ISI Research Report
ISI/RR-89-247
January 1990

4

AD-A220 427

University of Southern California

Leonard Friedman
David P. Benjamin

Generation of Expert Systems by Partial Evaluation

INFORMATION SCIENCES INSTITUTE

213/822-1511

4676 Admiralty Way/Marina del Rey/California 90292-6695

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT: This document is approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): ISI/RR-89-247

5. MONITORING ORGANIZATION REPORT NUMBER(S): -------

6a. NAME OF PERFORMING ORGANIZATION: USC/Information Sciences Institute

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION: -------

6c. ADDRESS *(City, State, and ZIP Code)*: 4676 Admiralty Way, Marina del Rey, CA 90292-6695

7b. ADDRESS *(City, State, and ZIP Code)*: -------

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Air Force Logistics Command

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F33600-88-C-0028

8c. ADDRESS *(City, State, and ZIP Code)*: Wright-Patterson Air Force Base, OH 45433-5320

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
--- ---	-------	-------	-------

11. TITLE *(Include Security Classification)*: Generation of Expert Systems by Partial Evaluation (Unclassified)

12. PERSONAL AUTHOR(S): Friedman, Leonard; Benjamin, David P.

13a. TYPE OF REPORT: Research Report

13b. TIME COVERED: FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*: 1990, January

15. PAGE COUNT: 16

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
09	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*: machine learning and diagnosis, partial evaluation

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

In this report, Partial Evaluation (PE) is shown to be a useful technique for machine learning. Our implemented version, LDS, generates expert diagnostic knowledge bases using PE. One of the benefits of the system is to simplify knowledge acquisition. PE is applicable whenever the problems to be solved share reasoning methods that can be represented by a common model with variables, and the knowledge to partially instantiate those variables is readily available.

The general model used in this report is a causal model of the failures observed in power supplies, supplemented by knowledge that links structures to function. Unlike Explanation-based Learning, learning by Partial Evaluation proceeds by specializing a general model rather than generalizing a specific example. The output of learning is a specific causal model for an input circuit and is ready to be used by the performance system, which is an expert diagnostician.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: Victor Brown Sheila Coyazo

22b. TELEPHONE *(Include Area Code)*: 213/822-1511

22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

ISI Research Report
ISI/RR-89-247
January 1990

University of Southern California

Leonard Friedman
David P. Benjamin

Generation of Expert Systems by Partial Evaluation

INFORMATION SCIENCES INSTITUTE
213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

This research was sponsored by the Air Force Logistics Command under Contract No. F33600-88-C-0028. Views and conclusions contained in this report are the authors' and should not be interpreted as representing the official opinion or policy of the Air Force Logistics Command, the U.S. Government, or any person or agency connected with them.

1 Introduction

We have developed a learning system that uses the well-known technique of Partial Evaluation (PE) (Kahn 1984) to automatically generate the complete knowledge base for an expert system diagnostician. PE is well suited to application areas with multiple problem domains that share common lines of problem-solving reasoning, which can be specialized with appropriate auxiliary information for a problem domain.

An important benefit of PE, as we use it, is that it eases the knowledge acquisition bottleneck in expert system development. In our Learning Diagnostic Skills (LDS) system, PE constructs an efficient diagnostic model for each given schematic from a general causal model of power-supply failures. Because the general model is reused for each schematic, knowledge acquisition for a new power supply is limited to adding features not already covered by the general model. The use of PE compensates for the computational costs caused by the generality of the initial knowledge and for the use of an expressive representation.

The report begins with a broad definition of PE. The relationship of PE to explanation-based learning is pointed out, followed by a description of the application of PE to constructing the specialized causal models that are employed in the performance of diagnosis in the LDS system.

2 The Role of PE in AI

As noted in both (Kahn 1984) and (van Harmelen and Bundy 1988), Partial Evaluation has mainly been applied in programming environments in order to optimize programs by generating compiled code from interpreted code, given a specification of the interpreter; this is an efficient way to extend languages such as PROLOG. Doing so speeds up the procedural part of a system. Our system uses PE as a key speed-up learning component in applications. Learning by Partial Evaluation, is not a common practice in machine learning.

