable`.

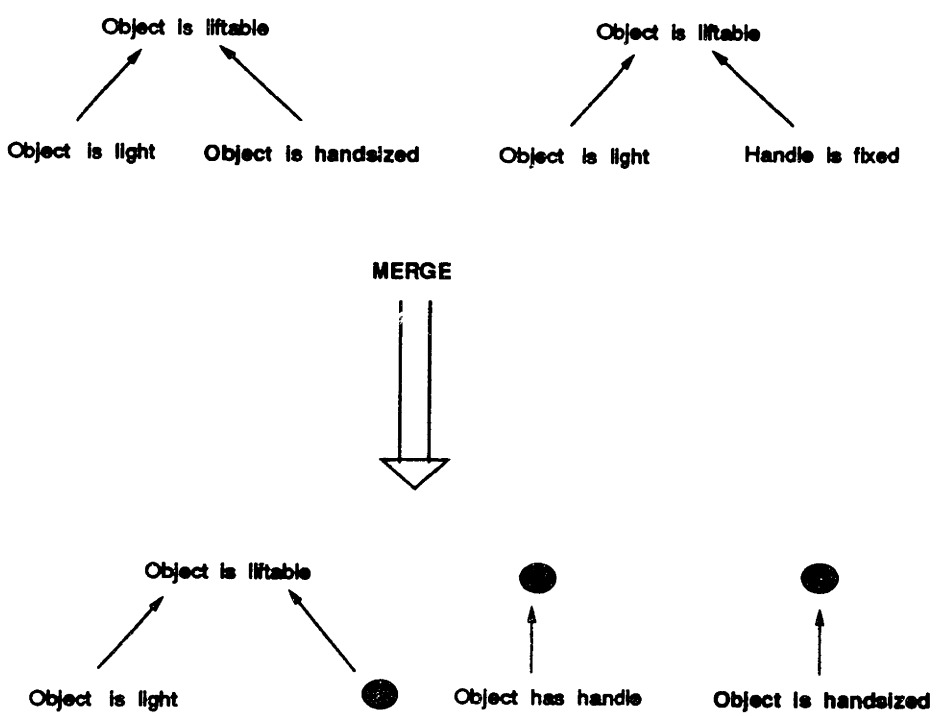

Figure 4.8: The two explanations for `Object is liftable` are merged by hypothesizing a nameless node.

The resulting recollection for `Object is liftable` is spliced into the rule instead of the old one. The repair is not yet over because we have yet to merge the two recollections for `Object is orientable`.

The attempt to merge the two recollections for `Object is orientable` however fails because the recollections do not have any common relations. This forces the repair process to terminate the tree below the `Object is orientable` relation. The final repaired rule and-tree is shown in figure 4.9.

The IF-THEN-PROVIDED form of the repaired rule is:

If:
The object has a bottom
The bottom is flat
The object has a concavity
The object is light weight
nameless-node-497
Then:
The object is a cup.
Provided:
The object is stable
The object enables drinking
The object carries liquids
The object is liftable
The object is orientable
The object is manipulable

The new auxiliary rules are :

If:
The object has a handle
Then:
nameless-node-497

If:
The object is handsized
Then:
nameless-node-497

4.11 Summary

This chapter showed the generalization of the cup-rule. As before correlation was used to isolate suspicious relations. The rule did not fire on the false-failure instances because one or more rule-relations in the tree is too specific.

REPAIRED AND-TREE:

