A Medical Reasoning Program that Improves with Experience

Phyllis Koton

MIT Laboratory for Computer Science
Clinical Decision Making Group
545 Technology Square, Room 371
Cambridge, MA 02139

ABSTRACT

A physician's problem-solving performance improves with experience. The performance of most medical expert systems does not. I have developed a diagnosis program for coronary disease that improves its performance by remembering and learning from cases that it has already solved. The program diagnoses commonly-seen problems efficiently by recalling similar, previous cases and adapting their solutions through simple modifications. When it lacks experience in solving a particular type of problem, the program resorts to reasoning from a physiological model, then remembers the solution for future use. The program can produce solutions identical to those derived by a model-based expert system for the same domain, but with an increase of two orders of magnitude in efficiency. The method described is independent of the particular domain and should be generally applicable.

Introduction

The accumulation of cases seen over a physician's career improves his day-to-day problem-solving ability [1]. Experiences with new patients augment and refine the physician's knowledge. Making generalizations about the patients a physician has seen lets him make predictions about future similar patients. Remembering how an unusual past case was resolved can help the next time a similar case is seen.

Unlike physicians, most expert medical reasoning systems do not improve with experience. These programs cannot modify or expand their knowledge as a result of their experience in solving problems. They rely on knowledge painstakingly compiled from human experts, a process that is time-consuming and labor-intensive. When faced with the same problem twice in succession, they work just as hard to solve the problem the second time. These limitations stem at least in part from the inability of current systems to recognize similarities between a new problem and a previously encountered problem.

If such a system were given the ability to remember and make generalizations about cases that it had solved, its performance could be enhanced in several ways. It would be able to solve common problems more efficiently by recognizing that it already knows the solution to certain types of problems. By reusing previous solutions some of the work involved in devising a solution *de novo* could be avoided. By remembering cases after it solves them, the system could continually increase the collection of problems that it knows how to solve. The system could also modify its knowledge by allowing the user to override the program's solution, and remembering the solution that the user preferred.

I have implemented such a system in a program called CASEY. CASEY is a learning program that is built around an existing model-based expert system for managing patients with heart failure (the Heart Failure program [4]).

The Heart Failure program takes as its input a list of feature/value pairs that describe the patient. There are over 100 measures defined for the Heart Failure model. A patient description typically consists of about 40 features. The description for a new patient presented to the system, Larry, is shown in Figure 1.

The Heart Failure program finds a causal explanation which relates patient findings to the physiological states that cause them. From the causal explanation, the Heart Failure program can deduce the diagnosis and recommend therapy for the patient. The Heart Failure program produces its solution by exhaustive examination of a causal physiological model of the cardiovascular system. The model contains about 140 states, for example "high left atrial pressure" and "aortic stenosis."

CASEY takes the same input as the Heart Failure program, and also produces a causal explanation as its output. However, CASEY does this differently. First, CASEY finds a case similar to the new patient in its case memory. CASEY then evaluates the significance of any differences between the new case and the retrieved case. During this phase the match can be invalidated if there are significant differences. If none of the differences invalidate the match, CASEY adapts the solution from the retrieved case to fit the new case. If a match is ruled out, or if no similar previous case is found, CASEY uses the Heart Failure program to produce a solution for the case *de novo*. The new case and its solution are stored in CASEY's memory for use in future problem solving.[1] Intuitively, CASEY *remembers* a previous causal explanation that explained a similar case, and then *confirms* that it explains the findings in the new case.

[1] The user has the option of rejecting CASEY's solution, in which case Heart Failure program is used to produce a causal explanation, which will be stored in memory.

```
(DEFPATIENT "Larry"
HISTORY
(age .  65)
(sex male)
(dyspnea on-exertion)
(orthopnea absent)
(chest-pain anginal)
(anginal within-hours unstable)
(syncope/near-syncope on-exertion)
(palpitations none)
(nausea/vomiting absent)
(cough absent)
(diaphoresis absent)
(hemoptysis absent)
(fatigue absent)
(therapies none)
LABORATORY-FINDINGS
(ekg lvh normal-sinus)
(cxr calcification)
(calcification mitral aortic-valve))
```

```
VITAL-SIGNS
(blood-pressure 138 80)
(heart-rate .  90)
(arrhythmia-monitoring normal)
(resp .  20)
(temp .  98.4)
PHYSICAL-EXAM
(appearance nad)
(mental-status conscious)
(jugular-pulse normal)
(pulse slow-rise)
(apex-impulse normal)
(parasternal-impulse normal)
(chest clear-to-auscultation-and-percussion)
(abdomen normal-exam)
(extremities normal-exam)
```

