Explanation-Based Methods for Simplifying Intractable Theories

A Thesis Proposal

Thomas Ellman

February 1987

CUCS-265-87

Table of Contents

List of Figures

Explanation-Based Methods for Simplifying Intractable Theories

A Thesis Proposal

Thomas Ellman[1]
Columbia University
Department of Computer Science
New York, N.Y. 10027
(212) 280-8182
Ellman@CS.COLUMBIA.EDU

Abstract

Existing machine learning programs possess only limited abilities to exploit previously acquired background knowledge. A technique called "explanation-based learning" (EBL) has recently been developed to address this problem. EBL is limited, however, by a requirement that the background knowledge meet restrictive conditions. EBL cannot operate without a complete, correct and tractable theory of the domain under study. In many cases no adequate domain theory can be found. The research proposed here will address this limitation. It will be primarily directed toward extending EBL methods to handle intractable theories. Techniques will be developed for using explanations of examples to make domain theories more tractable. The explanations will be used to find assumptions that can simplify intractable theories. A useful class of assumptions, called "optimistic assumptions", will be defined informally. A program will be developed to learn assumptions drawn from this class. The program will be tested in the domain of "hearts" and possibly other domains as well. This research will be significant inasmuch as the "optimistic" assumptions appear to be applicable to a wide variety of domains. The research will also be relevant to the problems of incomplete and incorrect theories as well as the problem of intractability.

1 Introduction

Researchers in the field of machine learning have recently paid considerable attention to the role of "background knowledge". This emphasis is based on the idea that learning should be viewed in a context. When a person encounters a new domain, he brings along a great deal of previously acquired knowledge. As he observes phenomena in the new domain, he may find his prior knowledge to be relevant. He may learn more effectively by making use of his background knowledge in some manner. Although this hypothesis seems plausible, it has yet to be determined precisely how background knowledge can be used for this purpose.

---

[1]This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0165.

In some cases background knowledge exists in the form of an initial "theory" or "model" of the domain. For example, consider the game of chess. A student of chess will usually start by learning the rules of the game. The rules constitute a theory of how to play a winning game of chess. They implicitly describe a search tree containing all the information needed to play optimally. As a student begins to actually play the game of chess he hopefully learns by observing his successes and mistakes. He might very well draw upon the initial theory to facilitate the process of learning from experience.

In order to illustrate the usefulness of the initial chess theory, consider the following hypothetical scenario. Imagine two students trying to learn to play chess by observing examples of good and bad moves provided by a teacher. Suppose the first student is told nothing about the rules of chess. He must learn to play without knowing either how the pieces move or the objective of the game. Suppose further that the second student is given a complete description of the rules. This information almost certainly gives the second student an advantage over the first. Knowledge of the rules should enable the second student to learn more quickly.

A question arises when considering how the rules of chess are actually used to facilitate learning from examples. They certainly cannot be used to perform a search of the entire game tree. Although the rules constitute a complete theory chess, they are intractable. A theory is considered to be intractable if necessary inferences cannot be made without using excessive time and space resources. A similar intractability problem arises in a variety of domains including games like hearts, circuit design, job shop scheduling and numerous others.

When a student learns about an intractable domain, he must use the initial theory in some manner other than exhaustive search. Perhaps he converts the intractable theory into a simplified, approximate model of the domain. Using this approach, he would face a problem if the initial theory can be simplified in more than one way. The training examples might provide some guidance in choosing the right simplification. The student might try to choose simplifications that are consistent with observed examples.

As a result of the foregoing considerations, two general questions have been raised about learning from examples in intractable domains.

- How can an intractable theory be used to facilitate a process of learning from examples?
- How can training examples be used to simplify an intractable theory?

The research proposed here will attempt to answer these two questions.

2 Background

2.1 Overview of Explanation-Based Learning

Researchers in the machine learning field have recently developed a technique called "explanation-based learning" (EBL) [Mitchell et al. 86; Dejong and Mooney 86]. EBL is a method by which previously acquired knowledge can be applied to the problem of learning from examples. In order to operate, EBL systems must be provided with a declarative "theory" or "model" of the domain under study. As an EBL system observes training examples, it uses the theory to facilitate the process of forming concepts. The domain theory enables EBL to form a general concept after observing just a single example. A survey of EBL programs may be found in [Ellman 87].

EBL can be contrasted with a more traditional method of learning from examples called "similarity-based learning" (SBL) [Lebowitz 86]. One difference between EBL and SBL involves the numbers of training examples the two methods require. While EBL techniques can generalize from a single example, SBL methods usually observe multiple examples before creating general concepts. Another difference involves the role of background knowledge. SBL programs require less background knowledge than EBL programs. Furthermore, SBL usually represents the background knowledge in a different manner. In contrast to the declarative domain theory used in EBL programs, SBL systems use background knowledge in the form of an "inductive bias" [Mitchell 80]. The bias is usually encoded in terms of the procedures and representations used by the SBL system. Examples of SBL programs are discussed in [Michalski 83; Michalski et al. 83; Cohen and Feigenbaum 82].

In order to generalize after observing a single example, an EBL program will typically use a two step process. During the first step, the program builds an explanation of the behavior or function of the example. The explanation is intended to capture a general principle embodied in the example. In order to build the explanation, the system uses rules drawn from its theory or model of the domain under study. The second step involves analyzing the explanation in order to construct a generalization of the example. The explanation is used to decide which features of the example can be justifiably generalized. The features are generalized as much as possible as long as the explanation remains valid. The resulting generalization will include other examples that can be understood using the same explanation and that manifest the same general principle.

2.2 Previous Work

In order to illustrate the methods of explanation-based learning, consider the following example from the domain of logic circuit design. The author has implemented an explanation-based learning program that is capable of generalizing from a single example of a logic circuit [Ellman 85]. The program takes as input a pair, (S,D), consisting of a specification of a circuit's behavior and a description of the circuit's design. The learning program creates a generalized schema, (S*,D*), containing a generalized specification and a generalized design. The program was developed and tested using a circular shift register as a training example. The shift register was generalized into a schema describing devices capable of computing arbitrary bit permutations. The program has also been tested on other example circuits including counters, clocks and some combinational circuits.

The program operates according to the two step process described above. During the first step, it builds a proof verifying that the example circuit correctly implements the specifications. In order to build the proof, the program utilizes a data base of rules describing the behavior of circuit components. During the second step, the program analyzes the proof of correctness. For this purpose it uses a procedure similar to goal regression [Nillson 80; Waldinger 77] and constraint back-propagation [Utgoff 86]. This procedure has the effect of generalizing the specifications S and design D as much as possible as long as the original proof remains valid. The resulting generalized schema represents all circuits that can be proven correct using the same proof that was used to verify the example.

2.3 Limitations of EBL

2.3.1 The Inadequate Theory Problem

Explanation-based learning can only operate in the presence of adequate background knowledge. In order for an EBL system to learn from examples, it must possess a knowledge base that can be used to build explanations. Consider the logic circuit domain. To operate in this domain, the EBL system requires considerable knowledge describing the behavior of circuit components. The knowledge base is used to prove the correctness of training example circuits. If this knowledge were not available, the system could not build explanations. Without explanations, the generalization process could not operate.