Typically, machine learning methods employ declarative representations of knowledge, which can be utilized by interpreters for a variety of purposes. For example, declarative knowledge can be more readily examined to check completeness or consistency than can procedurally encoded knowledge. We use PE on a declarative domain model to improve run-time efficiency before a problem is solved.

The Partial Evaluation Problem

The technique of partial evaluation can be applied to problems of the following form:

Given: General problem-solving knowledge for an application area, and
Given: Auxiliary information about a domain of problems within the application area,
Generate: A limited and more efficient domain-specific set of problem-solving knowledge.

Partial evaluation performs symbolic execution of the general problem-solving knowledge. This process restricts the range of values allowed for variables in the general knowledge base according to the auxiliary information. Where possible, variables are replaced with constants. The result is further processed to precompute values of constant expressions and to eliminate redundancy. This produces domain knowledge that efficiently combines the general and auxiliary knowledge.

The Relationship of PE to EBL

The methods of explanation-based generalization (EBL) and PE can be shown to be equivalent (van Harmelen and Bundy 1988). In EBL, a completed problem-solving episode initiates learning. Domain knowledge then guides the generalization of the problem solution. This learning can also be seen as specializing the knowledge used in constructing an explanation of the problem solution, though this viewpoint is seldom used explicitly. Compared with applying the learning algorithm to all available data beforehand, using problem instances to trigger learning reduces the creation of specialized knowledge that is unlikely to be used. An accepted heuristic is that previously needed knowledge is most likely to be needed again.

Likewise, in applying PE, problem-solving episodes trigger learning in such a way that no effort is expended on irrelevant knowledge. The natural viewpoint for PE is that general knowledge is specialized. Unlike EBL, PE is applied to a fragment of the general knowledge when problem solving first tries to access it. The specialized knowledge is then constructed and used in solving the triggering problem.

Both EBL and PE only generate speed-up improvements in system performance. The deductive closure (Dietterich 1986) of the knowledge in the system is unchanged.

3 Learning in the LDS System

The LDS system demonstrates the use of PE for learning in the application area of power supply diagnosis. A specialized domain is the diagnosis of failures for a particular power supply design. The general problem-solving knowledge covers standard (non-switching) power supplies. Figure 1 shows that there is no data feedback to the learning system from the performance system, an abduction diagnostician. The output of the learning system is a domain-specific causal model (SCM). The source of knowledge enabling the use of PE for learning is the general causal model (GCM). Currently, the annotated circuit schematic is entered by hand. It will be derived automatically from the basic circuit schematic when structure recognition, shaded in Figure 1, is implemented. As detailed later in this section, PE constructs enough of the SCM so that LDS can present a menu of possible "presenting" or initially observable symptoms to the user; PE then constructs branches of the SCM as required by abduction diagnosis. LDS has been applied to three distinct power supply schematics of increasing complexity.

The implemented learning and performance systems are planned to be part of a larger system that incorporates Case-based Reasoning (CBR), shaded at the bottom of Figure 1. We expect to demonstrate, by Case-based Reasoning, the acquisition of diagnostic skill in selecting the best next measurement to make while diagnosing a circuit, and to learn how to "jump to conclusions" when expectations warrant it. LDS will try to solve each problem by CBR before "falling back" to abduction diagnosis, which inherits its measurement ordering from the GCM. Further discussion of structure recognition and CBR is beyond the scope of this report.

Before describing the learning component, we give a brief overview of abduction diagnosis and the required circuit-specific causal model of failures. This model is the end product that

