Learning New Principles From Precedents and Exercises*

Patrick H. Winston

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Recommended by Saul Amarel

ABSTRACT

Much learning is done by way of studying precedents and exercises. A teacher supplies a story, gives a problem, and expects a student both to solve a problem and to discover a principle. The student must find the correspondence between the story and the problem, apply the knowledge in the story to solve the problem, generalize to form a principle, and index the principle so that it can be retrieved when appropriate. This sort of learning pervades Management, Political Science, Economics, Law, and Medicine as well as the development of common-sense knowledge about life in general.

This paper presents a theory of how it is possible to learn by precedents and exercises and describes an implemented system that exploits the theory. The theory holds that causal relations identify the regularities that can be exploited from past experience, given a satisfactory representation for situations. The representation used stresses actors and objects which are taken from English-like input and arranged into a kind of semantic network. Principles emerge in the form of production rules which are expressed in the same way situations are.

1. Precedents and Exercises

This paper presents a theory of how it is possible to learn principles from precedents and exercises. The theory holds that causal relations identify the regularities that can be exploited from past experience in Management, Political Science, Economics, Medicine, and Law as well as from everyday life.

The paper begins with a presentation of some examples that illustrate the sort of learning to be explained. Next, there is a review of a representation and some matching issues developed in a previous paper which concentrated on

*This research was done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

Artificial Intelligence **19** (1982) 321–350

establishing two-situation correspondences [21]. And finally there is a description of the principle forming and indexing ideas that enable principles to be created and stored. There are references throughout to an implemented system that actually does reasoning, principle forming, and indexing using precedents and exercises.

The implemented system has a number of key ingredients, the following in particular.

- *Analogy-based reasoning.* Analogy is based on the assumption that if two situations are similar in some respects, then they must be similar in other respects as well. Here the causal structure of the precedent situation is assumed to map onto the exercise situation.
- *Learned if-then rules.* In contrast to current practice in Knowledge Engineering, if-then rules emerge as problems are solved. Teachers supply precedents and exercises, leaving the work of formulating the if-then rules to the system.
- *Actor-oriented indexing.* Context is established by the classes of the actors involved in problem situations. This is effected by storing the learned if-then rules according to the classes of the actors that are involved.
- *Mixed induction and deduction.* Under certain circumstances a relation may be assumed if it both explains the presence of some existing relations and allows a new relation to be deduced.

In addition, the implemented system inherits some key ingredients from previous work.

- *Actor–object representation.* Situations are represented using relations between pairs of situation parts. Supplementary descriptions can be attached to the relations when elaboration is needed.
- *Importance-dominated matching.* The similarity between two situations is measured by finding the best possible match according to what is important in the situations. Importance is determined by causal connections in the situations themselves. Causal connection is viewed as a common importance-determining constraint.

The system works in domains where the situations satisfy certain criteria: The first is that each situation can be described by using a vocabulary of a few hundred properties, classes, acts, and other relations; the second is that one situation is similar to another if the important relations in their descriptions can be placed in correspondence; the third is that the important relations of each situation are the ones explicitly said to be important by some teacher or implicitly known to be important by being involved in cause-effect relations; and the fourth is that the causes in similar situations generally lead to similar effects. The system will not work in domains such as vision and speech where different descriptive apparatus is necessary and where different regularities emerge.

2. What is to Be Understood

One way to determine if work in Artificial Intelligence has been successful is to use these criteria: first, there must be an implemented program capable of performing specified tasks; and second, the program must get its power from the knowledge embodied in an identified set of regularities or constraints. Thus, for there to be success by these criteria, there must be both *competence* (expressed knowledge) and *performance* (a demonstration that competence has been put to use).

Let us begin, then, by specifying some tasks to be performed. Consider the following precis of *Macbeth*, given by a teacher as a precedent.

> In MA there is Macbeth, Lady-Macbeth, Duncan and Macduff. Macbeth is an evil noble. Lady-Macbeth is a greedy ambitious woman. Duncan is a king. Macduff is a loyal nobel. Macbeth is evil because Macbeth is weak and because Macbeth married Lady-Macbeth and because Lady-Macbeth is greedy. Lady-Macbeth persuades Macbeth to want to be king. Lady-Macbeth influenced Macbeth because Lady-Macbeth is greedy and because Macbeth married Lady-Macbeth. Macbeth murders Duncan with Lady-Macbeth using a knife because Macbeth wants to be king and because Macbeth is evil. Lady-Macbeth kills Lady-Macbeth. Macduff is angry. Macduff kills Macbeth because Macbeth murdered Duncan and because Macduff is loyal.

Next, consider the following exercise.

> In EXERCISE-MA-1 there is a man and a woman. The man married the woman. The man is weak and the woman is greedy. Show that the man in EXERCISE-MA-1 may be evil.

Told by a teacher that *Macbeth* is to be considered a precedent, a student should establish the correspondence between the man and Macbeth and between the woman and Lady-Macbeth. Next the student should announce that the precedent suggests that the man is evil in the sense that he is likely to commit evil acts. Then the student should form a principle-capturing if-then rule suggesting that the weakness of a man and the greed of his wife can cause the man to be evil. The rule should be accessible whenever it is desired to see if there is reason to think that a man is evil.

Another situation is described and another question is posed in this second example.

> In EXERCISE-MA-2 there is a nobel a lady a prince and a king. The nobel married the lady. The noble is evil and the lady is

greedy and the prince is loyal. Show that the prince in EXERCISE-MA-2 may kill the noble.

Responding, the student should establish appropriate correspondences as before. Then the student should announce that the prince may kill the noble, given *Macbeth* is a precedent. Then the student should create a rule relating the evil of a noble and the greed of his wife to the noble's death at the hands of a loyal prince. The rule should be indexed so as to be accessible whenever it is desired to show that a prince kills a noble.

There is an implemented program that behaves as the student does in these representative examples, doing problem solving by analogy, rule formation, and rule indexing and retrieving. Thus a specified task is performed by the program, satisfying the first criterion for success. The program works because causal connections between relations in past situations tend to show up in future situations. Thus the program works because it exploits an identifiable regularity, satisfying the second criterion for success.

So far, the program has learned about two dozen simple rules using toy-like precedents and exercises from *Macbeth* World, Medicine World, and Politics World.

3. Representing and Matching Situations

I define a representation to be a vocabulary of symbols together with some conventions for arranging them to form descriptions of objects in a domain. Some desirable attributes for a representation are that it makes the important facts explicit, it suppresses irrelevant detail, it exposes constraint, and it can be reached by an existing translator working from a natural input.

3.1. Actor–object representation

The particular representation used in this work is based on two key assumptions: first, that what needs to be represented can be expressed in simple English; and second, that much of what can be expressed in simple English can be described using a representation that has slots for both a relation and the things that the relation ties together.

For the simplest sentences, the things involved are just the actor and the object, and the relation is an act. For others, the actor and object are supplemented by other things that participate in act description, such as an instrument, a time or location, or perhaps a source or destination if the act involves motion. In the following, for example, an instrument is specified.

Macbeth murders Duncan with a knife.

Knowing the kinds of things that are involved in describing an act and understanding how to recognize those things in sentences is the objective of *case-grammar theories* of sentence meaning [6]. Actors, objects, instruments, and similar terms are used as names for *case slots*. Sentence analysis is viewed as the job of filling case slots using sentences as raw material.

In this work, the actors and objects in situations are represented graphically as nodes that are tied together with acts and other relations forming a kind of semantic net. When more than an actor and an object is involved in an act, a supplementary description is tied to the act-specifying relation, making the representation capable of bearing case-like information. The supplementary description itself consists of a node related to other nodes as illustrated by Fig. 1.

Information about properties and classes is easily conveyed by using A-KIND-OF and HAS-PROPERTY relations. Supplementary description nodes can be attached to A-KIND-OF, HAS-PROPERTY, and other relations as well as to acts.