As a result of the initial knowledge requirements, EBL cannot operate in many domains. Although it may be possible to build adequate initial theories for some highly constrained domains, in most real world situations no adequate initial theory is available. The available domain theories may be deficient in a number of ways. The various types of deficiencies are categorized in [Mitchell et al. 86] and [Ellman 87]. A domain theory is "incomplete" if it fails to

describe parts of the domain under study. An "approximately correct" theory makes erroneous predictions on some examples. An "inconsistent" theory is one that contains contradictory assertions. Even when a domain model is complete, correct and consistent, it may nevertheless be of limited value due to the problem of "intractability".

In order to illustrate the problem of intractability, consider the chess domain. Suppose an EBL system were were built to model the learning process of the hypothetical student described above. The system might be given the chess rules as an initial domain theory along with a series of examples of correct moves. The system would attempt to explain each correct move. The obvious explanation procedure would perform a minimax search of the entire game tree. The minimax search would presumably return a result indicating that the example move is optimal. Unfortunately, such an exhaustive search procedure would be hopelessly intractable. Unless some other method could be used to build an explanation, the remaining steps of the EBL process could not proceed.

2.3.2 Questions about the Value of EBL

EBL methods are limited to performing only a specific type of generalization. In particular, EBL can only create generalizations that are "justified". The generalizations are said to be "justified" because they can be explained in terms of the initial domain theory. There are both advantages and disadvantages to this type of generalization. Assuming the initial domain theory does not contain errors, a justified generalization is guaranteed to be correct. Unfortunately a price must be paid to obtain this guarantee. Since the generalizations are deducible from the initial domain theory, they do not add anything substantially new. To put it more formally, the justified generalizations do not change the deductive closure of the initial theory [Dietterich 86].

A related issue involves the role of examples in explanation-based learning. Considering that EBL produces generalizations that are deducible from the initial domain theory, one wonders if the system could function without training examples provided by a teacher. Consider the circuit design domain. Using the knowledge base of rules describing the behavior of circuit components, the system was able to verify the correctness of a shift register example. It seems likely that the same knowledge base would be sufficient for designing the shifter circuit in the first place. The system could design and generalize its own examples without relying on a human teacher.

Although the foregoing observations might appear to question the value of explanation-based learning, there are other resons why EBL techniques are useful. In particular, EBL is useful for reasons of efficiency. EBL can be viewed as a process of "chunking" whereby sequences of inference rules are combined into schemata called "chunks". After the chunks are

added to the initial knowledge base, some inferences can be made in fewer steps. The resulting reformulated theory is potentially more efficient than the original one. This viewpoint suggests why training examples are valuable. For any given set of rules, many different combinations can be combined into chunks. The training examples indicate which combinations are most useful [Ellman 87; Mitchell et al. 86].

2.4 Future EBL Research Tasks

A number of research tasks are faced by investigators working in the EBL field. These tasks can be naturally divided into two groups. The two groups may be characterized by the position they take regarding the fact that EBL cannot change the substantive content of a knowledge base, as measured by its deductive closure. One group would accept this limitation. EBL would be viewed as a method of "theory reformulation", which can make a knowledge base more efficient but cannot alter its content. The second group would try to extend EBL to be a method of "theory revision", which can change actual content of the initial domain theory.

Research in the "theory reformulation" paradigm would be mainly concerned with optimizing the efficiency of the reformulated theory produced by EBL methods. They might address the following issues: (1) Demonstrating that EBL really produces more efficient domain theories; (2) Determining when schema formation is warranted; (3) Deciding which schemata should be retained and how they should be organized; (4) How best to represent explanations; (5) How best to analyze explanations; (6) Making use of contextual knowledge; (7) Interacting effectively with human teachers. These issues are discussed in greater depth in [Ellman 87].

Research in the "theory revision" paradigm has the potential to address questions raised in the previous section about the value of EBL. This research would try to extend EBL methods so they can actually change the deductive closure of an initial domain theory. Such new methods might be developed by addressing the problem of deficient domain theories described above. The problems of incompleteness, inconsistency and approximate correctness can only be remedied by methods that change the deductive closure of a theory. Some investigators have taken the view that EBL should be combined with similarity-based learning methods in order to handle the deficient theory problems [Lebowitz 86; Pazzani et al. 86]. Since SBL involves using multiple examples, this approach also holds out the hope of clarifying the role of training examples.

The intractable theory problem also fits into the "theory revision" paradigm. This might appear surprising at first. If an intractable theory is both complete and correct, one normally wants only to make it more efficient without changing the actual content. In practice, however, the efficiency improvement may come at a price. Increased efficiency may be possible only if

approximations are introduced into the theory. Approximations would have the effect of changing the deductive closure of the original theory. As the initial theory is revised by approximations, it would, ironically, become incorrect.

2.5 Issues to be Addressed by Proposed Research

The research proposed here will address many of the outstanding issues described above. It will focus primarily on the problem of intractable domain theories. The techniques to be developed may also be relevant to the problems of incomplete and approximate theories. This research will also address the criticism that EBL does not change the deductive closure of the initial knowledge base. It will use explanation-based methods to simplify and approximate intractable theories. The approximations will have the effect of changing the deductive closure of the initial intractable theory. Finally, the proposed research will help to clarify the role of examples in explanation-based learning. The techniques to be developed will make use of multiple training examples. The results of learning will be seen to depend crucially on the examples actually observed. In this respect the proposed research is similar in spirit to other attempts to combine EBL and SBL learning techniques [Lebowitz 86; Pazzani et al. 86].

3 Proposed Research

3.1 The Problem

The intractable theory problem can be defined in more than one way. Two alternate problem definitions are shown in Figure 1. The first definition describes the problem in terms of concept formation. The goal is to find a concept that is consistent with a set of observed training examples. This problem is under-determined as stated, since mere consistency with observed examples is not usually a sufficient criterion for finding a unique concept description [Mitchell 80]. The intractable theory may therefore be used to guide the choice of an appropriate concept. The second definition describes the problem in terms of simplifying the intractable theory. Since a theory can be simplified in more than one way, this problem is also under-determined. The training examples may therefore be used to choose an appropriate simplified version of the initial theory. These two problem definitions are essentially equivalent. A simplified theory may be viewed as a concept description and visa versa.[2]

---

[2]The problem definition might be modified to include an additional input. As described below, the learning program will use some additional heuristic information to decide how to simplify the initial theory. These heuristics may or may not be considered to be formal inputs depending on whether they can be expressed in a domain independent manner.

```
Given:  a. An intractable theory of some domain.
        b. A set of training examples drawn from the same domain.

Definition #1:
Find:   A concept description that is consistent with the training
        examples. (Using the intractable theory as a guide.)

Definition #2:
Find:   A simplified version of the intractable theory. (Using the
        training examples as a guide.)
```

Figure 1: Problem Definition

3.2 Hypotheses

The proposed research is based on a several hypotheses. These claims are highlighted in Figure 2. The first hypothesis states that intractable theories can indeed function as a useful source of background knowledge for systems that generalize from examples. The second hypothesis asserts that examples are useful for the purpose of simplifying intractable theories. These first two hypotheses merely assert that the intractable theory problem can be solved as it is stated in Figure 1.

