THE CLASSIFICATION, DETECTION AND HANDLING OF IMPERFECT THEORY PROBLEMS

Shankar Rajamoney
Gerald DeJong

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

In recent years knowledge-based techniques like explanation-based learning, qualitative reasoning and case-based reasoning have been gaining considerable popularity in AI. Such knowledge-based methods face two difficult problems: 1) the performance of the system is fundamentally limited by the knowledge initially encoded into its domain theory 2) the encoding of just the right knowledge to enable the system to function properly over a wide range of tasks and situations is virtually impossible for a complex domain. This paper describes research directed towards the construction of a system that will detect and correct problems with domain theories. This will enable knowledge-based systems to operate with imperfect domain theories and automatically correct the imperfections whenever they pose problems. This paper discusses the classification of imperfect theory problems, strategies for their detection and an approach based on experiment design to handle different types of imperfect theory problems.

I INTRODUCTION

This paper addresses the problem of imperfect theories in AI systems. It is increasingly apparent that knowledge is essential for intelligent behavior. This has led to a new trend in AI towards knowledge-intensive methods like explanation-based learning [1, 2], qualitative reasoning [3], and case-based reasoning [4, 5].

The primary shortcoming of these approaches is not in the representation of the knowledge - a task that is relatively well understood - but in the subtleties of selecting the appropriate knowledge. The expert who is handcoding the knowledge has to anticipate the rich variety of tasks and the wide range of situations for which the knowledge may be used in order to insure that the system will function properly. Also, all AI systems that rely on a programmer-specified domain theory are fundamentally limited by their initial knowledge. For example, [6] shows how the knowledge built into a learning system drastically influences its learning capability.

What is needed is a system that will automatically detect and correct problems with its domain theory. This will free the expert from the tedious and often impossible task of handcoding all the relevant knowledge. It will enable the use of "quick and dirty" methods to facilitate the construction of operational but imperfect domain theories. These domain theories can then be automatically debugged and corrected by the system.

Mitchell et al. [1] have briefly classified problems with imperfect domain theories into three categories:

(1) the *incomplete theory problem*: the deductions required cannot be computed because relevant information is missing.

(2) the *inconsistent theory problem*: the system can derive inconsistent statements from its theory.

(3) the *intractable theory problem*: the deductions are computationally prohibitive and hence cannot be completed.

However, the underlying issues are too murky and subtle for the above categories to be cleanly separable. For example, inconsistencies and incompleteness in domain theories may be due to abstractions and approximations which make the theory tractable [7]. Inconsistent theory problems can be due to an incomplete theory if information necessary to nullify one of the inconsistent statements is missing. Inconsistent statements can also result from the incomplete theory problem if the system is operating under the closed world assumption and does not consider the possibility of new information influencing its computations [8]. Apart from the above problems of interacting categories, the classification of Mitchell et al. also ignores certain kinds of incompleteness and inconsistencies.

A complete taxonomy of imperfect theory problems includes two types of incompleteness and inconsistencies. The **first type of incompleteness** is the one discussed above in which a deduction cannot be completed because some relevant knowledge is missing. The **second type of incompleteness** is due to the lack of sufficient detail in the relevant knowledge. Unlike the first case, deductions can be constructed leading to a conclusion. However, the lack of detail results in the system having to make assumptions and leads to the problem of multiple mutually inconsistent proofs for a conclusion. This type of incompleteness also results in large search spaces because the system does not have the required control knowledge to select the correct path at each choice point. The **first type of inconsistency** involves wrong knowledge that has to be identified and retracted. The **second type of inconsistency** involves missing knowledge that would have defeated the deduction leading to one of the inconsistent statements.

There are two aspects to the imperfect theory problems - detection of the imperfections and the revision of the domain theory - and both of these present difficulties. This paper describes various strategies for detecting problems with the domain theory and a uniform approach based on experiment design to handle each type of problem. The system is assumed to start with an initially imperfect but operational theory. This is a psychologically motivated assumption since people also use simplified domain theories to make conclusions computationally tractable and they are still able to operate satisfactorily. During the course of the system's operation, problems with its domain theory are identified and corrected. No changes are made until a problem is detected.