Section A.1 (in Appendix A at the end of this paper) goes into more detail about the implementation of the representation.

3.2. English-like input and simple deductions

If descriptions are produced automatically from a natural input, then there is less danger of doing all the real work before the computer actually gets the data. Thus descriptions in the actor-object representation should be computable from simple English descriptions or from something that is at least close to simple English. Moreover, a translator from English-like descriptions to actor-object descriptions makes experimental work practicable.

In fact, two translators have been designed and implemented: a simple one, easily modified by me, for current use and a more powerful one, less easily modified by me, for future application. The simple one is based on LIFER, but lacks LIFER's pronoun-handling features [10]. The other one is the work of Katz [11]. The *Macbeth* example was given in the form that the simple translator handles. In Section A.2 has been illustrated what Katz's translator can do.

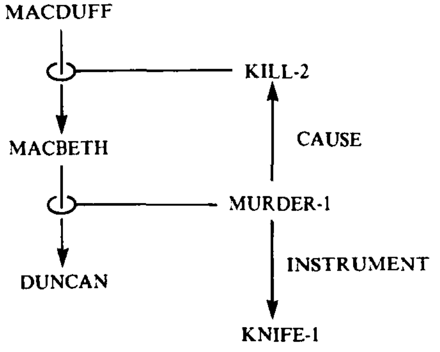

FIG. 1. Supplementary description nodes MURDER-1 and KILL-2 carry case-like information supplementing the actor and object.

The use of natural input also requires a way to make some simple deductions as sentences are read, reducing the need for tedious attention to details. For example, given that Macbeth marries Lady-Macbeth, Lady-Macbeth clearly marries Macbeth. Similarly since Macbeth kills Duncan, it is clear that Duncan is dead. Such obvious deductions are done in the implementation by a few demons that are invoked when relations are added into the data base.

3.3. Matching establishes part correspondence

Analogy is based on the assumption that if two situations are similar in some respects, then they must be similar in other respects as well. To determine if two situations are similar, the parts of the situations must be placed in correspondence. The purpose of matching is to establish the best way to do this.

The performance of the current matcher [2], is described in Section A.3. Two characteristics of the matcher are as follows.

- The matcher has a function by which it judges the similarity between two parts. The similarity function uses properties, classes, acts, and other relations, depending on what is important. (A previous paper gives more detail on the similarity function [21].)
- The matcher searches the space of all possible matches heuristically, beginning by linking up the most similar parts from the two situations and then moving on to those that are less similar. The parts that are linked supply helpful evidence for linking up those that are not.

At the moment, there are two ways to determine what is important. One way is to attribute importance to all relations; the other is to attribute importance only those relations that are the antecedent or consequent of a causal relation. Section A.4 describes the various words implying cause and shows how they are interpreted.

The reason that relations connected by CAUSE are important is that they are, ipso facto, the ones involved in a constraint that may transfer from a precedent to a new situation.

3.4. Cause can bias matching toward properties, classes, or relations

In her excellent papers on analogy, Gentner observes that properties and classes seem to influence correspondence more than relations when people are beginners in a problem domain [8]. Later on, there is a reversal: relations have more influence than properties and classes.

These observations may have to do with cause-determined importance. Beginners in a domain have less knowledge about what causes what than experts do, so it is hard to give special attention to any particular properties, classes, and relations when matching. As expert-level performance is reached, more is known about which properties, classes, and relations often are related by cause and effect, making it possible let those things weigh heavily in matching. Evidently, relations are more often involved in cause and effect than properties and classes, explaining the experts' dependence on relations.

3.5. Cause can bias matching toward desired use

Introspection tells us that the way things are matched depends on purpose as well as experience. The same precedent may match some exercise in more than one way depending on what is to be shown or explained. It is easy to make the existing matcher behave this way.

As it stands, the mechanism that marks relations as important does so whenever they appear in a causal chain. A bias-introducing mechanism could mark relations only when they appear in a causal chain that leads to some specified relation. With this, it is possible to construct an example whereby matches with *Macbeth* go different ways depending on whether the interest is in why Macbeth is evil or in why Macduff is loyal.

4. Reasoning and Creating Rules Using Analogy

How is it possible to know if some relation holds in an exercise, given that the exercise is analogous to another, well-understood precedent? The answer is that the constraint relations in the precedent suggest the right relations to check and the right questions to ask. Here we consider only constraint involving causal connections. The key assumption is that the causal structure of a precedent is likely to say something about the possible causal structure in an exercise to be analyzed.

4.1. Causal relations enable common-sense reasoning

Consider the *Macbeth* precedent, given earlier, together with the following exercise.

> In EXERCISE-MA-1 there is a man and a woman. The man married the woman. The man is weak and the woman is greedy. Show that the man in EXERCISE-MA-1 may be evil.

In the description of *Macbeth*, Macbeth is said to be weak and his wife, Lady Macbeth, greedy. The matching program uses these facts, together with sex classes, to claim correspondence between Macbeth and the man in the exercise and between Lady Macbeth and the woman. The matcher finds no people in the exercise corresponding to Duncan or to Macduff.

While the man in the exercise is not known to be evil already, Macbeth is. Moreover, Macbeth is said to be evil for reasons that hold for the man: both are weak and both are married to someone who is greedy. Thus insofar as *Macbeth* is a precedent for the exercise, it seems reasonable for the common reasons to lead to a common effect: the man may be evil.

In this simple illustration, there is only one precedent involved and there is only one level of CAUSE relations in that precedent. Naturally the implemented system handles more general situations. To be more precise, the following steps are taken when the user asks about some relation in an exercise given a precedent.

Step 1. The relation in question may actually be in the exercise. If so, no further action is needed.

Step 2. The relation in question may be caused by other relations in the precedent. If so, try to establish the other relations in the exercise.

Step 3. The relation in question may be both absent from the exercise and causeless in the precedent. If so, seek another precedent-establishing precedent to work with.

Thus the causal structure of the precedents serve as a template, guiding the student to the possible antecedents of the relation in question.

Examples further illustrate how these rules work. In Fig. 2, assume that there is enough background information, in the form of properties, classes, and other relations not shown, to establish that P-A-1 (for precedent agent number 1) matches best with E-A-1 (for evidence agent number 1), P-O-1 (for precedent object number 1) with E-O-1, and so on. Given this, R1 is justified between E-A-1 and E-O-1 as follows: first, R1 in the precedent is caused by R2 between P-A-2 and P-O-2; and second, R2 is in place between E-A-2 and E-O-2 in the exercise.

The example of Fig. 3 is more difficult only because the causal chain must be chased a little further. R2 in the exercise is at issue. By going back two steps in the precedent's causal chain from R2, R4 is encountered, which is in place in the exercise. Other relations that come after R2 and before R4 in the chain are ignored.

Finally, the example of Fig. 4 goes a little further still in that two precedents are needed. Moving back through the causes in the first precedent, P, relation R3 is encountered, but it is not in place in the exercise, nor is it caused by another relation. To establish R3, it is necessary to use another precedent, Q,

FIG. 2. An example illustrating reasoning by analogy. Relation R1 may be justified in the exercise by reason of the causal structure of the precedent and the known relation, R2, in the exercise. Unlabeled relations are CAUSE.