1. **Intractable theories can guide a choice between alternate ways of generalizing training examples.**

2. **Training examples can guide a choice between alternate ways of simplifying intractable theories.**

3. **Intractable theories can be made tractable by a process of adding simplifying assumptions.**

4. **Explanations of examples can help to find useful simplifying assumptions.**

Figure 2: Major Hypotheses

The third hypothesis defines a type of operation that might be used to simplify intractable theories. It asserts that theories can be made tractable by a process of adding "simplifying assumptions". An assumption A may be said to "simplify" a theory T if the theory T becomes more tractable when assumption A is included as an additional axiom. The significance of this hypothesis may be seen by considering alternatives. In principle, an intractable theory might be simplified by procedure that is more complex than the mere addition of assumptions. A more complex method might require both addition and retraction of axioms or changes of representation, for example. This hypothesis asserts that incremental addition of assumptions is a useful method of simplifying a theory, even if other methods are possible in principle.

The final hypothesis involves the role of explanations in learning from examples. It asserts that explanations are useful for the purpose of finding assumptions that make an intractable theory easier to handle. In the course of trying to explain an example, a system can discover assumptions that enable the explanation to go through. Simplifying assumptions are adopted by a system because they facilitate explaining examples. In effect, this process is similar to abductive reasoning [Pople 73]. This hypothesis envisions a role for explanations that is different from the role envisioned previously by EBL researchers. The traditional view assumes that explanations are useful for finding constraints that must be maintained while generalizing a single example. If this hypothesis is correct, previous EBL research may have failed to recognize a major reason why explanations are valuable for the purpose of learning from examples.

3.3 An Example from Hearts

In order to illustrate the notion of "simplifying assumptions", consider the following example from the card game hearts.[3] A student is learning to play hearts by observing the behavior of a teacher who is actually playing the game. The teacher is faced with the situation shown in Figure 3. The leader of the current trick has just played the eight of hearts. According to the rules, the teacher is required to play one of his hearts. Therefore, he can choose either the queen, the seven or the three. It turns out that the teacher chooses to play the seven of hearts.

The student might try to explain this example with the line of reasoning shown in Figure 4. The explanation is based on a strong assumption about the structure of the game. The assumption asserts that the outcome of future tricks is independent of the teacher's card choice in the current trick. By adopting this assumption, the teacher can ignore the future and focus on minimizing his score for the current trick. The assumption denies the possibility that the teacher might avoid many points in the future by accepting a few points in the current trick.

It is important to notice how the assumption dramatically improves the tractability of the

---

[3] Hearts is normally played with four players. Each player is dealt thirteen cards. At the start of the game, one player is designated to be the "leader". The game is divided into thirteen successive tricks. At the start of each trick, the leader plays a card. Then the other players play cards in order going clockwise around the circle. Each player must play a card matching the suit of the card played by the leader, if he has such a card in his hand. Otherwise, he may play any card. The player who plays the highest card in the same suit as the leader's card will take the trick and become the leader for the next trick. Each player receives one point for every card in the suit of hearts contained in a trick that he takes. In the simplest version of the game, the objective is to minimize the number of points in one's score. Other versions are more complicated. Complete rules are found in [Gibson 74].

```
Trick Number:    3

Current Scores: TEACHER: 0
                TOM:     0
                DICK:    2
                HARRY:   0

Lead Suit:      ♥

Cards on Table: ♥ 8, 6
                ♣ KING

Teacher's Hand: ♠ JACK, 7
                ♥ QUEEN, 7, 3
                ♦ ACE, 10, 4, 2
                ♣ JACK, 9

Teacher's Card Choice:  ♥ 7
```

Figure 3: Example #1 from Hearts

```
Assumption: The number of points the teacher will take on future
            tricks will be the same, regardless of his choice of
            card in the current trick.

0. Let C be the number of points the teacher takes in the current
   trick. Let F be the number of points the teacher takes in all
   future tricks.
1. Since hearts is the lead suit, the highest heart will win the
   trick.
2. If the teacher plays ♥ QUEEN, then he will win the trick,
   since he is the last player for this trick, and since the
   ♥ QUEEN is greater than the other hearts on the table.
3. If the teacher wins the trick, then C = 3, since he will pick
   up three hearts in the trick.
4. If the teacher plays ♥ 7, then he will lose the trick,
   since the ♥ 7 is lower than the ♥ 8 on the table.
5  If the teacher looses the trick, then C = 0, since he will not
   pick up any cards.
6. The teacher's score for the game will be C + F.
7. By assumption, F will be the same whether the teacher plays the
   ♥ QUEEN or the ♥ 7.
8. Therefore the teacher's total score will be less if he plays
   the ♥ 7.
9. Since the objective of the game is to minimize one's total score,
   the teacher should play the ♥ 7.
```

Figure 4: Explanation of Example #1 from Hearts

hearts theory. In the absence of the assumption, the student would be forced to perform a look-ahead search through all the possible future situations. After adopting the assumption, the student can restrict his attention to explaining what happens on the current trick. The student needs only to explain how the teacher's card choice avoids taking points in the current trick.

Thus the assumption truly simplifies the intractable theory. It enables the student to understand an example that he could not explain using the rules of the game alone.

Although the assumption is important, it is not a substitute for the intractable theory of the hearts game. The explanation is based on the rules of the game as well as the simplifying assumption. For example, the explanation uses rules specifying (1) how to decide which card takes a trick, (2) which cards are worth points and (3) the objective of the game, among others. Using the simplifying assumption alone, without these additional facts from the rules of the game, the explanation would not go through. The simplifying assumption by itself is not sufficient for explaining the card choice.

3.4 Criteria for Choosing Assumptions

In order to find useful assumptions, it helps to have some criteria for evaluating possible alternatives. Two general criteria for evaluating assumptions are shown in Figure 5. The "explanatory power criterion" evaluates assumptions in terms of their ability to help explain the observed training examples. The "simplifying power criterion" evaluates assumptions in terms of their effect on the tractability of the initial domain theory.

1. **Explanatory Power Criterion: Evaluate assumptions in terms of the degree to which they enable the system to explain the observed training examples.**

2. **Simplifying Power Criterion: Evaluate assumptions in terms of the degree to which they simplify the initial intractable theory.**

Figure 5: Criteria for Choosing Assumptions

3.4.1 The Explanatory Power Criterion

As an illustration of the "explanatory power" criterion, consider the assumption used by the student of hearts in the explanation described above. The student might not adopt the assumption immediately after using it in a single explanation. After all, the example might be explained in other ways. Suppose the student were to build explanations of numerous examples, all of which involved this very same simplifying assumption. In that case, he might be inclined to adopt the assumption as being at least approximately correct. With each increase in the number of examples explained using this assumption, the student's confidence in the assumption would probably grow.