II DETECTION OF THE IMPERFECT THEORY PROBLEMS

This section describes four strategies for detecting problems with domain theories. Though the detection strategies are discussed in the context of *explanation construction* for

This research was supported in part by a University of Illinois Cognitive Science/Artificial Intelligence Fellowship and in part by the Office of Naval Research under grant N-00014-86-K-0309.

explanation-based learning [1, 2] they are also applicable for other problem solving tasks like qualitative reasoning and planning. Explanation construction involves using facts and rules from the domain theory to show *why* a training instance is an example of the goal concept (Figure 1a). The problems due to imperfect domain theories that are encountered during explanation construction are:

Broken Explanation: There are gaps in the explanation leading to a broken explanation (Figure 1b). The rules or facts that are required to complete the explanation are missing from the domain theory (incompleteness - type I).

Contradiction: The system constructs explanations for conclusions which are contradictory (Figure 1c). This problem may be due to wrong rules or facts in the domain theory (inconsistency - type I) or due to missing rules or facts (inconsistency - type II) that would resolve the contradiction by defeating one of the explanations (e1 or e2) thereby leading to the withdrawal of the corresponding previously justified conclusion (P or (not P)).

Multiple Explanations: The system constructs multiple explanations for a conclusion when only one explanation is expected to be true in the real world (Figure 1d). This problem is due to lack of knowledge which would help distinguish between the alternate explanations (incompleteness - type II). This problem is especially important for explanation-based learning as is has implications for the new concept definition.

Resources Exceeded: The system exceeds the resources (time, memory, etc.) allotted to it while constructing an explanation. This type of problem can be further classified as:

Large Search Space Problem: The system has to search a large space during the construction of an explanation (Figure 1e). Though the explanation may exist and its size may be comparable to previous successful explanations the system cannot construct it since there are too many paths to explore. The system does not have the knowledge to decide between the alternate paths (incompleteness - type II) and is forced to search all paths.

Small Links Problem: The links connecting the explanation are too small and too many for the system to construct the complete explanation within the allotted resources (Figure 1f) (intractable theory problem). This problem is independent of the large search space problem and may occur even when no search is involved.

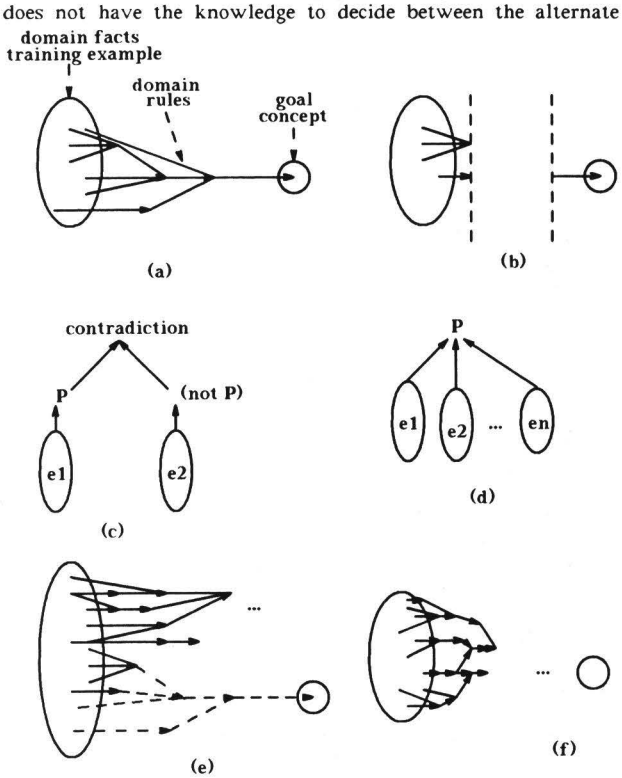

Figure 1: (a) a typical explanation (b) a broken explanation (c) a contradiction (d) multiple explanations (e) large search space problem (f) small links problem.

III DEALING WITH THE IMPERFECT THEORY PROBLEMS

Dealing with the above problems requires the acquisition of new knowledge. This section describes ongoing research on an extension to an approach discussed in [8, 9] that can be used to deal with the above problems.