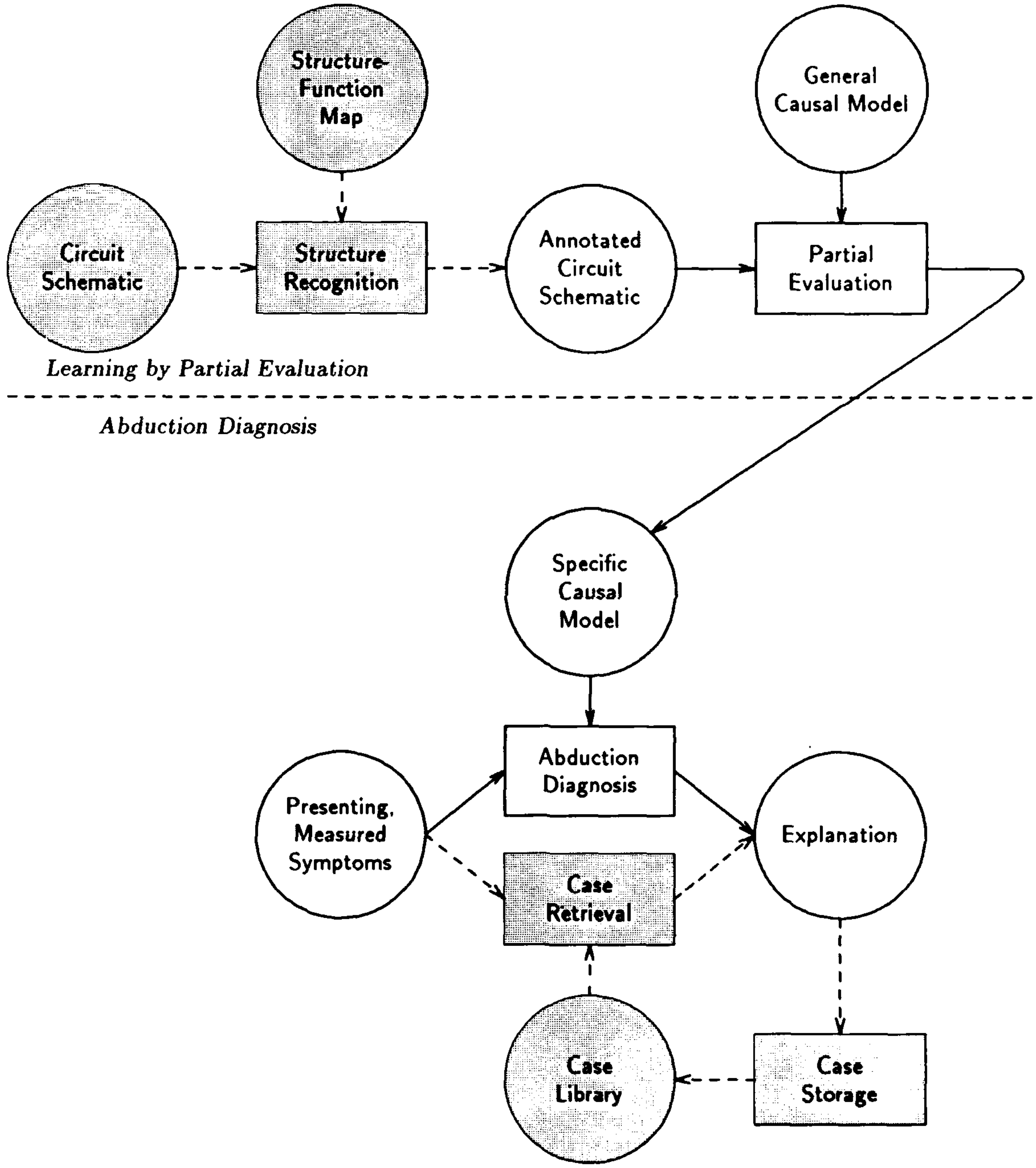

Figure 1: Data flow in LDS

must result from learning. The makeup of the GCM is then reviewed and the generation of the SCM by PE is explained.

The LDS Performance System

LDS uses an abduction algorithm to explain what repair is needed, given the observable failure symptoms and values for the measurements that have been requested. The domain-specific causal model (SCM) in LDS is based on the representation used in ABEL (Patil 1981a, 1981b). This is a network where hypotheses are linked by causal relationships and have "down" links to levels of greater detail. An explanation is a path through the causal model that has a presenting symptom at its head; each step along the path gives either a causal reason or greater detail explaining the preceding node. For a successful diagnosis, all tests along the path are satisfied and the path terminates at one or more events representing repairable causes.