Many issues must be addressed in order to measure the "explanatory power" of an assumption. For any given assumption, a number of different measures might be maintained. These include the numbers of examples predicted and contradicted by explanations involving

the assumption, as well as the number of examples that are independent of all explanations involving the assumption. An assumption A may be said to "predict" an example E if the example E can be deduced using assumption A in combination with the initial intractable theory. Likewise, an assumption A may be said to contradict an example E if it makes a prediction that is inconsistent with the example E. These three measures are alternate dimensions along which assumptions can be compared, along with numerous combined measures.

The explanatory power criterion alone is not sufficient to constrain the search for useful assumptions. To illustrate this fact, suppose a system observes examples e1,e2,...,eN. These can all be "explained" by using as an "assumption" the conjunction e1 $\wedge$ e2 $\wedge$ ... $\wedge$ eN. For any set of examples, there will always be an infinite set of assumptions that are equally capable of explaining the examples. Some additional criteria must be used to find "good" assumptions.

3.4.2 The Simplifying Power Criterion

The notion of "simplifying power" can be informally defined in the following way. Suppose a system starts with an initial intractable theory T and is considering adding either assumption A1 or assumption A2 to the axioms of theory T. If adopting A1 would increase the tractability of T more than adopting A2, then A1 may be said to have greater "simplifying power" than A2. This definition differs from other simplicity criteria such as "Occam's Razor". Instead of measuring the simplicity of the assumption itself, the "simplifying power criterion" measures the impact of the assumption on an existing intractable theory.

As an illustration of the "simplifying power" criterion, consider the following two informally stated assumptions about the hearts game:

- Assumption A1: Except for the number of cards played in the suit of hearts, the outcomes of the current and future tricks are independent of all information about prior tricks.
- Assumption A2: Except for the numbers of cards played in each suit, the outcomes of the current and future tricks are independent of all information about prior tricks.

Each of these assumptions asserts that the system can safely ignore information from prior tricks of the game. They enable a system to abstract information away from the description of the current situation. The first assumption ignores more information than the second and leads to reasoning in a more abstract problem space. Adopting the first assumption would probably lead to a more tractable reasoning process. For this reason the first assumption is considered to have greater "simplifying power" than the second.

In order to implement the "simplifying power" criterion, several problems must be overcome. A system using this criterion requires the ability to determine whether or not an assumption actually simplifies an intractable theory. In order to choose between two or more

simplifying assumptions, the system also needs a method of comparing the amounts by which they each simplify the theory. In principle, such tests might be performed by analyzing the time and/or space complexity of a domain model before and after an assumption is adopted. In practice, however, the task of automating the complexity analysis would probably be difficult.

An alternative approach holds out the possibility of avoiding a difficult analysis of the complexity of domain models. The alternative method would restrict the learning system's attention to a few general types of simplifying assumptions. A partial typology of such assumptions is shown in Figure 6. This typology is useful for several reasons. To begin with, each type of assumption is known to be capable of simplifying an intractable theory. For example, a system that adopts an "abstracting" assumption can be sure the assumption will simplify the domain model, without actually performing a complexity analysis. This typology is also useful for the purpose of comparing the "simplifying power" of alternative assumptions. When assumptions are restricted to certain types, such comparisons can be made more easily. For example, consider the problem of comparing two "abstracting" assumptions A1 and A2. Suppose assumption A1 recommends ignoring a superset of the information ignored by assumption A2. Adopting A1 would therefore lead to a simpler theory than would result from adopting A2. As another example, consider two "decomposing" assumptions A3 and A4. Suppose that A3 recommends ignoring a superset of the subgoal interactions ignored by assumption A4. Adopting A3 would therefore lead to a simpler theory than would result from adopting A4. In each of these cases, the comparison is facilitated by the restricted form of the assumptions.

1. **Decomposability:**
   a. **Assume a joint optimization problem can be solved by individually optimizing subproblems.**
   b. **Assume that solutions to a conjunction of subgoals can be found by composing solutions to the subgoals in any order.**
2. **Independence of Variables:**
   a. **Assume that the probability of A given B is equal to the probability of A.**
   b. **For some function F, assume that F(a) = F(b) for all values of a and b.**
3. **Greediness: Assume a problem of optimizing long run outcomes can be solved by making decisions to optimize the immediate effects.**
4. **Abstraction: Assume adequate solutions can be found by working in an abstraction space created by formation of abstract states or abstract operators.**

Figure 6: Typology of Optimistic Assumptions

The assumption types shown in Figure 6 all share a common theme. Each assumption

pretends that the domain under study is really quite simple. For this reason they shall be collectively known as "optimistic" assumptions. The typology in Figure 6 was obtained by a process of examining protocols of hearts games played by humans. The protocols contained verbal explanations of peoples' decisions about which cards to play. The protocols were analyzed for the purpose of finding simplifying assumptions contained implicitly in the explanations. A list of "optimistic" assumptions from the hearts domain is found in Appendix I.

Some of these types of assumptions may be equivalent to each other. For example, consider the assumption used in Figure 4, which asserts that future tricks are independent of the card choice on the current trick. This statement may be viewed as an assumption about the "decomposability" of the thirteen tricks of the game. It may also be viewed as a "greedy assumption" since it leads to optimizing the short term effects of card choices. As another example, consider the two assumptions mentioned above that recommend ignoring information about previous tricks of the game. These may be viewed as "abstracting assumptions" since they suggest using abstractions of the current problem state. They may also be viewed as assumptions about independence of variables.

In order to illustrate the relevance of the assumption typology in Figure 6, consider another example from hearts. Figure 7 shows a situation in which the teacher is the leader. He chooses to play the ♦ 6. This card happens to be the lowest rank card in his hand. It also happens to come from the suit of diamonds, a suit with the most cards "out".[4] The teacher's choice can be explained in the following way. The explanation involves two assumptions. One of these is the assumption from the hearts example described above, (Figure 4). Under this assumption the optimal card choice will minimize the odds of taking points in the current trick. The second assumption is one of the "abstracting assumptions" described above. It recommends ignoring all the information about preceding tricks except for the numbers of cards played in each suit. With only this information available, one can argue that the teacher's opponents are less likely to be "void"[5]in diamonds than in any other suit, since the number of diamonds "out" is greater than that of any other suit. Leading a diamond will therefore minimize the odds that a heart will be played. Since the ♦ 6 is the lowest card in the teacher's hand, this choice will also minimize the odds that the teacher takes the trick. Therefore the choice of ♦ 6 will minimize the odds that the teacher takes points in the current trick.

---

[4]A player considers a card to be "out" if it has not been played and is not in his own hand.

[5]A player is said to be "void" in some suit if he does not have any cards of that suit in his current hand

```
Trick Number:    6

Current Scores: TEACHER: 0
                TOM:     0
                DICK:    2
                HARRY:   0

Lead Suit:      Unknown

Cards on Table: None

Teacher's Hand: ♠ JACK, 7
                ♥ QUEEN, 7
                ♦ ACE, JACK, 10, 6
                ♣ JACK, 9

Cards Out:      Spades Out:   6
                Hearts Out:   6
                Diamonds Out: 9
                Clubs Out:    1

Trick History:  ...Record of Each Trick...

Teacher's Card Choice:  ♦ 6
```

Figure 7: Example #2 from Hearts

The typology of "optimistic" assumptions in Figure 6 may provide a means for implementing the "simplifying power" criterion for choosing assumptions. For this purpose the typology must first be extended into a more complete list of useful types of simplifying assumptions. After a more complete list is obtained, each type of "optimistic" assumption might be encoded in terms of a schema or a collection of schemata. An assumption would then be considered to be "optimistic" if it matches at least one of these "assumption schemata". The learning system would then be restricted to adopting only assumptions that match one of these "assumption schemata".