A. A Brief Review of the Experiment Design Approach

An approach that deals with the contradiction problem due to an inconsistent domain theory (type II) is described in [8, 9]. The approach involves: 1) Monitoring the execution of the system's plans. 2) Detection of contradictions if the system's predictions are not compatible with the observations. 3) Hypothesizing reasons which could resolve the contradiction. 4) Designing experiments to test each hypothesis. 5) Incorporating the information obtained by the experiments into the domain theory. Five classes of experiments are described in [8]. These experiments are used to discriminate among hypotheses, perform measurements, find, dependencies among parameters, classify objects based on their behavior with respect to a property and define new properties of objects based on their behavior in a situation. These experiments are used to obtain new knowledge that is relevant to the determination of the correct hypothesis.

B. Extending the Experiment Design Approach

The experiment design approach can be applied to each of the problems described in section 2:

Broken Explanation: The system must be able to hypothesize different ways of filling the gaps in the explanations. In [8, 9] the hypotheses were suggested by the system after an analysis of the situation that led to the failure. Alternatively such hypotheses may be formed by analogy to previous experiences [10]. Once alternate hypotheses that can complete the explanation have been formulated experiments are designed to determine the best hypothesis.

Contradiction: Experiments are designed to test each link in each explanation to isolate the faulty rule or fact that leads to the contradiction. Once the fault has been isolated then hypotheses are formulated to correct the fault. If the contradiction is due to wrong rules or facts (inconsistent - type I) then the hypotheses can involve retraction of rules. If the contradiction is due to missing knowledge (inconsistent - type II) then the hypotheses can involve positing rules that defeat the explanation. Experiments are designed to identify the best hypothesis.

Multiple Explanations: Multiple explanations arise due to the lack of knowledge required to distinguish between the alternative explanations (incompleteness - type II). Experiments are designed to gather the information that the system needs to decide which explanations cannot hold for the given situation. This will enable it to determine the correct explanation.

Resources Exceeded: The large search space problem can be handled by designing experiments to gather the information needed to make the right choice whenever alternatives develop. A number of approaches have been suggested for the small links problem [7, 11-13]. [13] shows how approximations can be used to make explanations tractable. [12] describes an incremental failure-driven technique to refine abstract theories when the current theory fails to provide a satisfactory explanation. The approach suggested by [7, 11] involves describing the domain theory at different levels of abstraction. This allows the explanation to be constructed using fewer higher-level links. How-

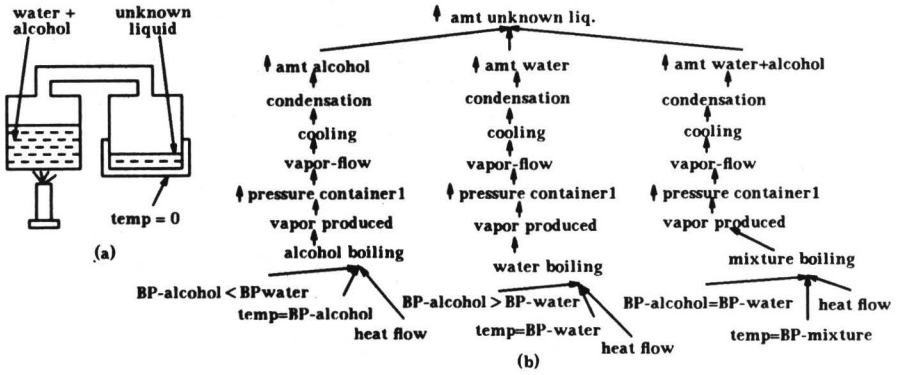

Figure 2: An example illustrating the multiple explanations problem due to incomplete knowledge.

ever, due to the abstractions and approximations a number of alternate low-level explanations may be possible for one higher-level explanation and this failure cannot be handled by examining the more detailed levels. This is the "hierarchical" multiple explanation problem and the experiment design approach can be applied to find the correct explanation.