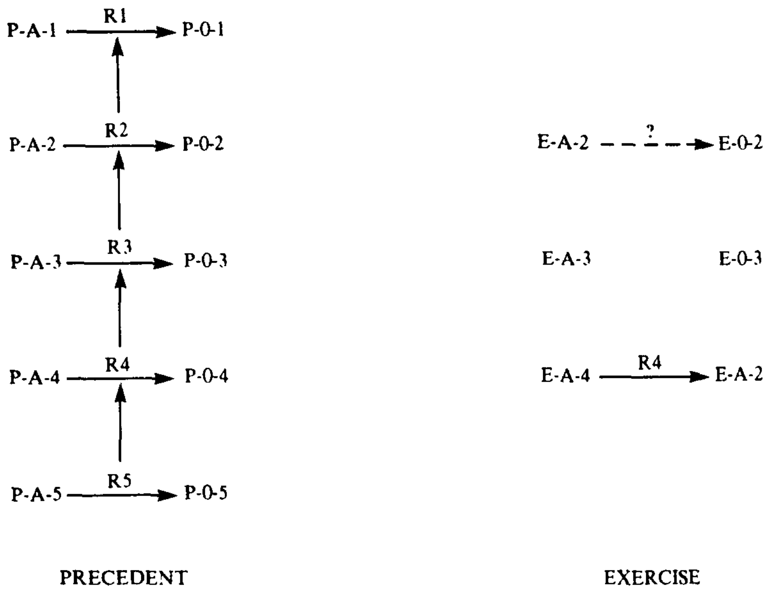

FIG. 3. An example illustrating reasoning by analogy with a two-step causal chain. Relation R2 may be justified in the exercise because the causal chain of the precedent leads to relation R4, which is in a corresponding place in both the precedent and the exercise. Unlabeled relations are CAUSE.

with another causal chain that can be exploited. Here the causal chain of the other precedent leads from R3 to R5, which is in place, thereby justifying both R3 and the original relation at issue, R2.

4.2. Reasoning may be for confirmation or prediction

When working with *Macbeth* and *Exercise-ma-1*, the precedent was used to confirm that a certain relation held *at the time of analysis*. The action would be the same if the precedent were used to predict that a certain relation would hold *at some time in the future*. Indeed, the implemented programs cannot know the difference since the representation used has no provision for denoting past, present, or future.

There is some implicit sequence information, however, since the word *because* is always taken to imply that one relation contributes to the existence of another at a later moment. In unrestricted English, *because* is used ambiguously. In "Macbeth killed Duncan because Macbeth was evil", the meaning is that Macbeth's evil quality was one of the things that contributed to the murder; in "Macbeth was evil because Macbeth murdered Duncan", the

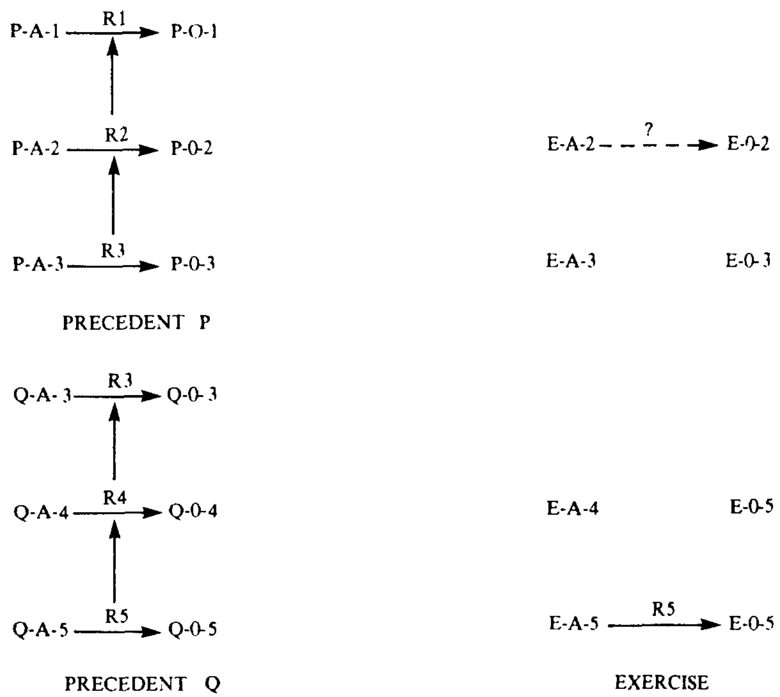

FIG. 4. An example illustrating reasoning by analogy with two precedents needed. Working on relation R2 in the exercise requires both precedent P and precedent Q. Unlabeled relations are CAUSE.

meaning is that Macbeth's murder of Duncan forces the conclusion that he must have been evil beforehand.

4.3. Common-sense reasoning generates rule-like principles

One way to explain how the analogy-oriented reasoning process works is to note that the precedent and the exercise determine an AND tree. The root is the relation to be shown in the exercise; the tips are relations that hold in the exercise; and the structure in between is supplied by the precedent.

In describing a precedent and constructing an exercise, the teacher causes attention to be drawn to a particular fragment of the causal structure exhibited in the precedent. One reason to do this is that the fragment involved may be worthy of special notice. Indeed, it may be that for some broad classes of actors and objects, the relations at the AND tree tips lead to the relation at the root often, not just in the precedent and the exercise.

Suppose that a new situation is formed from the relations at the tips and at the root of the AND tree. Further suppose that the parts involved are created by making instances of the parts in the exercise. In the simple example using *Macbeth* and *Exercise-ma-1*, the situation synthesized from the precedent and the exercise is illustrated in Fig. 5. Note that the synthesis procedure ties all the tips to the root directly with CAUSE relations.

- Situations synthesized from precedents and exercises constitute constraint-describing summaries that may be worth remembering as plausibility-indicating principles.

Displayed in another notation, these principles are plainly like the if-then rules that are so popular in Knowledge Engineering:

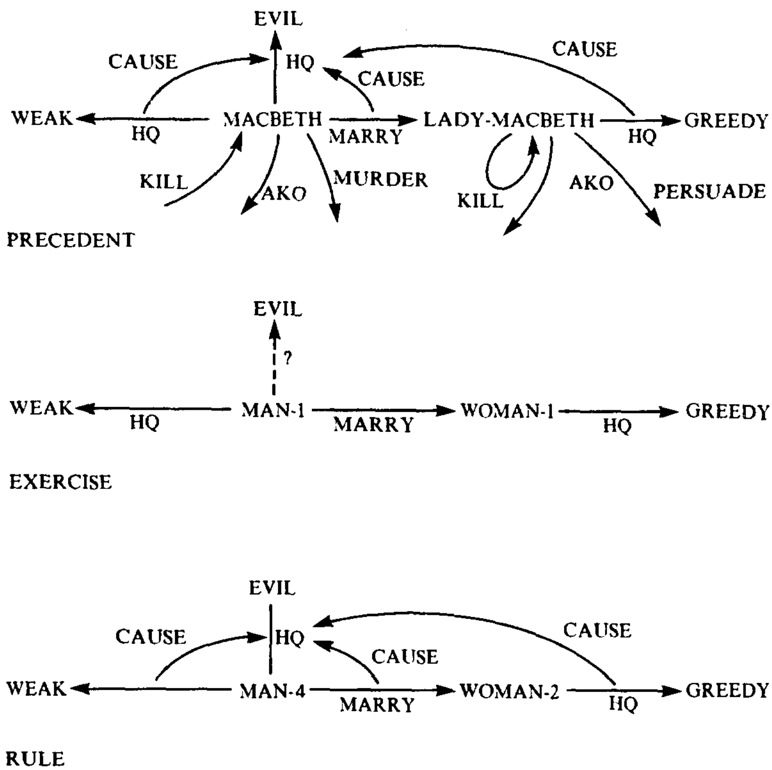

FIG. 5. An exercise causes part of the causal structure of the precedent to be extracted, forming a rule. The rule is in the same representation used for the precedent and the exercise. AKO = A-KIND-OF. HQ = HAS-QUALITY.

Rule
RULE-1
if
[MAN-4 HAS-QUALITY WEAK]
[MAN-4 MARRY WOMAN-2]
[WOMAN-2 HAS-QUALITY GREEDY]
then
[MAN-4 HAS-QUALITY EVIL]
case
MA

Since these plausibility-indicating rules are represented internally in the same way precedents are, they can be used by the same matching, reasoning, and learning machinery that works for ordinary precedents. Thus we will see examples in which old rules lead to new ones.

FIG. 6 shows how the synthesis procedure works when there is more depth to the AND tree.