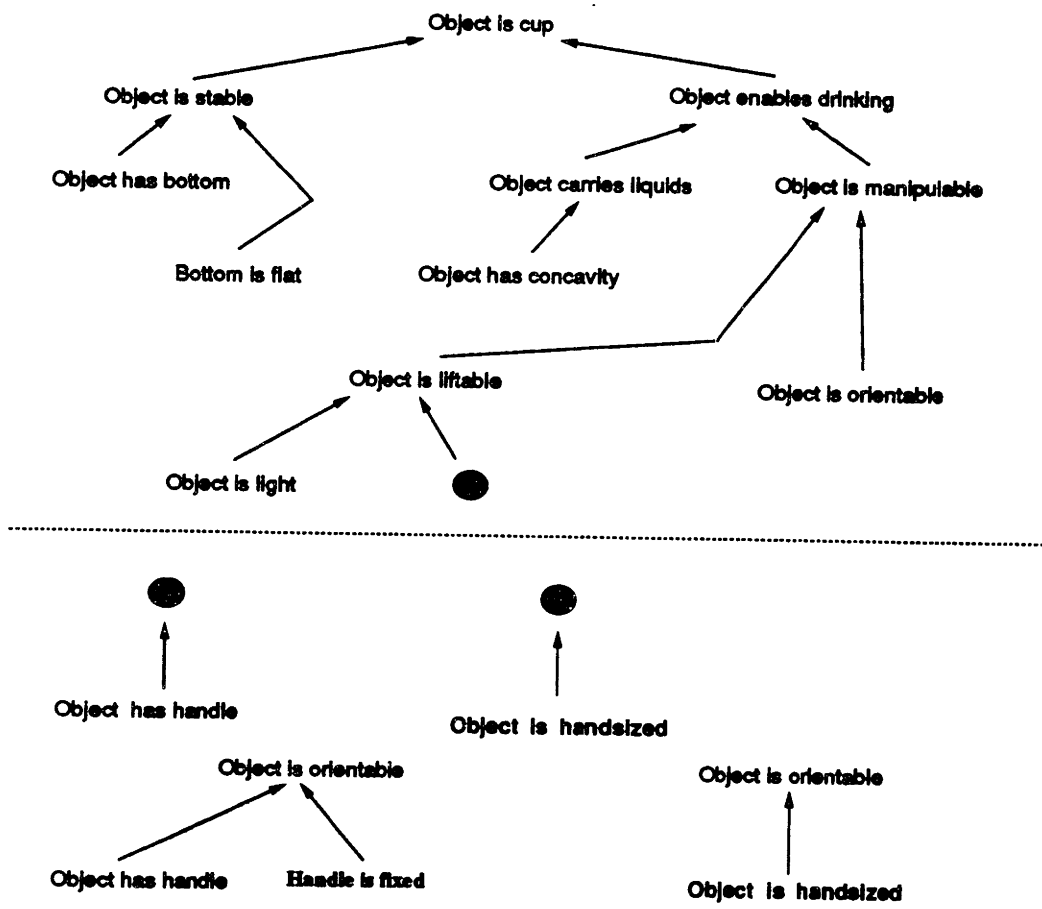

Figure 4.9: The rule recollection after generalization together with some new and-trees that involve a nameless node and some fragmented recollections.

The overly specific IF conditions provide a clue to the identity of these rule-relations. The false-failure suspicious relations provide a alternative explanation for why the rule should fire. This explanation is used to generalize the rule. In the process of generalization a nameless category was hypothesized.

Chapter 5

Nameless concepts

In the previous chapter the repair algorithm used a nameless concept in the final repaired rule. This chapter discusses the formation and function of nameless concepts from a psychological perspective.

Spoken and written communication are such an integral part of human culture that it is easy to blur the distinction between words and the information that they convey; the distinction between a symbol and its meaning. We attach words to many concepts so that we may communicate with another person, however it should be clear that words and concepts are by no means the same thing. In particular there could be concepts such as the smell of the earth just after it has rained, or the concept of either uncle-or-aunt; concepts for which there are no single words in English. We could assign words for these two concepts if we needed to convey them very frequently in everyday life, but then that's not the point. The point is that concepts are fundamental entities and words exist so that we may convey some of those entities, but in fact there are many concepts for which there are no single words.

A natural question to ask is why should our machines be limited to learning only those concepts to which *we humans* attach words? The answer is that they shouldn't be. We now explore the issue of how a machine could acquire a nameless concept.

5.1 The formation of nameless concepts

Concepts that apply to more than one entity in the world are called *categories* or *types*. If a concept can apply to only one particular entity in the world then it is called an *instance* or *token*. So "category" is a more accurate term then "concept" to describe the concept (cups) that we have been discussing so far. The proposal of how human beings could be acquiring certain categories without necessarily the words associated with them has been studied in the

Cognitive Science community. A look at their findings may help shed light on how machines could acquire nameless categories.