C. An Example

The system is given the distillation scenario shown in Figure 2. A mixture of alcohol and water is heated and it is observed that an unknown liquid is formed in the second container and that its amount is increasing. The domain theory does not have rules or facts that allow the system to determine which liquid will boil first (incomplete - type II) and therefore it has to take into account all possibilities. The system constructs three different explanations for the increase in the amount of the liquid in the second container (the multiple explanation problem). For example, if the boiling point of alcohol is less than that of water then when the temperature of the mixture reaches the boiling point of alcohol the heat flow to the mixture will cause alcohol to boil. Boiling will produce alcohol vapor which will cause the pressure in the container to increase. The pressure will become greater than the pressure in the second container and there will be a flow of alcohol vapor to the second container. This vapor will cool and condense since the second container is at a very low temperature. The condensing alcohol forms the explanation for the observed formation and increase in the amount of the unknown liquid. Similarly, if the boiling point of alcohol is less than or equal to that of water then water or a mixture of alcohol and water will condense in the second container. It is important to determine which explanation is correct since the explanation is worth generalizing and learning only if a useful goal is being achieved - for example, if alcohol is condensing then we have separated alcohol from water or obtained a purer version of alcohol (distillation). The system identifies the correct explanation by designing experiments to determine whether the liquid formed in the second container is water, alcohol or a mixture of both. This example also illustrates the large search space problem if the above task is part of a much larger task - like understanding a distillation factory - that builds in separate directions on each explanation. Then the above experiments help in pruning the search space by immediately eliminating two of the three choices for the unknown liquid. The system can also design experiments to select the correct path during explanation construction by determining independently whether the boiling point of water is greater than, equal to or less than that of alcohol and applying that information to the given situation.

IV CONCLUSIONS

In this paper we have discussed problems with and extensions of Mitchell et al.'s classification of imperfect theory problems. Four strategies for detecting imperfections in domain theories were described. A uniform approach for handling these problems based on experiment design was also described and illustrated by an example. These methods were discussed in the context of explanation construction for explanation-based learning. However the detection strategies and the experiment design approach are general and can be applied to other knowledge-intensive AI areas like case-based reasoning, expert systems and qualitative reasoning.

ACKNOWLEDGMENTS

We would like to thank Ray Mooney and Steve Chien for their helpful comments on drafts of this paper.

REFERENCES

1. T. M. Mitchell, R. Keller and S. Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View," *Machine Learning 1*, 1 (January 1986), pp. 47-80.
2. G. F. DeJong and R. J. Mooney, "Explanation-Based Learning: An Alternative View," *Machine Learning 1*, 2 (April 1986), .
3. K. D. Forbus, "Qualitative Process Theory," Technical Report 789, Ph.D. Thesis, MIT AI Lab, Cambridge, MA, August 1984.
4. R. C. Schank, *Dynamic Memory*, Cambridge University Press, Cambridge, England, 1982.
5. C. Stanfill and D. Waltz, "Memory-Based Reasoning," Technical Report 86-7, Thinking Machines Corporation, Cambridge, MA, March 1986.
6. P. E. Utgoff, "Shift of Bias for Inductive Concept Learning," in *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Morgan Kaufmann, Los Altos, CA, 1986.
7. R. Doyle, "Constructing and Refining Causal Explanations from an Inconsistent Domain Theory," *AAAI-86*, .
8. S. A. Rajamoney, "Automated Design of Experiments for Refining Theories," M. S. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, May 1986.
9. S. Rajamoney, G. F. DeJong and B. Faltings, "Towards a Model of Conceptual Knowledge Acquisition through Directed Experimentation," *IJCAI-85*, .
10. B. Falkenhainer, "An Examination of the Third Stage in the Analogy Process: Verification-Based Analogical Learning," *IJCAI-87*,
11. P. V. Tadepalli, "Learning in Intractable Domains," in *Machine Learning: A Guide To Current Research*, T. M. Mitchell, J. G. Carbonell and R. S. Michalski (ed.), Kluwer Academic Publishers, Hingham, MA, 1986, pp. 337-342.
12. S. A. Chien, "Extending Explanation-Based Learning: Failure-Driven Schema Refinement," *3rd IEEE Conference on AI Applications*, .
13. S. W. Bennett, "Approximation in Mathematical Domains," *IJCAI-87*, .