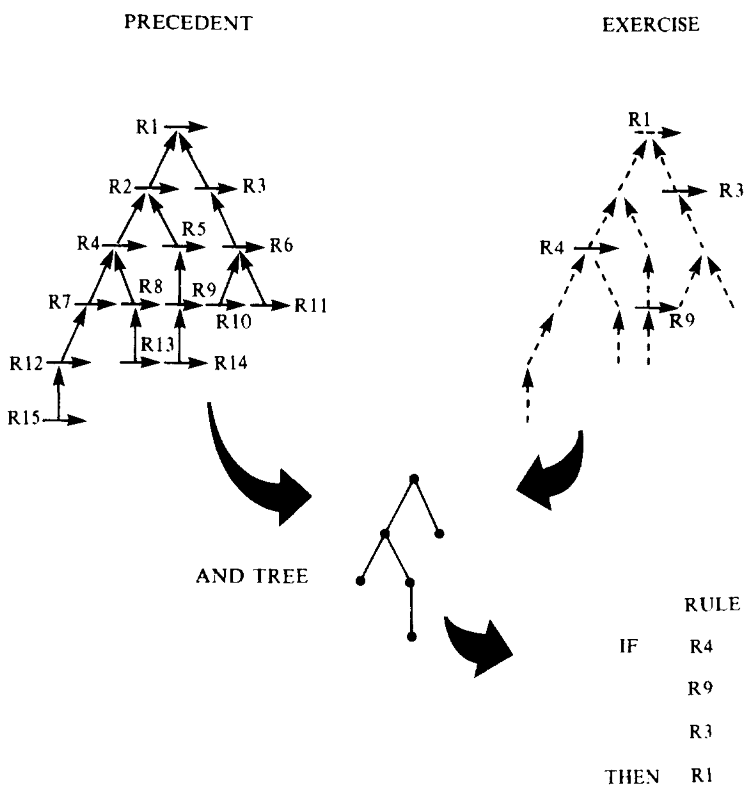

FIG. 6. The rule synthesis procedure works because the precedent and the exercise determine an AND tree, from which the rule is made. The tips go into the *if* part of the rule and the root into the *then* part.

4.4. Why have rules?

It is good to *have* rules for several reasons. First, the cause-effect knowledge in a rule is quite explicit, lacking the irrelevant clutter and intermediate reasoning steps of the precedents from which rules are derived. Explicit capture of the essentials is one of the prime desiderata of good representation.

Second, it is easier to match to rules than to match to their precedents. Rules have fewer parts and fewer properties, classes, and other relations. Matching a new situation against a rule is faster and less likely to be diverted from the most useful matching result.

Third, rules are the stuff of Knowledge Engineering, a part of Artificial Intelligence with established techniques for dealing with questions about why a certain conclusion was sought, about what facts were involved in supporting it, and about what knowledge was involved.

And finally, fourth, rules are easy to index and retrieve, as we shall see.

4.5. Why learn rules?

It is also good to *learn* rules. Again there are a number of reasons.

First, of course, rule learning is interesting for its own sake. What is it about the way the world is put together that enables us to capture, summarize, and exploit past experience? What are some particular processes for doing these things? Are there others? What about the way people do it? What about Medicine, Political Science, and Law?

Second, there is too much to do. We need systems that can learn rules from precedents because the traditional process of extracting rules from experts is too slow and too expensive.

Third, getting rules from experts may be unreliable because the existing experts may have difficulty verbalizing the rules they use, even though they may be able to teach the subject matter by citing the useful precedents.

And fourth, there may be no experts because a discipline is too new or too complicated. If successful research is done on the difficult problem of automatic precedent selection, then analogy-oriented learning systems could operate even when there are no human experts to help.

4.6. Why have precedents?

After learning a rule, it is good to keep its precedents because there may be a need to reason on many levels of detail. Existing expert systems have only knowledge distilled from the heads of those that contribute to them. The systems themselves have no way of asking about the origins of their own knowledge, a defect that can be fatal when a rule is misapplied.

In contrast, when the procedures described here derive a rule, the precedents used are noted. This means that the facts behind the rules can be examined when a rule seems to lead to a doubtful conclusion. Consider, for example, the following exercise, again to be used with *Macbeth*.

> In EXERCISE-MA-2 there is a noble a lady a prince and a king. The noble married the lady. The noble is evil and the lady is greedy and the prince is loyal. Show that the prince in EXERCISE-MA-2 may kill the noble.

Together with this exercise, *Macbeth* dictates the construction of the following rule:

```
Rule
      RULE-2
if
      LADY-6 HAS-QUALITY GREEDY]
      [NOBLE-8 MARRY LADY-6]
      [NOBLE-8 HAS-QUALITY EVIL]
      [PRINCE-4 HAS-QUALITY LOYAL]
then
      [PRINCE-4 KILL NOBLE-8]
case
      MA
```

Now consider the following exercise, which closely resembles the one just used to create a new rule.

> In EXERCISE-MA-3 there is a noble a lady and a prince. The noble married the lady. The noble is evil and the lady is greedy and the prince is loyal.

The difference between this exercise and the previous one is that there is no king. Nevertheless, analyzing the problem in the light of the rule just created, one finds that the match is good and that the analogy procedure is satisfied: it seems that the prince kills the noble. Here is the actual trace of the computation, with some cosmetic deletions.

```
Matching EXERCISE-MA-3 against RULE-2
41. values match with 48. and 75. possible.
Normalized score is 85. %
I note that [LADY-2 HAS-QUALITY GREEDY] for use with RULE-2
I note that [NOBLE-2 MARRY LADY-2] for use with RULE-2
I note that [NOBLE-2 HAS-QUALITY EVIL] for use with RULE-2
I note that [PRINCE-2 HAS-QUALITY LOYAL] for use with RULE-2
The evidence from RULE-2 indicates [PRINCE-2 KILL NOBLE-2]
```

There is a flaw, however. The exercise has no king for the evil noble to murder. Analyzing the problem using *Macbeth*, rather than the rule, has the following result:

```
Matching MA against EXERCISE-MA-3
42. values match with 184. and 52. possible.
Normalized score is 80. %
Match is decisive -- 10. better than next best.
I note that [PRINCE-2 HAS-QUALITY LOYAL] for use with MA

Beware! Use of match required creating KING-7 for DUNCAN

I note that [NOBLE-2 HAS-QUALITY EVIL] for use with MA
I note that [NOBLE-2 MARRY LADY-2] for use with MA
I note that [LADY-2 HAS-QUALITY GREEDY] for use with MA
The evidence from MA indicates [PRINCE-2 KILL NOBLE-2]
```

The line beginning with *beware* is the important one because it warns that some object in the exercise, possibly vital, is missing and must be conjectured. In this example, it can be argued that a really smart system would know there must be someone for the prince to be loyal to. The point, however, is that it is possible to follow up on doubted conclusions. The first step is to follow the trail from the doubted conclusion back through the rule involved to the governing precedent from which the rule was derived. The second step is to use the precedent for analysis rather than the rule.

```
The rule used for [PRINCE-2 KILL NOBLE-2] is RULE-2
The precedent for RULE-2 is MA
Matching MA against EXERCISE-MA-3
   :
Beware! Use of match required creating KING-7 for DUNCAN

   :
```

Another possibility is that some part of the causal chain in the precedent may be flawed, rarely likely, or inapplicable. Having the precedent handy means that these possibilities, in principle, could be pursued.

The key idea is that the precedents that support the rules can be examined in a variety of ways, on demand, unlike those of current Knowledge Engineering, which lie inside the human expert. The intermediate steps leading from causes to effects can be examined and questioned. It becomes possible to reason coherently at more than one level. The knowledge is more open, with less of a compiled flavor.

5. Indexing and Retrieving Rules

Systems with general intelligence will have to reason about all sorts of things, raising the question of how to cope when there are many domains to think about and when the combined number of rules gets large. Somehow things must be organized so that it is not necessary to consider the rules governing greed when dealing with, say, an unconscious diabetic.