Figure 1: **Patient description for Larry**

Remembering a similar case

CASEY remembers cases it has seen by storing them in a self-organizing memory system [2]. When presented with a new problem, CASEY searches its memory for a similar case. It matches a new case against cases in its memory using all the features in the patient description. However, all features are not equally important in matching a new case to a previous case. Furthermore, the important features for matching may *vary* from case to case. For example, the cardiac rhythm might be important and the heart rate unimportant for one case, whereas for another case, the opposite may be true. Therefore, for each new case presented to the system, CASEY dynamically determines the important features for matching. *Important features* are defined as those that are evidence for states in the causal explanation of previous cases.

CASEY does not require that the new case's features be identical to a previous case's features in order to match. In real-world domains, several different pieces of evidence may have equivalent implications. For example, LV strain on EKG and LV enlargement on chest x-ray are both evidence for the same pathophysiological state, LV HYPERTROPHY, even though they represent different features in a patient description. CASEY can repair a causal explanation that includes the state LV HYPERTROPHY to fit a new patient whose description includes evidence of LV HYPERTROPHY, say from an EKG, even if the evidence in the previous case came from a different source, such as a chest x-ray. For matching, therefore, it is sufficient to have features in both cases that are evidence for the same states in the model. CASEY generalizes features in the new case to refer to the states for which they are evidence. These are referred to as *observable states.*

In the patient Larry described above, the abnormal features are dyspnea on exertion, unstable anginal chest pain within hours, syncope on exertion, high heart rate, slow rise pulse, LV hypertrophy on EKG, and mitral and aortic valve calcification.

The best match for Larry found in CASEY's memory is a patient named Cal. Cal was a 66 year old male with unstable anginal chest-pain and syncope/near-syncope on-exertion. Cal had no history of dyspnea, orthopnea, palpitations, nausea/vomiting, cough, diaphoresis, hemoptysis, or fatigue. He was on no therapies. His vital signs were: blood pressure 147/81, heart rate 78, respiration rate14 and temperature 98.3. Cal's physical exam revealed a patient in no apparent distress. He had a normal jugular-pulse and a normal pulse. Auscultation revealed a single s2 and a murmur of as. Cal's apex impulse was sustained, his parasternal impulse was normal, his chest was clear, and his abdomen and extremities were normal. Cal's EKG showed normal-sinus rhythm and lv-strain. His chest x-ray revealed LV cardiomegaly and mitral and aortic valve calcification .

Cal's causal explanation is shown in Figure 2. Larry has evidence for six out of the seven observable states in Cal's causal explanation.

CASEY will often retrieve more than one case that matches the new patient. Retrieved cases are ranked using an algorithm that considers the observable states and the total number of features that the new case and the retrieved case have in common. CASEY examines the retrieved case with the highest rank first. If this match is ruled out and there is another retrieved case with a close score (currently, within 10% of the highest score) that case is examined. This continues either until a match is accepted or there are no remaining high-scoring matches.

CASEY can fail to find a case in its memory that is similar to the new patient. This can happen if none of the features in the new case are evidence for states that have been used to explain the findings of previous patients. In intuitive terms, CASEY can recognize that it does not know how to solve a

