

18

Methods for Generating Explanations

A. Carlisle Scott, William J. Clancey,
Randall Davis, and Edward H. Shortliffe

A computer program that models an expert in a given domain is more likely to be accepted by experts in that domain, and by nonexperts seeking its advice, if the system can explain its actions. This chapter discusses the general characteristics of explanation capabilities for rule-based systems: what types of explanations they should be able to give, what types of knowledge they will need in order to give these explanations, and how this knowledge might be organized (Figure 18-1). The explanation facility in MYCIN is discussed to illustrate how the various problems can be approached.

A consultative rule-based system need not be a psychological model, imitating a human's reasoning process. The important point is that the system and a human expert use the same (or similar) knowledge about the domain to arrive at the same (or similar) answers to a given problem. The system's *knowledge base* contains the domain-specific knowledge of an expert as well as facts about a particular problem under consideration. When a rule is used, its actions make changes to the internal data base, which contains the system's *decisions* or *deductions*.

The process of trying rules and taking actions can be compared to reasoning, and explanations require displays of how the rules use the information provided by the user to make various intermediate deductions and finally to arrive at the answer. If the information contained in these rules adequately shows why an action was taken (without getting into programming details), an explanation can simply entail printing each rule or its free-text translation.

This chapter is a revised version of a paper originally appearing in *American Journal of Computational Linguistics*, Microfiche 62, 1977. Copyright © 1977 by American Society for Computational Linguistics. All rights reserved. Used with permission.

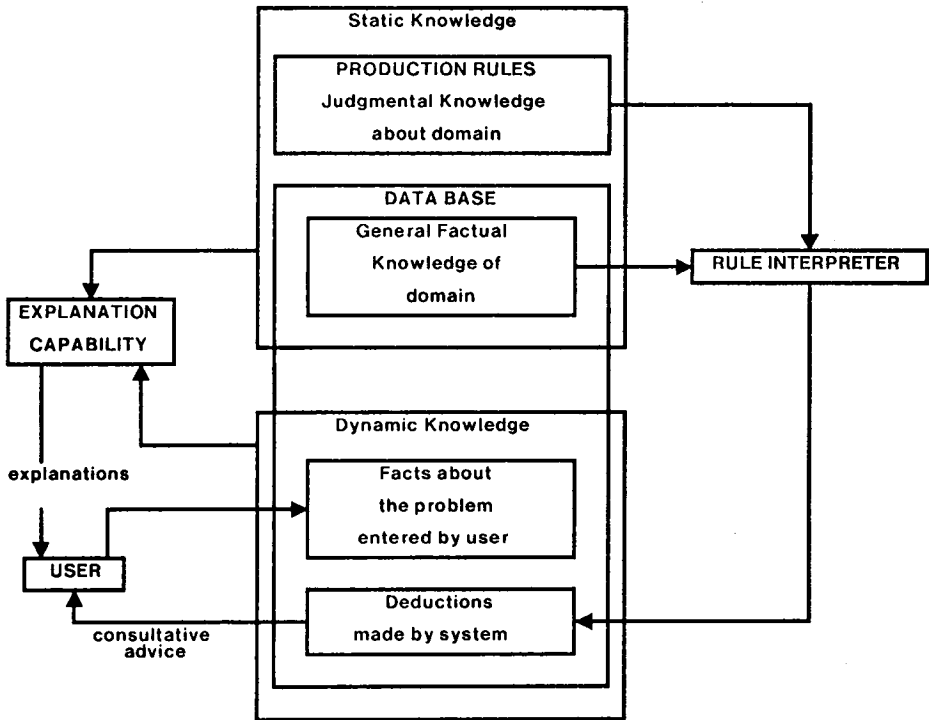


FIGURE 18-1 A rule-based consultation system with explanation capability. The three components of a rule-based system (a rule interpreter, a set of production rules, and a data base) are augmented by an explanation capability. The data base is made up of general facts about the system's domain of expertise, facts that the user enters about a specific problem, and deductions made about the problem by the system's rules. These deductions form the basis of the system's consultative advice. The explanation capability makes use of the system's knowledge base to give the user explanations. This knowledge base is made up of static domain-specific knowledge (both factual and judgmental) and dynamic knowledge specific to a particular problem.

Performance Characteristics of an Explanation Capability

The purpose of an explanation capability (EC) is to give the user access to as much of the system's knowledge as possible. Ideally, it should be easy for a user to get a complete, understandable answer to any sort of question about the system's knowledge and operation—both in general terms and

with reference to a particular consultation. This implies three major goals in the development of an explanation capability:

1. It is important to ensure that the EC can handle questions about all relevant aspects of the system's knowledge and actions. It should be capable of giving several basic types of explanation, for example,
 - how it made a certain decision
 - how it used a piece of information
 - what decision it made about some subproblem
 - why it did not use a certain piece of information
 - why it failed to make a certain decision
 - why it required a certain piece of information
 - why it did not require a certain piece of information
 - how it will find out a certain piece of information (while the consultation is in progress)
 - what the system is currently doing (while the consultation is in progress)
2. It is important to enable the user to get an explanation that answers the question completely and comprehensively.
3. Finally, it is also necessary to make the EC easy to use. A novice should be able to use the EC without first spending a large amount of time learning how to request explanations.