5.1.1 Category acquisition

The acquisition of categories is a fundamental and pervasive aspect of human cognition. Categorization is essential because if each new experience were given a unique mental representation, the complexity would quickly overwhelm us and we would not be able to apply previously learned knowledge to new situations. Categorization therefore performs two essential functions:

1. It *reduces the complexity* of the world by allowing us to relate new experiences to old ones.
2. It allows us to draw *reliable inferences from partial information.* Categorizing something as an apple by the way it looks allows us to draw the additional inference that it is edible without actually eating it.

Categorization is vital for memory, reasoning, problem solving and language.

The prototype theory proposes that people abstract a *prototype* or typical member of a category from all the instances they see. The prototype reflects the central tendency or average of the instances of the category, it need not correspond to an actual instance of the category.

Unlike the category models in Machine Learning humans do not need teachers to learn most categories, nor do the categories have to be attached to words. The question then is what are those categories and how do humans learn them?

5.1.2 Basic level categories

One proposal is that there is a *basic level* at which people naturally divide the world into alternative categories [10]. This level maximizes the *perceptual* similarity among instances of the category while maximizing the differences between instances of different categories [1]. For example in the hierarchical sequence Tuna, Fish, and Aquatic creatures. Aquatic creatures is a relatively abstract category, it cannot be defined by a prototype. Fish on the other hand has a clear perceptual representation (shiny scales, fins, body, glassy eye), also instances of fish differ considerably from the related category of crabs. Tuna also has a specific perceptual representation but one that is too similar to the representation of close alternatives like salmon. Fish therefore is a basic level category because its perceptual representation maximizes the similarity between other instances of the same category while also maximizing the difference with members of other categories.

The basic level may differ from person to person based on their individual experiences, a fisherman may have tuna and salmon as basic level categories.

In summary the basic level category hypothesis proposes one way by which nameless categories could be learned; by bottom-up perceptual processes. It does not specify what these processes are but it does provide a computational criterion for distinguishing the resulting category (at the basic level) from other categories.

As demonstrated by the generalization repair mechanism, there is another way in which nameless categories could be formed

5.1.3 Nameless categories formed in relation to other categories

While instances and categories could exist by themselves, they become really useful only when they can be related to other categories. Each relation of the type **Object is liftable** is a named category, the causal links specify the relations between different categories. During the course of generalization a new nameless category was formed. It was defined completely in terms of other categories. It causes an object to be liftable when the object is light. It is caused separately by the relations **Object has handle** and **Object is handsized**.

In summary nameless categories can also be defined operationally in relation to other categories.

5.2 The function of nameless categories

When defined perceptually, nameless categories perform the same function as any other named category. The functions of these categories has already been discussed in section 5.1.1.

The nameless categories created operationally perform two important functions:

- *They tie together important categories.* In the cup example if it was not for the nameless node, the cup recollection would have to be terminated at the **Object is liftable** relation, and the **Object is light** relation *would not appear in the recollection.* Thus the formation of the nameless node prevents fragmentation of the rule recollection.
- *They serve as place holders* until they can be grounded perceptually or named. Assume that after the repair of the cup rule the system sees a new precedent "Pebble" shown in figure 5.1. structure:

  The **Object is graspable** relation can be used to replace the nameless node in the cup rule if necessary because they both have the same causal structure. The nameless node thus serves the function of the **Object is graspable** relation in the cup recollection until it can be replaced.