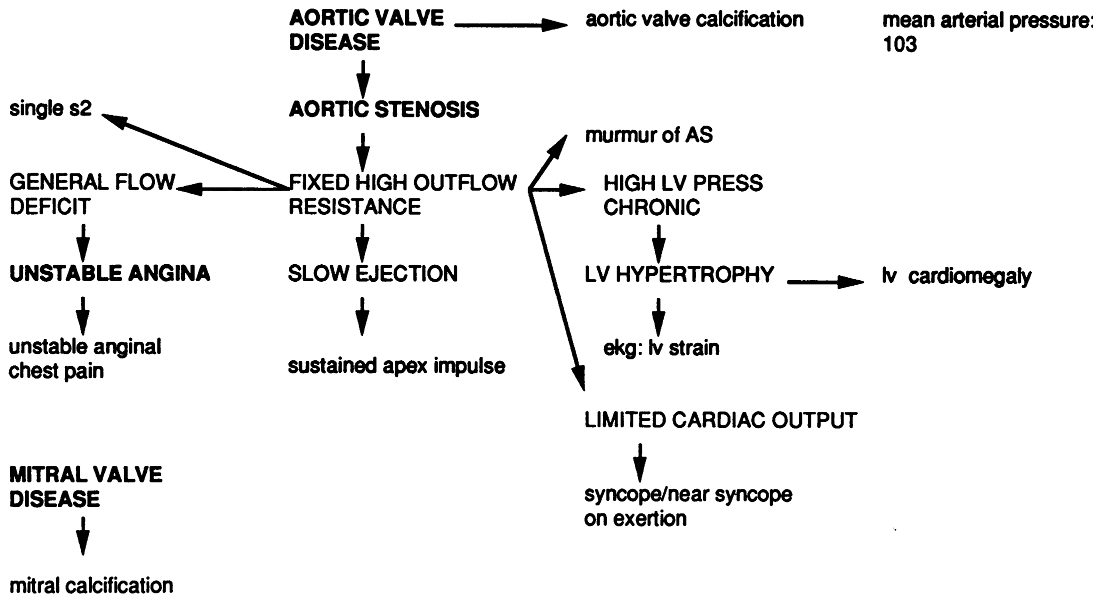

Figure 2: **Causal explanation for Cal.**

particular case. This is analogous to a physician encountering a patient with a constellation of symptoms the physician hasn't seen before. Just as a physician in this circumstance would consult his pathophysiology books, CASEY solves such a problem by invoking the Heart Failure program to find a solution for the new patient.

Reusing a previous solution

It is unlikely that any two patient descriptions will be identical, so matches between a previous case and the current case are usually only partial. Two cases may have many similar features yet have one critical difference that invalidates the match. A key question physicians answer is whether different constellations of findings still support the same diagnosis. Likewise, CASEY determines whether different features in the patient description still support the same causal explanation.[1] CASEY uses a set of *evidence principles* to evaluate differences between the new case and a retrieved case by examining the relationship between evidence and physiological states in the Heart Failure model. The evidence principles embody such concepts as alternate lines of evidence for states, additional supporting evidence for states, and inconsistent evidence.

The differences between patients Larry and Cal are shown in Figure 3. A difference is considered *insignificant* if it does not affect the retrieved causal explanation. CASEY judges the first five differences listed for Larry and Cal to be insignificant, because they are either very similar, such as the values for temperature, or because the feature is not involved in the casual explanation of the old case, such as heart rate.

Some differences cannot be ignored, but they can be explained using information in the Heart Failure program's causal model. A difference is said to be *repairable* if the features of the new case can be fit to the retrieved causal explanation. For example, dyspnea on exertion, present only in Larry, can be used as evidence for limited cardiac output, which is a state in Cal's causal explanation. CASEY can justify the match between Larry and Cal if it can explain all the differences between the two cases, and in fact, it can.

CASEY rejects a match on either of two grounds: a significant difference cannot be explained, or all the diagnosis states in the retrieved solution are ruled out. If the match between the new case and all retrieved cases are rejected, the Heart Failure program is invoked to provide a solution to the new case.

Modifications to the solution are necessary when there is a partial match between cases. *Repair strategies* are invoked by the justifier when it discovers a repairable difference between the new case and the retrieved case. Repair strategies adapt a previous solution (consisting of causal explanation, diagnosis, and therapy) to a new case. Causal explanation repair strategies add or remove nodes and links to a copy of the retrieved causal explanation. Diagnosis repairs add or remove diseases from the transferred diagnosis. Therapy repairs add or remove therapy suggestions. All three types of repair strategies are implemented in a similar manner.

The changes that must be made to Cal's causal explanation to fit the details of Larry's description are shown in Figure 4. The causal explanation produced by CASEY for Larry is shown in Figure 5. It is identical to the causal explanation produced *de novo* by the Heart Failure program.[2] CASEY obtained its solution by examining 578 states in the Heart Failure model, whereas the Heart Failure program examined about 7000 states.

[1]The differences can also affect the diagnosis and therapy, however these follow from the causal explanation.

[2]The user has the option of rejecting CASEY's solution and running the Heart Failure program on the data. In either case, the patient description and solution are saved to be used for future problem-solving.