We will distinguish two functions for an EC: the reasoning status checker (RSC) to be used during the consultation, and the general question answerer (GQA) to be used during the consultation or after the system has printed its results. An RSC answers questions asked during a consultation about the status of the system's reasoning process. A few simple commands often suffice to handle the questions that the RSC is expected to answer. A GQA answers questions about the current state of the system's knowledge base, including both static domain knowledge and facts accumulated during the consultation. It must recognize a wide range of question types about many aspects of the system's knowledge. For this reason, a few simple commands that are easy to learn but still cover all the possible questions that might be asked may be difficult to define. Consequently, natural language processing may be important for a useful GQA.

In an interactive consultation, the system periodically requests information about the problem. This offers the user an opportunity to request explanations while the consultation is in progress. In noninteractive consultations, the user has no opportunity to interact with the system until after it has printed its conclusions. Unless there is a mechanism for interrupting the reasoning process and asking questions, the EC for such a

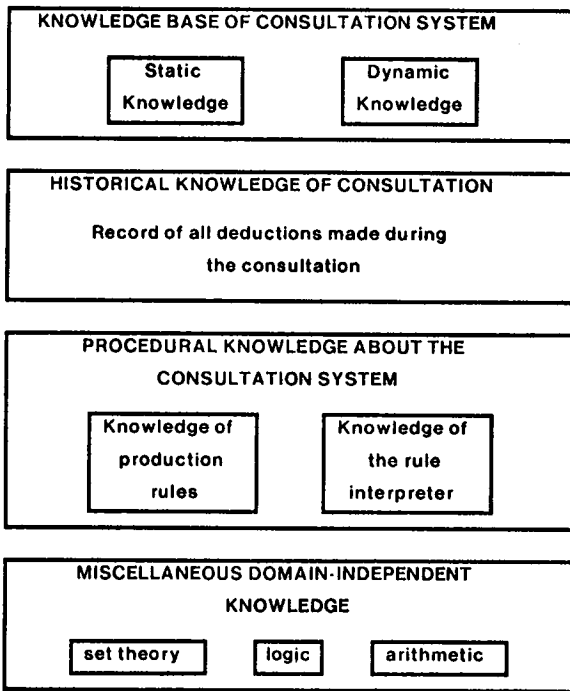


FIGURE 18-2 Knowledge requirements for an explanation capability (EC). Access to the consultation system's knowledge base is a prerequisite for adequate performance of the EC. Other types of knowledge may be added to the system to enable the EC to answer a wider range of questions.

system will be limited to questions about the system's final knowledge state. It will have no RSC.

An EC must know what is in the system's knowledge base and how it is organized (Figure 18-2). In order to give explanations of the system's actions, an EC also needs to understand how the system's rule interpreter works: when rules will be tried, how they can fail, and what causes the interpreter to try one rule but not another. This general "schema" for how or why certain rules are used, together with a comprehensive record of the specific actions taken during a particular consultation, can be used as a basis for explaining the results of that consultation.

An RSC will need a record of what the system has done in order to explain how it arrived at the current step. General knowledge of how the rule interpreter works is necessary to explain where the current step will lead. The ability to understand individual rules is necessary to the extent

that the content of a rule may explain why it was necessary to use that rule or may affect which rules will be tried in the future.

A GQA will need more information about the system since the scope of its explanations is much broader. It must know how the system stores knowledge about its area of expertise (the static knowledge with which it starts each consultation), how it stores facts gathered during a particular consultation (its dynamic knowledge), and how the dynamic knowledge was obtained or inferred. Thus the GQA must have access to all the information that the RSC uses: a detailed record of the consultation, an understanding of the rule interpreter, and the ability to understand rules.

18.1 Design Considerations

To complement the preceding discussion of an EC, we must describe relevant design considerations for the parent consultation system. This discussion is not meant to define the "correct" way of representing or organizing knowledge, but rather to mention factors that should be taken into account when deciding what representation or organization will be best for a developing system.

The first step is to decide what basic types of questions the system should be able to answer. This will have a direct influence on how the EC is implemented. It is important, however, to make the initial design flexible enough to accommodate possible future additions; if the basic forms are sufficiently diverse, limited natural language understanding may be necessary, depending on the level of performance expected of the EC.

The format and organization of the consultation system's knowledge base will also affect the design of an EC because both static and dynamic knowledge must be readily accessible. The more disorganized the knowledge base, the more difficult will be the task of the EC because more complicated routines will be needed to access the desired information. Similarly, when the ordering of events is important, the dynamic record must reflect that ordering as well as the reasons why each event occurred.

The EC often needs to understand the underlying semantics of individual rules. This requirement can be met by having the system's knowledge base include a description of what each rule means, encoded in a form that is of use to the EC. If the format of the system's rules is highly stylized and well defined, however, it is possible instead to implement a mechanism for "reading" the rules and describing their meaning in natural language. This can be achieved through a high-level description of the individual components of the rules, one that tells what each element means. If the rule set consists of a large number of rules, and they are composed entirely of a relatively small number of primitive elements, this second approach has the advantage that less information needs to be

stored—a description of each of the primitive components, as opposed to a description of each rule. When new rules are added to the system, the first approach requires that descriptions of these rules must be added. With the second approach, provided that the new rules are constructed from the standard rule components, no additional descriptive information is needed.