Abduction diagnosis proceeds in a recursive cycle to find a successful explanatory path in the SCM, starting with hypotheses causally linked to the observable symptoms. The cycle considers hypotheses in turn in order to explain the current question. Any associated tests that may confirm or deny the hypothesis under consideration are measured. Then, if the hypothesis is still viable, the cycle runs recursively until a terminating cause is reached. If no questions remain, the current path represents a diagnosis of the failure. When a hypothesis is falsified or leads to an inconsistent explanation, backtracking allows the exploration of alternative choices for earlier parts of the explanation.

An efficient SCM has branches for each failure of the device. We show one branch in Figure 2. A complete SCM is a large structure because each component and functional aggregation of components in the device must be named in specific hypotheses about potential failures. Likewise, a large number of tests are needed to rapidly prune branches during abduction. It is a tedious task to write complete SCMs for a large number of similar devices - a task we avoid by applying PE to a general causal model and to each circuit schematic.

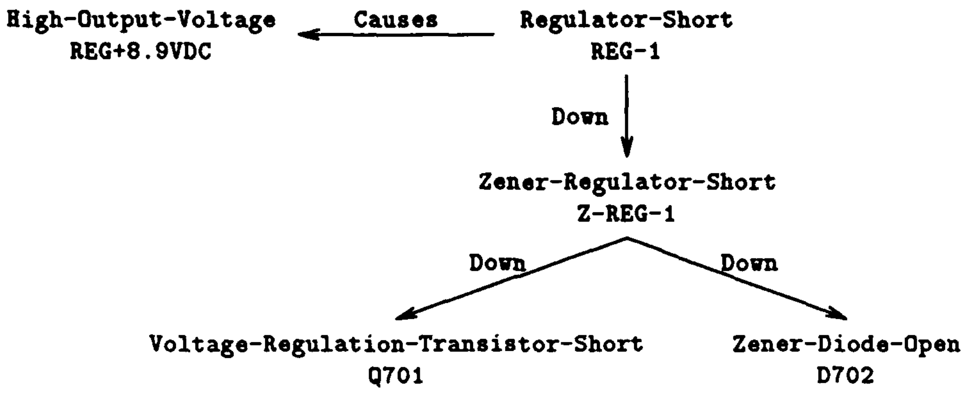

Figure 2: Specific causal model portion for a regulated power supply

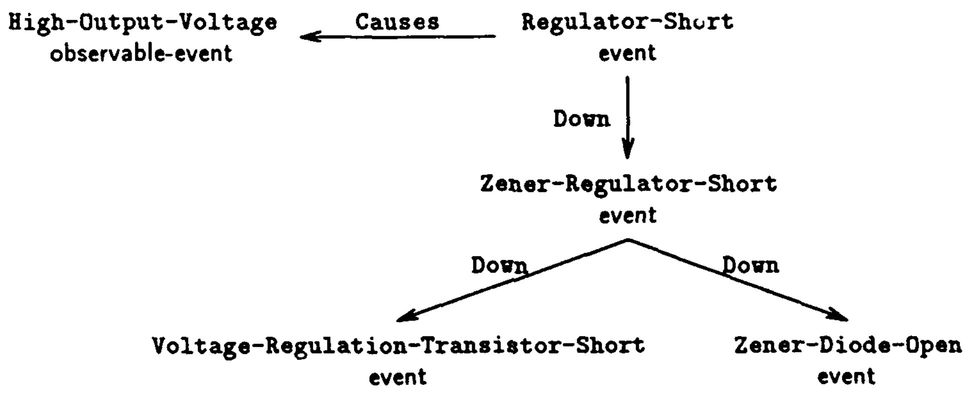

Figure 3: General causal model portion for power supplies

Partial Evaluation in LDS

Currently, LDS starts with the general causal model (GCM) and the annotated circuit schematic. Figure 3 shows the structure of a small portion of a GCM constructed for power supplies. This takes the form of a generalized SCM, with variables replacing component and functional block names. Associated with each node of the GCM is a *locator* pattern that PE uses in finding valid values for a node's variable when those values are matched against a given annotated circuit schematic. Figure 4 shows a locator pattern for a `High-Output-Voltage` presenting symptom. LOOM (MacGregor 1988), the knowledge representation language we use, provides support for this pattern matching.