Feature name	Value for Larry	Value for Cal
age	65	66
temperature	98.4	98.3
respirations	20	14
heart rate	90	78
blood pressure	138/80	147/81
ekg	LVH	LV strain
apex impulse	normal	sustained
pulse	slow-rise	normal
dyspnea	on-exertion	absent
s2	unknown	single
auscultation	normal	murmur of AS
chest x-ray	calcification	calcification LV cardiomegaly
angina	unstable within-hours	unstable

Figure 3: **Differences between patients Larry and Cal.**

Learning from a new case

CASEY stores the new case and its solution in its case memory so that it can be used for future problem solving. New cases are stored in the memory indexed both by the input features that describe the case and the solutions that were derived for the case (the causal explanation, diagnosis, and therapy suggestions).[1] CASEY also makes generalizations about the cases it has solved by finding similarities between the new case and cases already in its memory.

Generalizing the patient descriptions allows CASEY to make predictions about patients who share features [3] by recognizing co-occurrences. CASEY generalizes *all* the features in the patient description, not just the important features. The Heart Failure model is incomplete, so it is possible that certain features not deemed important are related to (and therefore can predict) some cause not represented in the model. The sex of the patient is an example: no state in the causal model uses the sex of the patient as evidence, yet there are causal relationships between gender and heart disease. The age of the patient is another example. The use of case-based reasoning therefore allows CASEY to *improve* on the performance of the Heart Failure system by learning new associations between features and solutions. At the same time, making generalizations about groups of similar patients reduces the effect of noise (random, unimportant features in the patient description) on the performance of the program. This is because spurious features are likely to occur randomly, whereas important features will tend to recur in cases presented to the program.

After producing the causal explanation for a new case, CASEY identifies the states in the causal explanation of the new case that are directly linked to findings, and indexes the new case in memory using these features. In the case of Larry, these states are limited cardiac output, unstable angina, slow ejection, lv hypertrophy, aortic valve disease, and mitral valve disease. Future cases that contain evidence to support these states will retrieve Larry's case as a match.

[1] This is true whether the solution was produced by CASEY or by the Heart Failure program.

Results

CASEY's performance was evaluated on two counts: *efficiency*, and *quality* of the solution. The program was tested on a set of 45 patients with symptoms of heart failure covering about 15 different diseases.

The quality of CASEY's solution was evaluated by comparing its explanation to the Heart Failure program's explanation for the same patient. A solution was considered *successful* if it was identical to the Heart Failure program's solution. A solution was considered *satisfactory* if it was identical to the Heart Failure program's solution except for the features which CASEY could not explain. In these latter cases, CASEY had already performed most of the task of deriving the causal explanation, and the Heart Failure program could be used to incrementally account for the remaining features. CASEY produced a solution that was either successful or satisfactory for 86% of the test cases for which there was a similar case in its memory. CASEY produced a solution identical to the Heart Failure program's solution in 14 out of the 45 test cases. It produced a satisfactory explanation for an additional 18 test cases. It gave up on six of the test cases, and produced an incorrect causal explanation for seven test cases. An examination of the test cases for which CASEY failed to reproduce even part of the Heart failure program's solution revealed that each one of these cases had a causal explanation that was completely different from any other patient in the memory. Even on these cases, CASEY could often produce part of the causal explanation, but could not account for the combination of features seen in the patient.

CASEY's efficiency was evaluated by comparing the number of states (of the Heart Failure program) it examined to the number states examined when the Heart Failure program solved the same problem. CASEY always examined fewer states than the Heart Failure program by at least an order of magnitude, and often by two or three orders of magnitude. Problems that can be solved quickly by the Heart Failure program have features which are specific to only one (or a small number) of states. Problems that require a lot of effort for the Heart Failure program are those with many symptoms that are evidence for a large number of states, which generate a large number of possible explanations that must be evaluated. By contrast, a simple case for CASEY is one in which there are few differences between the precedent and the new case. A difficult case for CASEY is one in which many differences between the precedent and the new case must be analyzed. A consequence of this difference is that as the number of cases solved by CASEY increases, it requires less effort to solve subsequent cases because it is more likely to find a close match. The Heart Failure program, conversely, cannot increase its efficiency except by re-implementation.