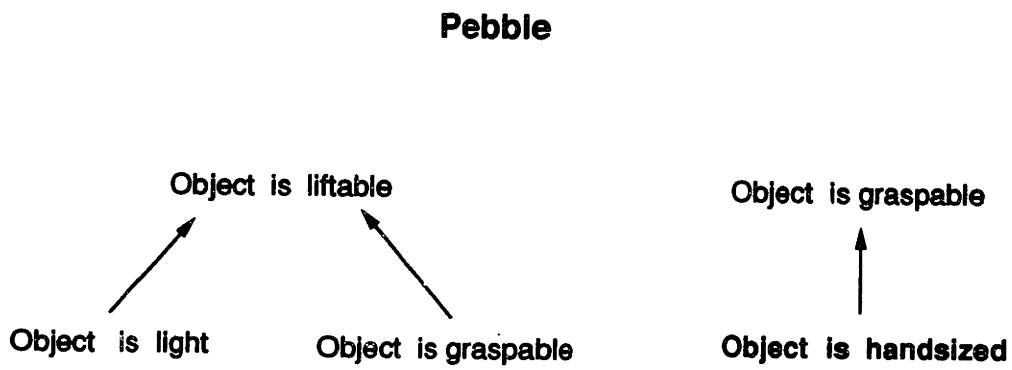

Figure 5.1: The contents of the Pebble precedent

5.3 Nameless categories and the meanings of words

A fascinating connection between nameless basic level categories and language acquisition is to be found in the report [10] that children seem to learn the words for basic level categories before other categories.

To me the tantalizing implication is that the children *first* acquire basic level categories as nameless categories simply because their formation is perceptually driven. These nameless categories then serve as place holders until the child can associate a word with the category. Thus children learn the basic level category names first because it is easy to learn words for *concepts that the child already possesses.*

Putting it another way: the meanings of many words are grounded in perception, so there is no reason why the meaning cannot be acquired before the word is ever heard. The nameless meanings that are first acquired correspond to the basic level categories. The child subsequently attaches words to these meanings that it already possesses. It is no accident therefore that the first words that the child learns correspond to basic level categories.

There is more to be said about the fascinating relationship between language acquisition, perception and nameless categories but that will be the subject of future work.

5.4 Summary

This chapter stresses the distinction between concepts and the words attached to them. Humans are not limited to learning only those concepts for which there are words, so there is no reason why a machine should be. In fact concept formation may very well precede word acquisition for an important class of categories called basic level categories. Nameless categories are formed either by perceptual bottom-up processes or in relation to other categories as demonstrated by the generalization repair mechanism. The function of nameless categories formed by either of these methods was discussed. The nameless concept idea shows the way toward breaking the artificial barrier of having the system learn only what the teacher explicitly tells it to learn.

Chapter 6

Assessment

This chapter describes some problems of the repair process and the EBAL system. In fairness to EBAL it should be mentioned that *all* the other Machine Learning paradigms also suffer from these problems in varying degrees.

6.1 Problems

6.1.1 The role of the teacher

The teacher is responsible for:

- *Providing all the precedents.* Each precedent must contain the relations and the and-tree structures. So far the teacher has had to handcode all the information.
- *Guiding the system towards useful recollections of knowledge.* The system would be perfectly happy to come up with nonsense recollections of the type shown in figure 2.4 if the teacher gave it the appropriate problem. It is the teachers duty to see that he/she gives the system only those problems that lead to useful recollections of knowledge.

To what extent should a teacher be involved in the learning process? The answer I think varies with the application in question. If the system is trying to learn some efficient time-schedule for inserting the control rods in the core of a nuclear reactor - then extensive human aid in the learning process may be pardoned for safety reasons. However if the task is to navigate in some uncontrolled environment and recognize simple objects, then a learning algorithm that doesn't depend extensively on a teacher is to be preferred. If we are to put robot miners on asteroids, dependence on a teacher is out of the question.

6.1.2 Robustness & Scalability

The EBAL systems operates on the *causal structure* of the relations in the precedents, it is quite insensitive to the meanings of the relations themselves. This can be quite misleading to the human dealing with the system who expects the relation `Object is small` to be judged similar to (if not match) the relation `Object is handsized`. The consequence of this is that the system is overly sensitive to the exact symbols used in relations. For this reason the system is not very robust at the moment. However the problem may be overcome by having the system learn that `small` and `handsized` mean the same thing in most situations, this is within the capability of the current system.

The scalability of the system is limited by two factors. One is the effort involved in handcoding a large database, the other is that because of the sensitivity to the exact symbols used, the teacher will have to take care to use the same symbol whenever he/she means the same thing. This can get very cumbersome once we start dealing with thousands of relations.

6.2 What the thesis has shown

The object of the thesis was to see if it was possible to come up with a repair mechanism that in the absence of a sound domain theory does not resort to ad-hoc measures but can use *prior experience* to conduct the repair. The main question is to see if any mechanism *exists* which can do this. The question of efficiency is secondary. Well known graph algorithms can be used to speed up problem solving and repair, however the issue is one of competence, not performance. The last two chapters have described a repair mechanism that demonstrates that it *is* possible to use the structure of precedents in a very natural way to repair faulty knowledge. The notion of nameless categories was introduced and the utility of operationally defined nameless categories was demonstrated in the generalization of the cup rule.