A major open issue involves the degree to which the "simplifying power" criterion can be implemented in a domain independent manner. For purposes of generality, one would like to encode the "optimistic" assumption typology of Figure 6 into a domain independent set of "assumption schemata". Once the schemata are encoded in a domain independent form, some process must instantiate them in the context of the domain theory the system is currently using. Since the schemata would refer to terms like "subproblem", "variable" and "joint optimization", the system must be able to determine the parts of the domain theory to which these terms refer. For this purpose the learning system would require some knowledge of the formal structure of the domain theory. In practice, such a level of domain independence might be difficult to

achieve. It might be difficult to represent the assumption typology in a domain independent form. The automatic instantiation process might also be difficult. If these problems were to arise, a simpler approach might be used. The "optimistic" assumption typology would be encoded in a domain specific form. The domain specific "assumption schemata" would presumably be much simpler to instantiate.

3.4.3 Combining the Criteria

In the course of combining the "explanatory power" and "simplifying power" criteria, additional issues must be addressed. These two criteria might conflict with each other. The most tractable theory might not be the one with the greatest power to explain the observed examples. In this case the system must evaluate a tradeoff between tractability and correctness. It may also turn out that these two criteria together are not sufficient to constrain the search for assumptions. In that case some additional criteria must be found.

3.5 Additional Key Issues

3.5.1 Representing Explanations

A major open question involves finding an appropriate representation for explanations of training examples. For this purpose two related problems must be addressed. As described above, explanations will use two types of information, including simplifying assumptions and facts drawn from the initial intractable theory. In the course of representing explanations, it will be necessary to find representations for both the theory and the assumptions. These two problems are interdependent. The theory must be represented in such a way that it can make use of the assumptions. The assumptions must likewise be represented in a manner that leads to a tractable domain model when they are added to the axioms of the initial theory.

A difficulty may arise when adopting simplifying assumptions that are not strictly true. Consider the assumption described in Figure 4, which asserts the independence of current and future tricks. This assumption is not strictly true in most if not all hearts game scenarios. When this assumption is incrementally added to a data base containing the rules of hearts, the result is an inconsistent theory. The inconsistency can be a problem if proof by contradiction or another equivalent inference method is used. If this becomes a practical problem, there are two possible remedies. One approach would attempt to retract the parts of the initial theory that are inconsistent with the added assumption. If this fails it might be necessary to investigate methods of theory simplification that are more complex than incremental addition of assumptions.

3.5.2 Validating the Role of Examples

In the course of building a learning system, investigators must be aware of the danger of rigging the outcome. Sometimes the "results" of a learning process are implicitly encoded in the knowledge representation scheme or the learning procedures. In the context of this research, there exists a danger that the "optimistic" assumption typology, shown in Figure 6, will completely determine the assumptions to be learned. To avoid this pitfall, the proposed research will attempt to demonstrate that the training examples actually do make a difference. In order to demonstrate the indispensibility of training examples, the learning system must be capable of acquiring different assumptions depending on the examples that it observes.

3.6 The Hearts Domain

3.6.1 Why Choose the Hearts Domain?

This research will be pursued primarily in the domain of the card game "hearts". The hearts domain has been chosen for several reasons. To begin with, it manifests several of the inadequate theory problems described above. Depending on how one defines the objective of the game, the hearts rules may be seen as an instance of several types of deficient domain theories. Using some definitions of the hearts game objective, the rules constitute a complete but intractable theory of the game. Using another definition, the rules constitute an incomplete intractable theory. Yet another definition leads to an approximate and intractable model. For these reasons hearts is suitable for studying most if not all of the inadequate theory problems.

In order to obtain a complete but intractable theory of the hearts game, the objective is defined as optimizing the "worst case" outcome of the game. This "worst case" analysis would envision both the worst possible deal and that the opponents will choose cards leading to the worst score for the player analyzing the game. In this scenario one could determine which card to play by performing a mini-max search of the entire game tree. The search would perform a minimizing operation over all of the opponents' possible moves and over all the hands they might possibly hold.[6] The entire game tree is specified by the rules of the game. Therefore the rules constitute a complete model of the game. Since the game tree is large, the model is intractable in practice.

In order to obtain an incomplete intractable game theory, one removes the assumption that the opponents will play the worst card for the player analyzing the game. The remaining

---

[6]A different complete but intractable model would not assume the "worst" possible deal. This alternate theory would still perform a minimizing operation over all the opponents' moves, but would average over all the hands they might hold.

game theory says nothing about how the opponents will play. The theory would therefore be incomplete and intractable. If one introduces some rules about how people behave, (e.g., that they avoid risk, or that they have limited memory), one would have an approximate theory of hearts.

The hearts domain also provides another type of flexibility. The level of complexity of the game can be varied depending on the implementation difficulties encountered. There are several different versions of the game, each with rules of varying degrees of complexity [Gibson 74]. In addition to the "official" versions, there are several ways in which the game can be simplified. It can be modified by reducing the number of cards and/or players. Another modification would change the scoring rules so that a person's score is equal to the number of tricks he takes.

3.6.2 Representation Issues in Hearts

Many representation issues must be addressed in the course of studying the hearts domain. One such issue involves the training examples. There are several possible ways of representing the examples. An example might consist of a single situation/action pair describing a card played at some point in a game. The examples in Figures 3 and 7 are of this type. If this type of training example is used, it will be necessary to determine how to represent the current game situation. An especially important issue concerns how to represent the past history of the game. An alternative approach would consider a whole series of card plays to constitute a single training example. Sometimes a series of card choices can be viewed as a "plan", e.g., a plan for getting "void" in some suit. This type of example appears to require a more complex representation.

Another representation issue concerns the initial intractable hearts domain theory. The theory was described above in terms of certain types of mini-max search procedures. In order to build explanations, however, a declarative representation is needed. A declarative representation might be obtained by replacing a procedure for searching a tree with a set of facts that describe the structure of the search tree as a formal object. It may also be possible to represent the hearts theory in a manner that is unrelated to mini-max search. Such an alternate theory might be constructed by attempting to directly encode the rules of the game as they are described in [Gibson 74]. The initial theory might also contain information not included in the rules. Facts about combinatorics and probability theory might be included, since this information is often used by human hearts players. It might also be useful to include some information about human game playing behavior.

3.6.3 Heuristics for Playing Hearts

A heuristic version of the hearts domain theory has already been implemented. This theory consists of a set of rules for choosing cards in various game playing situations. The rules are described informally in Appendix II. They were obtained by analyzing the protocols of hearts games played by humans. The implemented rule set uses a Horn clause representation suitable for use by a backward chaining theorem prover. The theorem prover and rule set have been incorporated into a performance program that actually plays the game of hearts.