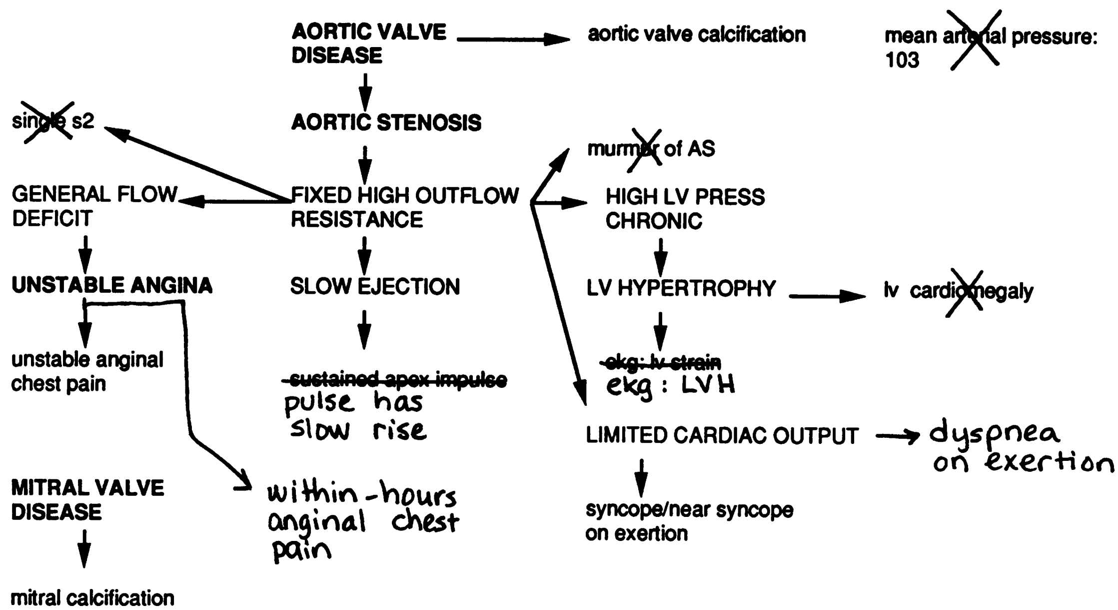

Figure 4: Modifications to Cal's causal explanation.

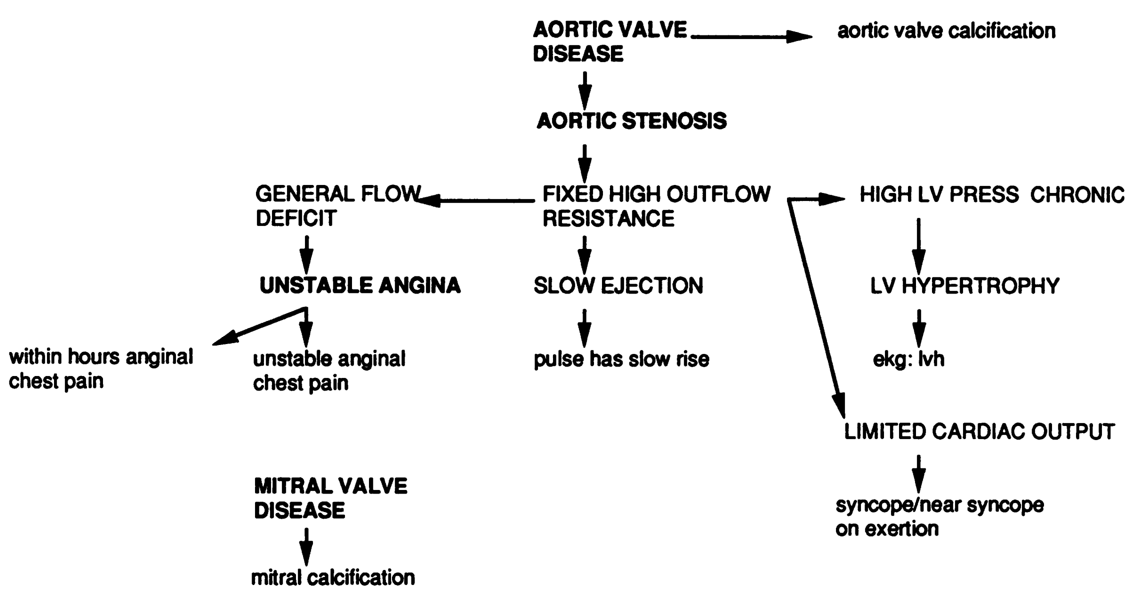

Figure 5: Causal explanation for Larry.