A `High-Output-Voltage` observable event can happen for any X:

that is a `Power-Supply` output port,
that is not `Ground`, and
there is some `Voltage-Regulator`
with some `Part` on
the `Signal-Path` before X

Figure 4: `High-Output-Voltage` observable event locator pattern

Figure 5 shows an annotated circuit diagram, the eventual output of a structured-object recognition component, for the circuit shown in Figure 6. Figure 5 leaves out the connections for clarity; the actual data contains all the connections at the base component level and between higher level functional blocks. The annotated circuit schematic adds several layers of functional blocks to the component level represented in the original circuit diagram. Supplying the annotation by hand is straightforward, as long as the device is easily represented by block diagrams that characterize the functions that are subject to failure.

Generating the Specific Causal Model

Before the performance system is used, the specific causal model is initialized from GCM nodes representing presenting symptoms. For each match of the GCM node's locator pattern (such as the one in Figure 4) in the annotated circuit schematic, a new SCM node is created. In the example of Figures 5 and 6, there are multiple instances of voltage-regulated output ports, so for each such instance, an SCM instance of `High-Output-Voltage` for that port is generated. The measurement associated with each new node is instantiated with the corresponding nominal voltage from the annotated circuit diagram. The remainder of the SCM generation process takes place during performance.

At run time, missing SCM branches are generated, and the measurement values furnished by a technician or automated test equipment determine the path towards a final diagnosis - a path from the observable symptom to a terminating node. The first time a user selects a presenting symptom, abduction diagnosis asks for all the causal links to that symptom, triggering the learning system to go deeper into the GCM to create SCM nodes