6.3 Future work: grounding Learning in Perception

The problems described at the beginning of the chapter are by no means particular to the EBAL system but are quite pervasive in Machine Learning. I suggest that the root of these problems lies in the lack of an effort to ground learning in perception.

Though it has always been known that learning must somehow ground out in perception, most researchers in Machine Learning have not been very keen

about doing this or at least justifying that their representations could be easily grounded out in perception. For many special purpose applications such as learning the symptoms of human diseases this is fine because perception has no meaning in those cases; the information is provided by a human. However when we deal with cognitive abilities such as understanding natural language, perception certainly enters the picture. *The bottom-up description delivered by the perceptual system constrains what can be learned.* In the previous chapter we saw that entire categories may be constructed in a bottom-up manner, this suggests that the role of the teacher in guiding the EBAL system towards useful recollections of knowledge (useful categories) can in fact be diminished by a bottom up process that constructs basic level categories. The problem of handcoding data disappears because the data is delivered by sensors. Feedback about failure which originally came from the teacher could now come from the environment. How exactly all this is going to be accomplished is the subject future work.

6.4 Summary

The main drawbacks of the repair process stem from problems that pervade much of Machine Learning. The teacher plays an extensive role in guiding the system towards useful knowledge and in keeping track of the learners progress. The system is not easily scalable because the and-tree structures of the precedents have to be put in by hand. The suggestion is that these problems can be remedied by grounding learning in perception - an area for future work. The thesis by Siskind [11] is a valuable step in this direction. In spite of the problems mentioned above this thesis succeeds in showing:

- how knowledge from the precedents can be effectively used for repair of faulty knowledge.
- the formation and utility of nameless concepts.

Bibliography

[1] Arnold Lewis Glass and Keith James Holyoak. *Cognition.* Random House, 1986.

[2] Ryszard S. Michalski James G. Carbonell and Tom M. Mitchell. An overview of machine learning. In Ryszard S. Michalski James G. Carbonell and Tom M. Mitchell, editors, *Machine Learning Volume 1*, chapter 1. Morgan Kaufmann Publishers, Inc., 1983.

[3] W. Labov. The boundaries of words and their meanings. In *New Ways of Analysing Variation in English.* Georgetown University Press, 1973.

[4] David Marr. *Vision*, chapter 7, page 358. W. H. Freeman and Company, 1982.

[5] Marvin Lee Minsky. *The Society of Mind*, chapter 12, page 131. Simon and Schuster, 1986.

[6] Marvin Lee Minsky. *The Society of Mind*, chapter 12, page 127. Simon and Schuster Inc, 1986.

[7] Boris Katz Patrick Henry Winston, Thomas O. Binford and Michael R. Lowry. Learning physical descriptions from functional definitions. In *Proceedings of National Conference on Artificial Intelligence*, 1983.

[8] J. R. Quinlan. Induction of decision trees. In Jude Shavlik and Thomas Dietterich, editors, *Readings in Machine Learning.* Morgan Kaufmann Publishers, Inc., 1990.

[9] Ronald Rivest. Formal models of machine learning. Course notes for the Fall 90 Machine Learning course at M.I.T.

[10] Eleanor Rosch and Loyd. *Cognition and Categorization.* Hillsdale, 1978.

[11] Jeffrey Mark Siskind. *Acquiring Word Meanings from Correlated Visual and Linguistic Input.* PhD thesis, M.I.T, 1991.

[12] Patrick Henry Winston. Learning new principles from precedents and excercises. *Artificial Intelligence Journal*, 19(3), 1982.

[13] Patrick Henry Winston. *Artificial Intelligence.* Addison-Wesley Publishing Company, 1984.

[14] Patrick Henry Winston and Satyajit Rao. Repairing learned knowledge using experience. In Patrick Henry Winston and Sarah Alexandra Shellard,

editors, *Artificial Intelligence at M.I.T Expanding Frontiers*, chapter 14. MIT Press, 1990. Volume 1.