Discussion and conclusions

CASEY integrates model-based and case-based reasoning techniques in a program which is efficient, can learn from its experiences, and solves commonly-seen problems quickly, while maintaining the ability to reason using a detailed knowledge of the domain when necessary. The program is efficient because it makes small local changes to an existing solution. It can learn new associations and compile detailed reasoning structures into simple associations between features and solutions.

Since determination of important features is based on information in the model, it is reasonable to ask why the Heart Failure model is not simply "compiled" to produce all this information in the form of associational rules relating important symptoms and physiological states. In fact, that is exactly what CASEY is doing, but it is compiling the knowledge incrementally, associating features of problems with solutions for the cases it has seen. Also, although the Heart Failure program produces solutions involving multiple diagnoses, its model provides the relative importance of features only for single diagnoses. To compile all of the Heart Failure program's knowledge taking into account multiple diagnoses would be computationally intractable. Because CASEY also makes generalizations about patients who have multiple diagnoses, it can create associational knowledge relating features to solutions involving multiple diseases.

CASEY's current implementation has limitations. Some problems presented to the system have a large number of "reasonable" explanations. CASEY does not use all the quantitative information available in the Heart Failure model that would allow it to distinguish between statistically more- and less-likely solutions. For example, the program is parsimonious about adding additional states to the causal explanation. If a new feature could be attributed to two different physiological states, one of which is already included in the transferred explanation, CASEY will use the state that is already there rather than add a new state. It is possible that a feature has a higher probability of being caused by the state not already in the explanation. The Heart Failure model contains information that CASEY could use to discover this circumstance.

CASEY works by modifying one particular solution, rather than by generating a solution. This makes it difficult to evaluate the likelihood of its solution being correct compared to other possible explanations for the same data. The Heart Failure program, on the other hand, calculates the probabilities of all possible causal explanations that fit the data, and chooses the one with the highest probability. Again, the model contains information that could be used to roughly determine whether or not CASEY is pursuing the most likely explanation for the data.

When it has seen a relatively small number of cases, CASEY can introduce biases in the importance weights of features, for obvious reasons. If CASEY was given a large number of cases (for example, the same number as were used to develop the statistics used in the Heart Failure program), it would overcome the biases in its importance weights that were due to the small sample size of patients that it has reasoned about.

Acknowledgments

Robert Jayes, MD kindly provided the example cases. William Long's Heart Failure program provided an excellent resource for this work. Peter Szolovits, Ramesh Patil, and William Long gave much helpful advice in supervising this research. This paper was greatly improved by comments from Benjamin Neel, MD, PhD.

The work reported here has been supported in part by National Institutes of Health grants R01 LM 04493 from the National Library of Medicine and R01 HL 33041 from the National Heart, Lung, and Blood Institute.

References

[1] Paul J. Feltovich, Paul E. Johnson, James H. Moller, and David B. Swanson. LCS: The role and development of medical knowledge in diagnostic expertise. In William J. Clancey and Edward H. Shortliffe, editors, *Readings in Medical Artificial Intelligence: The First Decade*. Addison Wesley, 1984.

[2] Janet L. Kolodner. Maintaining organization in a dymanic long-term memory. *Cognitive Science*, 7:243–280, 1983.

[3] Janet L. Kolodner. Experiential processes in natural problem solving. Technical Report GIT-ICS-85/22, School of Information and Computer Science, Georgia Institute of Technology, 1985.

[4] William J. Long, Shapur Naimi, M. G. Criscitiello, and Robert Jayes. The development and use of a causal model for reasoning about heart failure. In *Symposium on Computer Applications in Medical Care*, pages 30–36. IEEE, November 1987.

[5] Roger C. Schank, G. C. Collins, and Larry Hunter. Transcending inductive category formation in learning. *Behavioral and Brain Sciences*, 9:639–686, 1986.

[6] Reid Simmons and Randall Davis. Generate, test, and debug: Combining associational rules and causal models. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1071–1078, 1987.

[7] Patrick H. Winston. Learning new principles from precedents and exercises: The details. AIM 632, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, 1981.