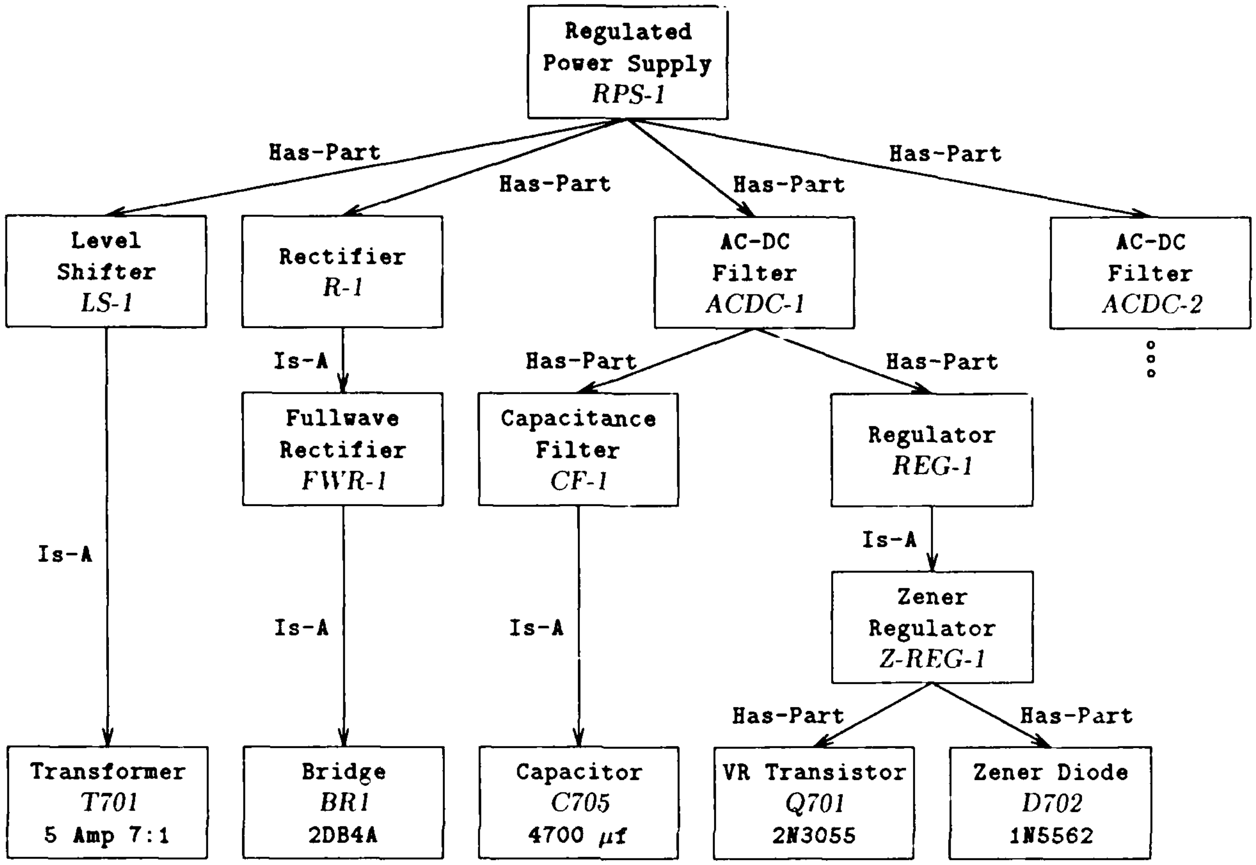

Figure 5: Annotated circuit schematic for a regulated power supply

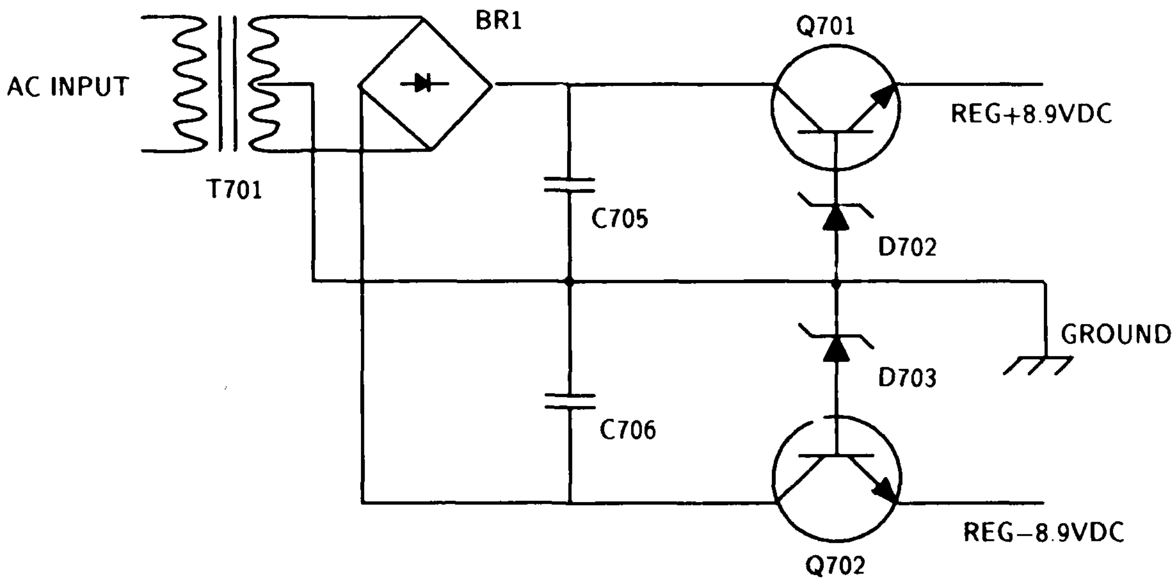

Figure 6: Circuit schematic for a regulated power supply

A `Regulator-Short` event can happen for any X:

where X is a `voltage-regulator`
with some `Part` on
the `Signal-Path` preceding
the object of the `High-Output-Voltage` event

Figure 7: Regulator short locator pattern

for each causal branch leading to that presenting symptom. Figure 3 shows the portion of the GCM in which one cause of the observed symptom `High-Output-Voltage` is found to be a `Regulator-Short`. The locator pattern stored at that node of the GCM is shown in Figure 7. Once again, this pattern is used to search the annotated circuit schematic for matches. Each voltage regulator that matches the pattern has its own SCM node created. Proceeding in this way, the structure of the GCM is imitated to create the SCM, including measurements that confirm or deny causal hypotheses. The constructed portion of the SCM is stored for future use.

4 Benefits of Partial Evaluation

Benefits of PE accrue in three ways.