A natural idea is that the rules must be grouped with context deciding which groups should be considered in any given situation. But this idea raises questions. What are the natural groups? Is it possible to fix them in advance? Is it possible to discover them while acquiring the rules? What determines context? How does context establish that a rule group is active?

5.1. Class determines storage and access

Recall again that the precedent involving Macbeth and the exercise involving a man resulted in a rule involving a man, inasmuch as the corresponding object in the exercise was a man. Thus the classes of the objects involved in rules is under the control of the teacher who supplies the exercises.

It seems reasonable to assume that a rule about a man may be useful in establishing something about man, a noble, or anything else that is a kind of man. It is not so reasonable to assume that a rule about a noble will be useful in establishing something about a man or anything else that a noble is a kind of. Thus we have the following principle.

- Rules whose conclusion involves some particular actor and object classes should be stored so that they may be accessed when those same classes are involved in a question.

With a view toward reasoning by backward chaining, one way to store rules is by sorting them into buckets keyed by the class of the actor in the conclusion, then by the act, and then by the class of the object. In the implementation, each rule ends up in a multilevel association list stored in the frame describing the actor's class. Fig. 7 illustrates.

For retrieving, there is the following complementary principle.

- Questions that involve some particular actor and object classes should cause access to rules whose conclusion involves actors and objects of those same classes.

In the implementation, the classes that the actor in the question is a kind of are examined starting with the most general and progressing to the most specific. All the rules associated with a given class are tested to see if the act is the same as that in the question and if the object class is something that the object in the question is a kind of. If all tests are satisfied for any rule, the rule is tried. For all those involving actors of the same class, the most recently created is tried first, on the weak ground that increasing knowledge makes rule formation more reliable.

A rule may fail, of course. The indexing and retrieving apparatus simply makes rule application contingent on two hurdles, rather than just one: first the rule must be found; and second the *if* part of the rule must be compatible with the current situation.

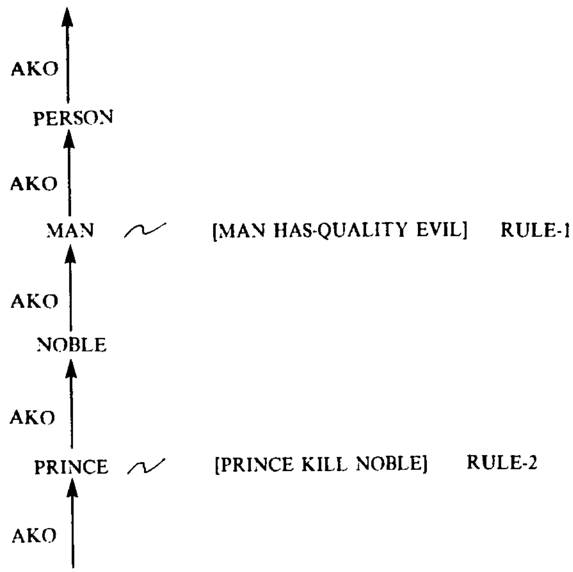

FIG. 7. A rule is stored according to the class of the actor involved in the then part of the rule. Further indexing is done using the relation and the class of the object.

5.2. Class establishes context

Part classes determine context since part classes determine what rules can be tried. Rules about nobles are for situations involving nobles, and rules about diabetics for situations involving diabetics. The same person may be described in one situation as a noble and in another as a diabetic, with the sensible context established and the sensible rules tried automatically.

5.3. Storage and access needs are suggestive

Some interesting ideas leap to mind as a consequence of the indexing and retrieving mechanisms. It is to be understood that these ideas are particularly speculative in view of the small size of the experiments done so far.

Top-down accessing. The argument for top-down access is that the rules with the most general actor classes involved will be the most reliable.

Rules as constraints on hierarchy. Little is known about what classes there should be and about when and how new classes should be formed. It can be that thinking about rule indexing and retrieving can help us find the determining constraints. For one thing, class refinement may reduce the number of times a rule is found but cannot be used. It may be easy to push such an overtried rule into a subclass or to create a new subclass for it by noting exactly how the rule fails. New subclass creation would be natural as one becomes an expert in a domain, learning many new rules of increasing specificity.

Rules as intermediates. It is hard to get rules out of some experts, raising doubt about whether they really use them at all. Some doctors, for example, seem to dislike having their knowledge put into rules. These doctors talk of having a feel for a case that seems to involve more than those characteristics that would be in causal chains and hence in rules.

One argument is that people use rules, but have difficulty articulating them. Another is that people do not use rules at all, but somehow work with partly remembered, partly forgotten cases from school and practice.

There is still another possibility, however. Boris Katz notes that a rule may just miss applying while its precedent applies perfectly (conversation). He observes that it makes heuristic sense to check the precedent when a rule's *then* part and many of its *if* parts match the current problem. Katz has suggested the term *near match* for these situations. Indeed, it may be that the rules are not used as 'rules', but rather as 'abstracts'. After all, rules summarize a way in which a precedent can be used. If an exercise matches most the if-then parts of a rule, then the precedent from which the rule was derived may be a good precedent. Earlier discussion suggested that it is always possible to go back to a precedent if an application of a derived rule gives a suspicious result. Here the suggestion is that going back may be the normal way, using the rule-like summary only to do indexing and retrieving.

Establishing classes using relations. In a sense, using classes to establish context just pushes the context problem back a bit, to the question of how classes are established. Of course classes often are given in a problem statement. Otherwise they may be deduced by rules from existing relations: sons of nobles are nobles; and people whose pancreases are bad are diabetics. Of course current context determines which rules are available for establishing classes using relations, thus suggesting the loop illustrated in Fig. 8.

Note, incidentally, that the problem solver is influenced by whatever information it has about class. It can be biased toward silly conclusions by being

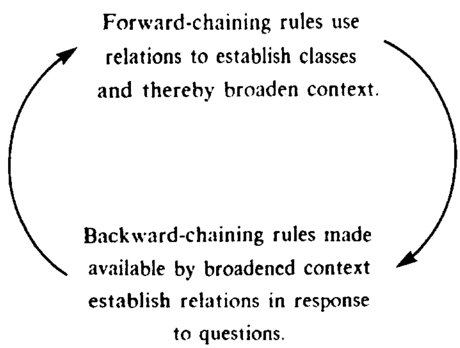

FIG 8. Class, context, and relations interact.

told inappropriate or wrong class identifications, just as humans are prejudiced by being told a person is a slob, a hero, or socially in or out.

6. Macbeth, Politics, and Diabetes

About two dozen rules have been generated. Some are discussed in this section to illustrate additional points. Keep in mind that the performance of the system has to be judged in the light of the knowledge in the given precedents. A complete trace is available [22].

6.1. Generating rules from other rules—Macbeth world

In this experiment two rules are used to produce another. One is RULE-1, seen before.

```
Rule
        RULE-1
if
        [MAN-4 HAS-QUALITY WEAK]
        [MAN-4 MARRY WOMAN-2]
        [WOMAN-2 HAS-QUALITY GREEDY]
then
        [MAN-4 HAS-QUALITY EVIL]
case
        MA
```

Another is derived from the following exercise.

> In EXERCISE-MA-4 there is a man. The man is evil. The man wants to be king. In EXERCISE-MA-4 there is a king. Show that the man in EXERCISE-MA-4 may murder the king.

Which produces the following.

```
Rule
        RULE-3
if
        [MAN-5 WANT [MAN-5 AKO KING]]
        [MAN-5 HAS-QUALITY EVIL]
then
        [MAN-5 MURDER KING-8]
case
        MA
```

Now consider the following exercise.

> In EXERCISE-MA-5 there is a noble and a lady. The lady is greedy and the noble is weak. The noble wants to be king. In

EXERCISE-MA-5 there is a king. Show that the noble in EXERCISE-MA-5 may murder the king.