The heuristic hearts theory can be used in several ways to advance the proposed research. To begin with, the heuristics can help with the process of finding useful simplifying assumptions in the hearts domain. By comparing the heuristics with the official rules of hearts, one attempts to determine what assumptions must be added to the official rules in order to derive the heuristics. Some of the assumptions shown in Appendix I were obtained through such a process. The heuristics may also be used to define an objective for the learning process. A learning system might be given the task of learning a set of simplifying assumptions that are equivalent to these heuristics. The heuristic rule set might also provide a source of training examples. Examples could be generated by using the heuristic rules in the game playing program. In each situation the rules would suggest a set of possible cards to play. The recommended cards would be taken as positive examples. The cards not recommended would be taken as negative examples.

3.6.4 A Focus for Experimentation

This research might best proceed by focusing on one part of the hearts game. The task of learning rules for "leading" is one possible focus of attention. Consider the following two heuristics for choosing a card to lead the trick:

- Play any card of lowest possible rank.
- Play any card in your shortest suit other than hearts.

These two heuristics may both be explained as attempts to avoid taking points in the current trick. The first one attempts to minimize the odds of taking the trick while ignoring the odds that hearts will be played during the trick. The second one seeks to minimize the odds that hearts will be played while ignoring the odds of taking the trick. These different heuristics can be each be explained by making different assumptions. The first heuristic can apparently be derived by ignoring the suits of all cards. The second heuristic can apparently be derived by ignoring the ranks of all cards.[7] The foregoing discussion suggests a reasonable objective of building a system that learns one or the other of these two heuristics, depending on the set of training

[7]Both explanations also seem to require forgetting about past tricks and ignoring the interaction between the current trick and future tricks. The details of the proofs remain to be worked out.

examples that is observed.

3.7 Alternate Domains

Although this research will focus mainly on the hearts domain, some other domains will be examined in order to validate the learning techniques to be developed. The system may not be implemented in other domains. Nevertheless, it will be important to demonstrate that the techniques apply in principle to other domains. In particular, the typology of "optimistic" assumptions should be examined in the context of alternate domains. This is especially important since the "optimistic" assumption types are claimed to be domain independent. Many of the assumptions involve the idea of abstraction in one form or another. This suggests looking at a domain in which people have previously studied abstraction and abstraction spaces. Domains under consideration include other card games, circuit design, job-shop scheduling, floor plan generation, tiling problems, the blocks world and others.

3.8 The POLLYANNA System

The ideas described above will be implemented in a computer program called "POLLYANNA".[8] POLLYANNA is designed to take two inputs including (1) a series of training examples and (2) an intractable domain theory.[9] POLLYANNA faces the task of finding a set of simplifying assumptions that are useful for the purpose of explaining the observed training examples.

3.8.1 The Components of POLLYANNA

The POLLYANNA system can be divided into several main components. Although the precise relationship between the modules has not yet been determined, they can be described in a general way. The "explanation builder" attempts to build explanations of training examples. For this purpose it makes use of the "assumption schemata" as well as an "assumption schemata instantiator" to guide the process of adopting assumptions. The "learning component" processes the explanations built by the "explanation builder". Three methods of processing explanations are currently under consideration. These include a procedure called "voting", a method based on "failure-driven learning" and an application of "dependency-directed backtracking". To be a complete system, POLLYANNA includes a "performance element" that can apply the results of learning to actual problems from the domain under study.

---

[8] A perennial optimist.

[9] Possibly the "assumption schemata" should be counted as a third input. This depends on the degree to which they can be written in a domain independent form.

3.8.2 The Explanation Builder

The "explanation builder", EB, has the task of taking observed examples as input and creating explanations of them. In order to build the explanations, the EB module will be forced to use simplifying assumptions. In principle the explanation builder is required to find all possible explanations of the observed instances, along with all the simplifying assumptions that might possibly be useful in explaining the examples. This criterion has the effect of factoring the learning process into two parts. The EB module finds all the "optimistic" assumptions that can help explain a given training example. Other learning modules will determine which of these candidates have greatest "explanatory power" measured across multiple examples. The explanation builder has two types of information at its disposal. One type of information consists of the initial intractable domain theory. The second source of information consists of the "assumption schemata" that describe which assumptions are considered to be "optimistic". [10] One possible approach would design the EB module as a backward chaining theorem prover. [11] The observed example would be proposed as a goal. The system would chain backward until satisfying termination conditions of (1) proposing a goal matching a fact describing the current game situation or (2) proposing a goal that is an instantiation of one of the "assumption schemata". If this design strategy does not pan out, the EB module may be somewhat more complicated.

3.8.3 The Voting Method

Once the explanation builder finds a set of "optimistic" candidate assumptions, some process must actually select the ones with the most explanatory power. One approach to this problem might be called the "voting" procedure. The voting procedure would involve processing multiple explanations. Each explanation would explain one example, E, possibly using several assumptions A1...AN. In the course of processing an explanation, the voting procedure would record the fact that E depends on A1...AN. The recorded dependencies indicate that E "votes" for each of A1....AN. A similar method could be used to process explanations that lead to contradictions. If assumptions A1...AN lead to a contradiction with example E, then the example E would cast negative votes for each of A1...AN. The voting procedure is similar to the one suggested by Lebowitz to update confidence levels of explanation rules in the explanation-based version of UNIMEM [Lebowitz 86].

In general, one can say that an assumption is more likely to be true if it has a high

---

[10] A possible third source of information would consist of the set of assumptions already adopted or the assumptions under consideration and their associated confidence levels.

[11] The current version of the explanation builder is essentially the same as a backward-chaining Horn clause theorem prover.

number of votes than if it has low number of votes. Nevertheless, the precise method of tabulating and utilizing the votes has not yet been determined. A simple experiment might use the "voting" approach to choose among a fixed small set of assumptions. Given a predefined set of assumptions, A1...AN, the system could process a set examples and collect votes for and against the assumptions in the set. The assumption or group of assumptions with the most votes would be adopted.[12]

3.8.4 A Role for Failure-Driven Learning

Another approach would make a distinction between "finding assumptions" and "correcting erroneous assumptions". This distinction is important because the techniques of "failure-driven learning" [Schank 82] appear to be relevant to the problem of correcting assumptions that have been contradicted by an observed instance. This method would work by (1) observing an example that contradicts an adopted assumption (2) explaining why the contradiction occurred and (3) using the explanation to limit the scope of the faulty assumption.

As an example, suppose that POLLYANNA has adopted an assumption, A1, asserting that "Only the numbers of cards out in each suit need be remembered, and all other parts of the game history can be forgotten". Suppose further that A1 gets used in an explanation advising playing a club card C1, because more clubs are out than any other suit. The explanation would be using "number of cards out" as a guide to determining the odds of opponents being void in any given suit. Now suppose that this recommendation is contradicted by an example in which the teacher plays a spade card C2. POLLYANNA might explain the choice C2 by using a weaker assumption, A2, asserting "Only the numbers of cards out in each suit *and the suits in which someone has shown void* need be remembered, and all other parts of the history can be forgotten". The teacher played a spade because someone had shown a void in clubs, indicating that "number of cards" out was not a reliable guide to the odds of players being void.