1. PE exhibits the greatest efficiency gain when the final subset of problems solved by the limited model is relatively small compared to the scope of the original domain model. This is the case when the constructed knowledge will be much more specialized than the general knowledge.
2. The gain in problem-solving efficiency occurs for each problem solvable by the constructed model. In tradeoff with the previous benefit, a larger subset of problems covered by the specialized knowledge yields more aggregate benefit.
3. When many specialized models are needed (that draw on similar, overlapping, general model knowledge), PE's reuse of the general model decreases the amount of expert input needed per specialized model. Any knowledge encoded in the general model so

as to allow PE to apply it to more than one specialized domain need only be entered once into the general model.

We illustrate these three benefits with our example in the domain of electrical power supply diagnosis. All power supplies that output direct current, given an alternating current input, have much in common. There are just a few fundamental designs that are varied to meet particular specifications for the basic functions of the power supply. Thus, it is possible to write a general domain model for power supply diagnosis. This general model is expensive to use directly for diagnosis because of the processing required to account for the generalized representation of circuit structures. PE uses the schematic for a particular power supply as auxiliary information in order to create an efficient, circuit-specific model from the general model.

The benefit of using PE and storing an efficient model for each power supply circuit is quite large. The three benefits of PE apply as follows:

1. The causal model specific to a particular design that is output by PE is much more specialized than the general model that covers all power supplies. This large amount of specialization indicates a great reduction in the amount of computation required during a diagnosis.
2. Each constructed model sees a lot of use because, typically, a large number of units are made for a particular power supply design, and power supplies are notoriously prone to frequent failures.
3. The large amount of reuse of the general knowledge for power supply diagnosis, multiplied by the large number of distinct designs, clearly shows the potential savings of using PE over separately implementing a diagnostic system for each design. For a new power supply, only those features that are not already represented in the general model require additional expert knowledge. Knowledge added to the general model accumulates to become available for other circuit designs with similar structural or functional features. Note that it is a relatively simple task to obtain a machine representation of a power supply schematic.

5 Summary

The use of Partial Evaluation in LDS allows the generation of multiple expert diagnosticians from a single, general causal model. The input of a circuit schematic in annotated form provides sufficient information for partial evaluation to generate the specific causal model to be used for diagnosis. The inclusion of structured-object recognition in our system will automate the total process of generating a diagnostician when a circuit schematic in a conventional form, such as that produced by a CAD system, is input.

Although formally equivalent to explanation-based generalization, our method works from general models towards specific solutions in PE, rather than generalizing. Since the GCM needs to be acquired only once (in theory) and the results are applicable to many devices, the knowledge engineering effort is greatly reduced when compared with conventional techniques for generating expert systems. PE eliminates the efficiency loss caused by the use of a general representation by constructing of useful branches of the SCM.

In addition to diagnostics, we speculate that text understanding and intelligent interface systems can benefit from using PE. In a paraphrasing or question-answering task, general knowledge would be specialized to explain the input text. In intelligent interfaces, an array of general interfacing methods would be specialized by PE to meet the needs of each system to be integrated.

We recognize that because our general model contains no statistical knowledge, the resulting expert is really a novice, albeit an "educated" one. Due to this lack, the performance system may explore unlikely paths before hitting on the correct one. When we incorporate Case-based Reasoning, the novice can gather its own statistics automatically and gain skill in diagnosing each specific device. We are at present engaged in this enterprise.

References

Dietterich, T., 1986, "Learning at the knowledge level," *Machine Learning*, Vol. 1, pp. 287–316.

Kahn, K. M., 1984, "Partial evaluation, programming methodology, and artificial intelligence," *AI Magazine*, Vol. 5, No. 1, Spring 1984.

MacGregor, R., 1988, "A deductive pattern matcher," *AAAI-88*, Saint Paul, MN, August 21-26, 1988.

Patil, R. 1981a, *Causal Representation of Patient Illness for Electrolyte and Acid-Base Diagnosis*, Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Patil, R., Szolovits, P., and Schwartz, W. B., 1981b, "Causal understanding of patient illness in medical diagnosis," *IJCAI-81*, Vancouver, B.C., August 1981.

van Harmelen, F., and Bundy, A., 1988, "Explanation-based generalization = partial evaluation," *Artificial Intelligence*, Vol. 36, No. 3, October 1988.