To do the analysis, as shown by the following trace, both rules are retrieved using the class-oriented scheme. In addition, one question is answered directly by the experimenter. Of course, either rule could have been replaced by a direct use of the precedent from which it was formed.

```
I am trying to show [NOBLE-3 MURDER KING-3]
Supply t = true, ? = don't know, l = look, or a precedent:
>l
For [NOBLE-3 MURDER KING-3]
I use indexes in NOBLE-3 NOBLE KING MAN RULER PERSON
and I have discovered: RULE-3
Matching EXERCISE-MA-5 against RULE-3
39. values match with 55. and 52. possible.
Normalized score is 75. %
Match is decisive — 29. better than next best.
I note that [NOBLE-3 WANT [NOBLE-3 AKO KING]] for use with RULE-3
To use RULE-3 I need to know if [NOBLE-3 HAS-QUALITY EVIL]
I am trying to show [NOBLE-3 HAS-QUALITY EVIL]
Supply t = true, ? = don't know, l = look, or a precedent:
>l
For [NOBLE-3 HAS-QUALITY EVIL]
I use indexes in NOBLE-3 NOBLE KING MAN RULER PERSON
and I have discovered: RULE-1
Matching EXERCISE-MA-5 against RULE-1
12. values match with 55. and 45. possible.
Normalized score is 26. %
Match is decisive — 4. better than next best.
I note that [NOBLE-3 HAS-QUALITY WEAK] for use with RULE-1
To use RULE-1 I need to know if [NOBLE-3 MARRY LADY-3]
I am trying to show [NOBLE-3 MARRY LADY-3]
Supply t = true, ? = don't know, l = look, or a precedent:
>t
I note that [LADY-3 HAS-QUALITY GREEDY] for use with RULE 1
The evidence from RULE-1 indicates [NOBLE-3 HAS-QUALITY EVIL]
The evidence from RULE-3 indicates [NOBLE-3 MURDER KING-3]
Rule RULE-4 is derived from RULE-3 RULE-1 and looks like this:

Rule
    RULE-4
if
    [LADY-7 HAS-QUALITY GREEDY]
    [NOBLE-9 MARRY LADY-7]
    [NOBLE-9 HAS-QUALITY WEAK]
    [NOBLE-9 WANT [NOBLE-9 AKO KING]]
then
    [NOBLE-9 MURDER KING-9]
case
    RULE-3 RULE-1
```

In the example, one *if* part of RULE-3 is replaced by the *if* parts of RULE-1, and the classes are specialized to LADY and NOBLE. Thus the new rule, RULE-4, has a larger *if* part than those of the rules it is based on. Not enough has been done to speculate on whether a trend toward larger and larger *if* parts will emerge and become a problem. (Note that it is easy to construct other examples in which rules collapse together when common causes are discovered for things in the *if* parts.)

The example illustrates that new learning can be built on top of old. The material for a new rule can come from a mixture of retrieved rules and specified precedents as the situation demands. Several rules and precedents may be needed to do something complicated.

6.2. Examples from politics world

Consider the following description.

> In WW there are Germany Hitler Poland and England. Germany is an misled country. Hitler is a greedy ambitious dictator. Poland owns Poland. England is a loyal country. Germany is misled because Germany is weak and because Hitler rules Germany and because Hitler is greedy. Hitler persuades Germany to want to own Poland. Hitler influenced Germany because Hitler is greedy and because Hitler rules Germany. Germany attacks Poland with Hitler using some tanks because Germany wants to own Poland and because Germany is misled. Poland is a victim-country. England is angry. England attacks Germany because Germany attacked Poland and because England is loyal. Germany is defeated because England attacks Germany. Hitler kills Hitler.

It may seem familiar, for it was constructed by word substitution and minor modification from *Macbeth*. Consequently all the *Macbeth* exercises have parallels too, the following for example.

> In EXERCISE-WW-1 there are a country and a dictator. The dictator ruled the country. The country is weak and the dictator is greedy. Show that the country in EXERCISE-WW-1 may be misled.

```
:
The evidence from WW indicates [COUNTRY-1 HAS-QUALITY MISLED]
Rule RULE-15 is derived from WW and looks like this:
```

```
Rule
      RULE-15
if
      [COUNTRY-7 HAS-QUALITY WEAK]
      [DICTATOR-8 RULE COUNTRY-7]
      [DICTATOR-8 HAS-QUALITY GREEDY]
then
      [COUNTRY-7 HAS-QUALITY MISLED]
case
      WW
```

6.3. Indexing and retrieving with parts—Diabetes world

In this example, a medical case, parts of things are involved.

> In C1 there is John. John is a patient and a diabetic. John's blood is some blood. The blood's blood-sugar is high because the blood's blood-insulin is low. The blood-insulin is low because John's pancreas is unhealthy. John is a diabetic because John's pancreas is unhealthy. John's heart is unhealthy because the blood-sugar is high. John takes some medicine-insulin because the blood-insulin is low. John needs to take the medicine-insulin because it is bad that the blood's blood-insulin is low and because John's taking of the medicine-insulin prevents the blood's blood-insulin being low.
>
> In C2 there is Tom and some medicine-insulin. Tom is a diabetic. Tom's blood is some blood. The blood's blood-sugar is high. The blood's blood-insulin is some blood-insulin. In C2 show that Tom's heart is unhealthy.

In the result, two parts appear, HEART-3 and BLOOD-SUGAR-4. These parts are known to be parts of the same thing by virtue of PART-OF relations. (The means that the rule-using apparatus can deal properly with rules in which the parts involved come from either the same or different things.)

```
I am trying to show [Heart-2 HAS-QUALITY UNHEALTHY]
Supply t = true, ? = don't know, l = look, or a precedent:
 > C1
Matching C1 against C2
46. values match with 151. and 53. possible.
Normalized score is 86. %
Match is decisive — 2. better than next best.
I note that [BLOOD-SUGAR-2 HAS-QUALITY HIGH] for use with C1
The evidence from C1 indicates [HEART-2 HAS-QUALITY UNHEALTHY]
Rule RULE-31 is derived from C1 and looks like this:
```

Rule
RULE-31
if
[BLOOD-SUGAR-4 HAS-QUALITY HIGH]
then
[HEART-3 HAS-QUALITY UNHEALTHY]
case
C1
⋮

There are so many parts of John and parts of those parts that it is much faster to do the exercise problem with the discovered rule than with the original case.

In C2 verify that Tom's heart is unhealthy.

I am trying to show [HEART-2 HAS-QUALITY UNHEALTHY]
Supply t = true, ? = don't know, l = look, or a precedent:
>l
For [HEART-2 HAS-QUALITY UNHEALTHY]
I use indexes in TOM DIABETIC PATIENT PERSON
and I have discovered: RULE-31
Matching C2 against RULE-31
27. values match with 54. and 54. possible.
Normalized score is 50. %
Match is decisive — 6. better than next best.
I note that [BLOOD-SUGAR-2 HAS-QUALITY HIGH] for use with RULE-31
The evidence from RULE-31 indicates [HEART-2 HAS-QUALITY UNHEALTHY]

Note also that the indexing places the entry under DIABETIC, not HEART. The result is that the rule will be found whenever it is desired to show that the heart of a diabetic is unhealthy, not whenever it is desired to show that any heart is unhealthy. In general, when working with actors that are parts of something, it seems sensible to chase the PART-OF relations to the end and store in what is found. Fig. 9 illustrates.

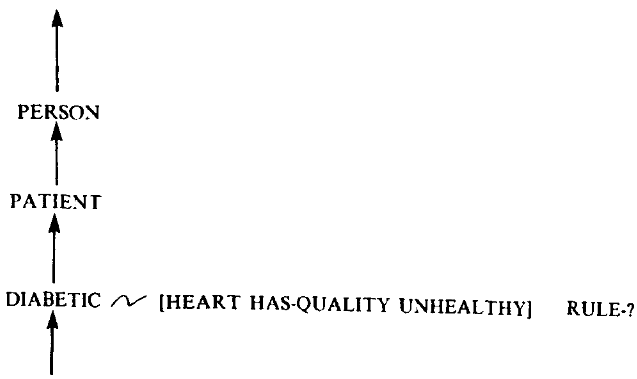

FIG. 9. The rule for unhealthy hearts is stored under DIABETIC since the unhealthy heart used to construct the rule was part of a diabetic.