The scope of the faulty assumption, A1, can be limited in the following way. Let E be an explanation using assumption A2 that explains explains why card C1 is wrong. In effect, E explains why the contradiction occurred. To put it more technically, E must explain why C1 is a negative instance. POLLYANNA would then generalize the explanation E using traditional EBL methods. Traditional EBL methods can be used to find the "generalized antecedents" of an explanation. These are the most general facts that must be true for an explanation to succeed.

---

[12]This type of procedure might work well on the problem of learning which parts of the game history should be remembered and which parts can be forgotten. An experiment might be run with just two predefined assumptions. The assumptions A1 and A2 could represent hypotheses that two distinct aspects of the history can be safely forgotten. A1 might say: "Only remember the number of cards out in each suit". A2 might say "Only remember which suits are the ones in which people have shown themselves to be void".

This process would yield general conditions under which the explanation E would apply. These conditions also indicate when assumption A1 would fail. If R represents the conditions under which explanation E will apply and A1 will fail, then POLLYANNA would replace assumption A1 with "A1 unless R".

As an additional example, suppose POLLYANNA has already adopted an assumption, A, that says: "The choice of a card in the current trick will have no effect on the number of hearts taken in future tricks." Suppose further that this assumption is contradicted by an observed example in which the teacher takes hearts in the current trick by playing card C2, despite the fact that he could have played a card C1 that would lose the current trick. POLLYANNA might explain the contradiction by performing a very limited look-ahead search, perhaps only looking one trick ahead. By generalizing an explanation based on this look-ahead search, POLLYANNA could limit the range under which the faulty assumption, A, gets applied.

POLLYANNA's failure-driven learning process can be summarized in the following way. POLLYANNA would start with a very simple theory T1. When theory T1 leads to a contradiction with an observed example, POLLYANNA will expend the extra computational effort required to explain the contradiction using a more sophisticated theory T2. Theory T2 might differ from T1 by (1) remembering more information from the game history, (2) considering more types of goal interactions, (3) using a less abstracted model or (4) looking farther ahead in a search tree, etc. The explanation of the contradiction is then generalized to find conditions under which theory T1 will fail. The process outlined here is similar to DeJong's notion of "explanation-based concept refinement" [Dejong and Mooney 86].

3.8.5 A Role for Dependency-Directed Backtracking

A final learning method would use techniques for dependency-directed backtracking [Doyle 79]. Suppose POLLYANNA has a heuristic rule, R, that leads to choosing a wrong card. A derivation of the rule R would be retrieved. The derivation might use assumptions A1...AN. POLLYANNA would try to evaluate each of A1...AN to see which are true in the current situation. Perhaps each assumption could be directly compared to the facts of the current situation. Alternatively, POLLYANNA could try retracting various subsets of the assumptions until finding a specific subset, the retraction of which leads to recommending the correct card choice. The derivation is useful in this context for "focusing" the system's attention on the faulty assumptions. This approach is similar to one used by Smith, et al., in a Learning Apprentice System for the Dipmeter Advisor [Smith et al. 85]. Some related methods are discussed by Tadepalli in [Tadepalli 85].

3.8.6 Implementation Goals

This section describes a set of goals for implementing the POLLYANNA system. In particular, it describes which parts of the system will be implemented and which will not. It also describes which parts are currently implemented.

- The knowledge base of POLLYANNA will be implemented. This knowledge base will consist of (1) the initial intractable hearts theory and (2) a set of "assumption schemata". Implementing the knowledge base may well require changing the knowledge representation mechanism used in the existing implementation.
- The "explanation builder" will be implemented. A version of the "explanation builder" has already been built. This version will have to be modified for a couple of reasons. It must be changed in order to make use of the "assumption schemata". It may also be changed to handle new representations. The final version of the "explanation builder" may require interactive guidance from a human.
- The "learning component" will be implemented. As described above, this module will involve some combination of the "voting", "failure-driven learning" and "dependency-directed backtracking processes" described above. A portion of the "learning component" has already been built. This piece consists of several procedures for generalizing explanations. The generalization procedures will be used by the "failure-driven learning" process as described above.
- The "assumption schemata instantiator" might or might not be implemented. As described above, the "assumption schemata" might be represented in a domain dependent form, bypassing most of the task of instantiating domain independent versions.
- The "performance element" might or might not be implemented. This depends on the extent to which it can use the same procedures as used by the "explanation builder". A version of the "performance element" already exists. It uses essentially the same procedures as the existing "explanation builder". The "performance element" will only remain operational if it can be easily maintained as the "explanation builder" is modified.

3.9 Research Agenda

This section lists some tasks that must be completed in the course of the proposed research. Some of these tasks have already been completed. The others will be performed in roughly the order shown.

1. Collect protocols of hearts games played by humans. The protocols should contain records of the cards played as well as the players' verbal explanations of their card choices. (Completed).
2. Formulate one or more sets of heuristic rules for playing hearts by analyzing the hearts game protocols. Also find game playing heuristics in books on hearts game strategy. (In Progress.)
3. Find useful simplifying assumptions for the hearts game by analyzing both the explanations in the protocols and the heuristic game playing rules. (In Progress.)
4. Determine which types of assumptions occur most commonly, and define a typology of useful simplifying assumptions. (In Progress).

5. Choose a setting in which the system can learn any one of several conflicting game playing heuristics. For each of the conflicting heuristic rules, determine (1) a set of training examples that reflect the rule and (2) a set of simplifying assumptions that could be used to derive the rule. (In Progress.)
6. Develop representations for the training examples, the initial hearts theory, the simplifying assumptions and the assumption schemata. (In Progress.)
7. Modify the explanation builder to make use of the assumption schemata and to handle any new representations that are needed.
8. Implement one or more of the "voting", "failure-driven learning" or "dependency-directed backtracking" procedures described above, depending on their appropriateness to the chosen experimental setting.
9. Summarize lessons learned from the implementation process. Evaluate the knowledge representations, the learning procedures, the optimistic assumption typology and the two criteria for choosing assumptions.
10. Investigate alternate domains to determine the generality and range of applicability of the optimistic assumption typology.

4 Significance of the Research

The research proposed here will be significant for several reasons. One major reason involves the relation of this project to the rest of the machine learning field. As described above, researchers in machine learning have come to emphasize the role of background knowledge in learning from examples. The proposed research presents a new method for using background knowledge. The new method uses explanations to help a system find simplifying assumptions.

The proposed research is also valuable because it addresses major deficiencies with explanation-based learning techniques. EBL methods only work with complete, correct, tractable domain theories. This limitation is serious, since such theories are not available for most domains. POLLYANNA is designed mainly to handle intractable theories. Problems of intractability occur quite often. In practice, all real life domain theories may be intractable. Nevertheless, the techniques used may well apply to other types of inadequate domain theories. The criterion of adopting assumptions based on their "explanatory power" would appear to be widely applicable.

The proposed research will address additional deficiencies of EBL. Existing EBL methods do not properly account for the role of training examples. Furthermore, they do not change the underlying content of a system's initial knowledge base. The POLLYANNA system proposed here will address both of these problems. As described above, an effort will be made to show that examples are indispensible. Furthermore, the "simplifying assumptions" that are learned by "POLLYANNA" are not necessarily contained in the deductive closure of the initial knowledge base. In many cases the learned assumptions are technically inconsistent with the initial

intractable theory!