6.4. Reasoning by covering an antecedent—Diabetes world

Consider Fig. 10. A precedent suggests that a relation leads to a number of consequents. A number of relations in the exercise parallel those in the tree of precedent consequents. These parallel relations are situated such that the tree would be said to be satisfied if it were an AND tree. The CAUSE relations point the wrong way, however, so the tree is not an AND tree.

Still there is some heuristic evidence that the root relation holds in the exercise, for it is capable of explaining all of the known relations according to the precedent. Let us say that the tree is *covered.*

Indeed, even if one of the covering relations is the relation to be shown, it still may make sense to assume the root relation, since it seems natural and human-like to seek complete, tidy explanations.

The need for this came up in a diabetes experiment.

In C2 show that Tom takes the medicine-insulin.

```
I am trying to show [TOM TAKE MEDICINE-INSULIN-2]
Supply t = true, ? = don't know, l = look, or a precedent:
>c1
Should I use the last match between C2 and C1?
>y
Beware! Use of match require creating PANCREAS-2 for PANCREAS-1
I note that [BLOOD-SUGAR-2 HAS-QUALITY HIGH] for use with C1
I note that [TOM AKO DIABETIC] for use with C1
All consequences of [PANCREAS-2 HAS-QUALITY UNHEALTHY] are confirmed.
The evidence from C1 indicates [TOM TAKE MEDICINE-INSULIN-2]
```

It is concluded that the patient needs to take insulin even though the prec-

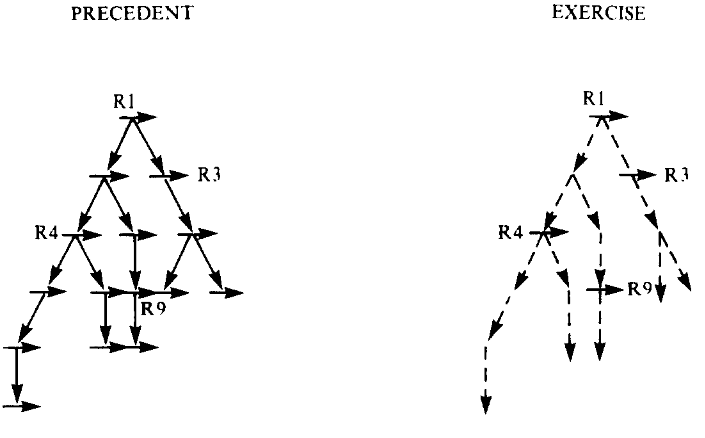

FIG. 10. One relation in the precedent leads to a number of consequences, some of which are mirrored in the exercise. In this illustration, R3, R4, and R9 are said to cover R1.

edent's causal chain leading to taking insulin includes only low insulin and an unhealthy pancreas, neither of which are established in the exercise. The unhealthy pancreas can be reasonably assumed, however, since it is covered by a combination of existing relations and the relation to be shown. The covering relations are that Tom is diabetic, that he has high blood sugar, and that he takes insulin. All these covering relations appear in the resulting rule.

```
Rule
      RULE-32
if
      [DIABETIC-2 AKO DIABETIC]
      [BLOOD-SUGAR-5 HAS-QUALITY HIGH]
then
      [DIABETIC-2 TAKE MEDICINE-INSULIN-3]
case
      C1
```

The example illustrates that things in the *if* part of a new rule are not necessarily antecedents to things in the *then* part. The learning procedure does more than rule concatenation.

7. Open Questions

It is plain that this work is only a beginning. The next iteration through thinking and implementation and experiment should be more ambitious, dealing with weaknesses like the following representative ones.

- There is no way to criticize a rule. Davis showed how the typical member of a rule group can be used to suggest additions to a newly acquired rule supplied directly by a human expert [3]. There may be a way to use something like Davis's ideas with automatically generated rules.
- There is no way to handle degree of certainty of cause. Moreover, there is no way to handle subcategories of cause such as those sketched by Rieger [13]. Distinguishing between things like enablement causes and main causes could make rule formation and rule indexing and retrieving better.
- There is no way to handle constraints about quantities such as those constraints that appear in the work of Forbus [7].
- There is no way to summarize an episode in a story so as to make a general precis.
- There are no satisfying ideas about the role of abstraction in doing matching and indexing and retrieving.
- The representation for time is impoverished. Similarly quantification, negation, disjunction, and perspective are missing.

8. Conclusion: Simple Ideas Have Promise

This paper is about a set of ideas that enable learning and reasoning to take place by analogy in domains that satisfy certain restrictions.

- The situations in the domain can be represented by the relations between the parts together with the classes and properties of those parts.
- The importance of a part of a description is determined by the constraints it participates in.
- Constraints that once determine something will tend to do so again.

Things that involve spatial, visual, and aural reasoning do not seem to satisfy the restrictions. Things that involve Management, Political Science, Economics, Law, and Medicine do seem to satisfy the restrictions, however, and are targets for the learning and reasoning ideas developed here.

- Actor–object representation.
- Importance-dominated matching.
- Analogy-based reasoning.
- Learned if-then rules.
- Actor-oriented indexing.
- Deduction by covering an antecedant.

It is now clear that simple exercises can be done by way of analogy with precedents. Rules can be created at the same time, and they can be indexed and retrieved using the classes of the objects involved. Experience informs, or at least prejudices.

9. Related Work

This work has roots in the ARCH system [19], and the FOX system [20], both identified by the typical things involved in the learning. The most direct antecedent, however, is the MACBETH system [21], which concentrated on representation and matching issues. Others who have worked on matching include Michalski, Hayes-Roth, and Vere. See [4] for an excellent review. For analogy itself, Gentner's work on the dimensions of analogy is another important precedent [8].

With respect to context, Marcus has addressed questions about natural rule groups in the domain of English parsing [12]. With respect to indexing and retrieving, both Shortliffe [17] and Davis [3] have used various mechanisms different from the one introduced here. With respect to constraint, Davis has also shown that the regular structure of if-then rules simplifies the problem of knowledge acquisition [3]. Similarly, Berwick has showed how to learn the rules for English parsing required by Marcus's parser, arguing that a highly constrained rule structure is a prerequisite for such learning [1].

Papers by both Shortliffe [17] and by Duda, Hart, and Nilsson [5] show how to superimpose a kind of probability theory on top of basic rule systems to establish the likelihood of the conclusions that are reached.

In addition, debt is due to Schank [16], Wilks [18], and many linguists who have wondered how thinking is determined by experience. William A. Martin may have had the deepest insight along these lines when he remarked in a lecture, "You can't learn anything unless you almost know already."

ACKNOWLEDGMENT

This paper was improved by comments from Robert Berwick, J. Michael Brady, Bruce Buchanan, Randy Davis, Gavan Duffy, Boris Katz, John Mallery, and Karen Prendergast.

Appendix A

This appendix contains the notes referenced in the body of this paper.

A.1. The implementation is based on FRL

The actual implementation was done using a version of FRL, an acronym for a LISP-based Frame Representation Language, developed by Bruce Roberts and Ira Goldstein [14, 15]. FRL was used because FRL has handy mechanisms for asserting relations and attaching supplementary descriptions to them. In FRL terms, an actor–act–object combination is expressed as a *frame*, a *slot* in the frame, and a *value* in the slot. A supplementary description node for an actor–act–object combination is expressed in the form of a so-called *comment frame* attached to the frame-slot-value combination.

A.2. A sample read by the Katz translator

The following illustrates the greater power of Katz's translator relative to the simple one used in this work.

> In the beginning of the story Duncan was a king. Macbeth who was a happy nobleman married Lady-Macbeth. She desired that he become king. She persuaded Macbeth who she loved to murder Duncan because she was greedy and ambitious and also because she was cruel. Soon she decided to kill herself because she and Macbeth were unhappy. Macbeth's murder of Duncan caused Macduff who was a loyal nobleman to kill Macbeth.

A.3. Matcher performance on Shakespeare precis

One curious sort of robust behavior was observed in the matcher: it gives the same relative results even as it continues to be debugged.

TABLE 1

	MA	HA	JU	OT	TA
Macbeth	199				
Hamlet	110	235			
Julius Caesar	71	50	154		
Othello	40	40	56	187	
Taming of the Shrew	20	18	21	25	111

TABLE 2

	MA	HA	JU	OT	TA
Macbeth	100				
Hamlet	55	100			
Julius Caesar	46	32	100		
Othello	21	21	36	100	
Taming of the Shrew	18	15	18	22	100

The *raw* scores, the total number of corresponding things, for some Shakespeare comparisons are as shown in Table 1. Similar relative ordering is observed when the similarity measure is taken to be the *normalized* scores, the number of corresponding things divided by the larger of the number of corresponding things observed when matching the two situations involved against themselves (Table 2).

Tables showing the same alternatives, but with only the cause-determined important relations counted, show similar relative ordering.

A.4. Several words imply CAUSE relations

Many people have attended to the role of cause, particularly Schank [16] and Wilks [18]. To handle cause here, the following conventions were honored, based on the work of Givon [9].

- Acts and relations can cause or prevent acts and relations. The inverses of CAUSE and PREVENT are CAUSED-BY and PREVENTED-BY.
- People can persuade and dissuade. Persuade indicates influence and cause. A CAUSE relation is therefore generated by a demon whenever PERSUADE is used. Dissuade similarly indicates prevent. Since persuaders and dissuaders normally intend for something to happen, INTEND relations are generated too, again by demons. Since the person persuaded or dissuaded retains control of what is happening, a demon-placed CONTROL relation so indicates. See part a of Fig. 11 for example showing what happens given this:

  Lady-Macbeth persuades Macbeth to murder Duncan.

- People can also order and forbid. For the moment ORDER and FORBID simply carry demons that place PERSUADE and DISSUADE relations and trigger their demons in turn.
- People can force. The person forcing has control of what is happening. See part b of the figure for an example showing what happens given this use of a FORCE relation.

  Lady-Macbeth forces Macbeth to murder Duncan.

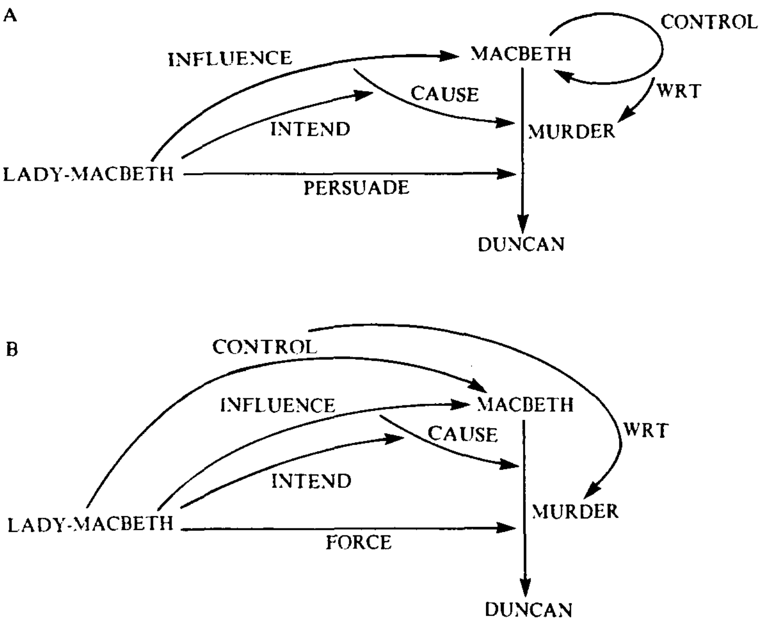

FIG. 11. The meaning of *persuade* and *force*. All relations are placed by demons whenever PERSUADE and FORCE are used.

These conventions differ slightly from those used before [21] in that only acts and relations are allowed to cause or prevent acts and relations. People are not.

REFERENCES

1. Berwick, R., Computational analogues of constraints on grammars, *Proc. 18th Annual Meeting of the Association for Computational Linguistics*, 1980.
2. Brotsky, D. Efficient matching of situations, unpublished.
3. Davis, R., Applications of meta level knowledge to the construction, maintenance, and use of large knowledge bases, Ph. D. Thesis, Stanford University, Stanford, CA, 1979.
4. Ditterich, T.G., and Michalski, S., Inductive learning of structural descriptions, *Artificial Intelligence* **16**(3) (1981) 257–294.
5. Duda, R.O., Hart, P.E., and Nilsson, N.J., Subjective Bayesian methods for rule-base inference systems, *Proc. National Computer Conference* (*AFIPS Conference Proceedings*) **45** (1976).
6. Filmore, C.J., The case for case, in: E. Bach and R. Harms, (Eds.), *Universals in Linguistic Theory* (Holt, Rinehart, and Winston, New York, 1968).
7. Forbus, K., Qualitative process theory, Artificial Intelligence Laboratory Memo No. 664, M.I.T., Cambridge, MA, 1982.
8. Gentner, D., The structure of Analogical models in Science, Tech. Rept., No. 4451, Bolt Beranek and Newman, 1980; ibid., Are scientific analogies metaphors, in: D.S. Miall, Ed., *Metaphor: Problems and Perspectives* (Harvester, Brighton, Sussex, 1981).
9. Givon, T., Cause and control: on the Semantics of Interpersonal Manipulation, in: J. Kimball, Ed., *Syntax and Semantics*, Vol. IV (Academic Press, New York, 1975).

10. Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., and Slocum, J., Developing a natural language interface to complex data, *ACM Trans. Database Systems* **3**(2) (1978).
11. Katz, B., A three-step procedure for language generation, Artificial Intelligence Laboratory Memo No. 599, M.I.T., Cambridge, MA, 1980.
12. Marcus, M.P., *A Theory of Syntactic Recognition for Natural Language* (M.I.T., Cambridge, MA, 1980).
13. Rieger, C., On organization of knowledge for problem solving and language comprehension, *Artificial Intelligence* **7**(2) (1978) 89--128.
14. Roberts, R.B. and Goldstein, I.P., The FRL primer, Artificial Intelligence Laboratory Memo No. 408, M.I.T., Cambridge, MA, 1977.
15. Roberts, R.B. and Goldstein, I.P., The FRL manual, Artificial Intelligence Laboratory Memo No. 409, M.I.T., Cambridge, MA, 1977.
16. Schank, R.C., *Conceptual Information Processing* (North-Holland, Amsterdam, 1975).
17. Shortliffe, E.C., *Computer Based Medical Consultations: MYCIN* (American Elsevier, New York, 1976).
18. Wilks, Y.A., *Grammar, Meaning, and the Machine Analysis of Language* (Routledge and Kegan Paul, London, 1972).
19. Winston, P.H. Learning structural descriptions from examples, Ph.D. Thesis, M.I.T., 1970; ibid., in: P.H. Winston, Ed., *The Psychology of Computer Vision* (McGraw-Hill, New York, 1975).
20. Winston, P.H., Learning by creating and justifying transfer frames, *Artificial Intelligence* **10**(2) (1978) 147–172.
21. Winston, P.H., Learning and reasoning by analogy, *Comm. ACM* **23**(12) (1980).
22. Winston, P.H., Learning new principles from precedents and exercises: the details, Artificial Intelligence Laboratory Memo No. 632, M.I.T., Cambridge, MA, 1981.

Received May 1981; revised version received November 1981