A final point involves the types of assumptions to be learned by the POLLYANNA system. These include assumptions about "abstraction" and "decomposability", among others. Previous investigators have found these types of assumptions to be widely applicable. Polya has stressed the role of abstraction [Polya 57]. Simon has emphasized the importance of "nearly decomposable systems" [Simon 81]. If a system is nearly decomposable, then the assumption of total decomposability may be a pretty good approximation. This observation suggests that the learning techniques to be developed by this research will apply to a large class of problems.

5 Acknowledgement

Many thanks to Michael Lebowitz for assisting in the process of preparing this proposal.

I. Optimistic Assumptions for Hearts

Decomposing Assumptions:

- Events in all future tricks will be the same regardless of the card choice in the current trick.
- Cards are not used up as they are played. After each trick the players retrieve the cards they played.

Independence Assumptions:

- The value of a hand depends only on the sum of the ranks of the cards in the hand.
- The value of a hand depends only on the number of hearts in the hand.

Abstracting Assumptions:

- The suits of all cards can be ignored, except for the distinction between hearts and non-hearts.
- The ranks of all cards can be ignored. Only the suits matter.
- Remember only the number of hearts played.
- Remember only the numbers of cards played in each suit.
- Remember only the suits in which a player has shown himself to be void.
- Remember which players have shown themselves to be void in which suits and forget all other information about the game history.
- Remember the number of times that each suit was led. All other information about the game history can be forgotten.
- Remember all the cards that have been played, but do not remember who played which cards and when they were played.

II. Heuristic Rules for Playing Hearts

The following set of rules implements a relatively naive strategy for playing hearts. They choose cards to play without using any "memory" of the previous tricks. The card choice is made to satisfy the primary objective of avoiding taking hearts during the current trick. Subject to this constraint, the rules suggest dumping dangerous cards. The first priority is to dump hearts, preferably high ranking hearts. The second priority is to dump high ranking cards of other suits.

Conditions:

- L: You're the leader.
- VLS: You're void in the lead suit.
- HH: You have hearts in your hand.
- HOH: You have only hearts in your hand.
- LP: You're the last player for the current trick.
- HS: At least one heart has been played in the current trick.
- PU: You have at least one card in the lead suit which is less than some card in the lead suit which has already been played in the current trick.

Actions:

- PHH: Play your highest heart.
- PHC: Play your highest card of any suit.
- PLC: Play your lowest card of any suit.
- PHLS: Play your highest card in the lead suit.
- PHU: Play your highest card in the lead suit which is less than the highest card in the lead suit which was already played in this trick.
- PLLS: Play your lowest card in the lead suit.
- PLNH: Play your lowest card which is not a heart.

Rules:

- (L and HOH) $\Rightarrow$ PLC
- (L and $\neg$HOH) $\Rightarrow$ PLNH
- ($\neg$L and VLS and HH) $\Rightarrow$ PHH
- ($\neg$L and VLS and $\neg$HH) $\Rightarrow$ PHC
- ($\neg$L and $\neg$VLS and LP and $\neg$HS) $\Rightarrow$ PHLS
- ($\neg$L and $\neg$VLS and LP and HS and PU) $\Rightarrow$ PHU
- ($\neg$L and $\neg$VLS and LP and HS and $\neg$PU) $\Rightarrow$ PHLS

- $(\neg L \text{ and } \neg VLS \text{ and } \neg LP \text{ and } PU) \Rightarrow PHU$
- $(\neg L \text{ and } \neg VLS \text{ and } \neg LP \text{ and } \neg PU) \Rightarrow PLLS$

References

[Cohen and Feigenbaum 82] Cohen, P. R. and Feigenbaum, E. A. (Eds.). *The Handbook of Artificial Intelligence, Volume 3.* William Kaufmann, Inc., Los Altos, California, 1982.

[Dejong and Mooney 86] DeJong, G. and Mooney, R. "Explanation-Based Learning: An Alternative View." *Machine Learning 1*, 1986, pp. 145 - 176.

[Dietterich 86] Dietterich, T. G. "Learning at the Knowledge Level." *Machine Learning 1*, 1986, pp. 287 - 315.

[Doyle 79] Doyle, J. "A Truth Maintenance System." *Artificial Intelligence 12*, 1979, pp. 231 - 272.

[Ellman 85] Ellman, T. Generalizing Logic Circuit Designs by Analyzing Proofs of Correctness. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, California, 1985.

[Ellman 87] Ellman, T. P. Explanation-Based Learning: A Survey. Unpublished Manuscript

[Gibson 74] Gibson, W. B. *Hoyle's Modern Encyclopedia of Card Games.* Doubleday and Company, Inc., Garden City, NY, 1974.

[Lebowitz 86] Lebowitz, M. "Integrated learning: Controlling explanation." *Cognitive Science 10*, 2, 1986, pp. 219 - 240.

[Michalski 83] Michalski, R. S. "A theory and methodology of inductive learning." *Artificial Intelligence 20*, 1983, pp. 111 - 161.

[Michalski et al. 83] Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., eds. *Machine Learning, An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA, 1983.

[Mitchell 80] Mitchell, T. M. The Need for Biases in Learning Generalizations. Technical Report CBM-TR-117, Rutgers University, New Brunswick, NJ, 1980.

[Mitchell et al. 86] Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, S. T. "Explanation-Based Learning: A Unifying View." *Machine Learning 1*, 1986, pp. 47 - 80.

[Nillson 80] Nillson, N. J. *Principles of Artificial Intelligence.* Tioga Publishing Company, Palo Alto, California, 1980.

[Pazzani et al. 86] Pazzani, M., Dyer, M. and Flowers, M. The Role of Prior Causal Theories in Generalization. Proceedings of the Fifth National Conference on Artificial Intelligence, Los Altos, CA, 1986, pp. 545 - 550.

[Polya 57] Polya, G. *How to Solve it.* Doubleday Anchor Books, New York, New York, 1957.

[Pople 73] Pople, H. E. On the Mechanization of Abductive Logic. Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford, California, 1973.

[Schank 82] Schank, R. C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, New York, 1982.

[Simon 81] Simon, H. A. *The Sciences of the Artificial.* MIT Press, Cambridge, MA, 1981.

[Smith et al. 85] Smith, R. G., Winston, H. A., Mitchell, T. M. and Buchanan, B. G. Representation and Use of Explicit Justifications for Knowledge Base Refinement. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, 1985, pp. 673 - 680.

[Tadepalli 85] Tadepalli, P. V. Towards Learning Chess Combinations. Unpublished Paper

[Utgoff 86] Utgoff, P. E. Shift of Bias for Inductive Concept Learning. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Ed., *Machine Learning: An Artificial Intelligence Approach, Volume II*, Morgan Kaufmann, Los Altos, CA, 1986, pp. 107-148.

[Waldinger 77] Waldinger, R. Achieving Several Goals Simultaneously. In E. Elcock and D. Michie, Ed., *Machine Intelligence 8*, Ellis Horwook, Limited, London, 1977.