As well as understanding rules in the knowledge base, an EC must also be able to “read” the interpreter or have access to some stored description of how the interpreter works. A third option is to build knowledge of how the interpreter works directly into the EC; the information need not be stated explicitly but can be used implicitly by the programmer in writing the actual EC code. The EC can then function as a set of “specialists,” each capable of giving a single type of explanation.

Finally, the GQA generally must be able to make deductions from facts in the knowledge base. If logic is needed only to determine the answers to questions of a certain type, it may be possible to build the necessary deductions into the specialist for answering that type of question. On the other hand, the GQA will often need to be expanded to do more than simply give explanations of the system’s actions or query its data base—it will be expected to answer questions involving inferences (e.g., to check for equality or set membership, to make arithmetical comparisons, or to make logical deductions). Information of this type can often be embodied in a new kind of specialist that deals with logical deduction or comparison.

18.2 An Example—MYCIN

MYCIN’s domain of expertise, its mechanisms for knowledge representation, and its inference mechanisms have been discussed in detail earlier in this book. We will not repeat those points here except to emphasize issues that relate directly to this discussion.

18.2.1 Organization of Knowledge in MYCIN

As we have discussed, an EC must have access to all components of the system’s knowledge base. MYCIN’s knowledge base consists of static medical knowledge plus dynamic knowledge about a specific consultation. Static knowledge is further classified as *factual* or *judgmental*. Factual knowledge consists of facts that are medically valid, by definition and with certainty, independent of the particular case. Judgmental knowledge, on the other hand, is composed of the rules acquired from experts. Although this knowledge is also assumed to be medically valid, the indicated inferences are often drawn with less than complete certainty and are seldom defini-

tional. The conventions for storing both dynamic and static knowledge, including attribute-object-value triples, tables, lists, and rules themselves, are described in detail in Chapter 5.

Knowledge of Rule Structure

Each of MYCIN's rules is composed of a small number of conceptual primitives drawn from a library of 60 such primitives that make up the language in which rules are written. This design has facilitated the implementation of a mechanism for translating rules into English (described in Chapter 5). Each primitive function has a *template* (Chapter 9) with blanks to be filled in using translations of the function's arguments. A large part of MYCIN's explanation capability depends on this ability to translate rules into a form that the user can understand.

In order to understand rules, the system's various specialists use a small amount of knowledge about rules in general, together with descriptions or templates of each of the rule components. As an example, the following rule (shown in LISP and its English translation) is composed of the units \$AND, SAME, and CONCLUDE:

RULE009

PREMISE: (\$AND (SAME CNTXT GRAM GRAMNEG)
(SAME CNTXT MORPH COCCUS))
ACTION: (CONCLUDE CNTXT IDENTITY NEISSERIA TALLY 800)

IF: 1) The gram stain of the organism is gramneg, and
2) The morphology of the organism is coccus
THEN: There is strongly suggestive evidence (.8) that the identity
of the organism is Neisseria

When the rule is used, the LISP atom CNTXT is bound to some object, the context to which the rule is applied; see Chapter 5. The template for CONCLUDE is shown below. This describes each of the arguments to the function: first, an object (context); second, an attribute (clinical parameter); third, a value for this parameter; fourth, the tally, or degree of certainty, of the premise; and last, the certainty factor, a measure of how strong our belief in this conclusion would be if the premise of the rule were definitely true.

Template for CONCLUDE: (CNTXT PARM VALU TALLY CF)

Having a small number of rule components also facilitates examination of rules to see which might be applicable to the explanation at hand. MYCIN's knowledge of rules, therefore, takes the form of a general mechanism for "reading" them. On the other hand, no attempt has been made to read the code of the rule interpreter. Procedural knowledge about the interpreter is embodied in "specialists," each capable of answering a single

type of question. Each specialist knows how the relevant part of the control structure works and what pieces of knowledge it uses.

To understand how a specialist might use a template such as that shown above, consider an explanation that involves finding all rules that can conclude that the identity of an organism is *Neisseria*. The appropriate specialist would start with those rules used by the system to conclude values for the parameter IDENTITY. Using templates of the various action functions that appear in each of these rules, the specialist picks out only those (like Rule 009) that have NEISSERIA in their VALU slot.

This also illustrates the sort of knowledge that can be built into a specialist. The specialist knows that the control structure uses stored lists telling which rules can be used to determine the value of each parameter. Furthermore, it knows that it is necessary to look only at the rules' actions since it is the action that *concludes* facts, while the premise *uses* facts.

The History Tree

Many of the EC's specialists need a record of the interaction with the user. This record is built during the consultation and is organized into a tree structure called the *history tree*, which reflects MYCIN's goal-directed approach. Each node in the tree represents a goal and contains information about how the system tried to accomplish this goal (by asking the user or by trying rules). Associated with each rule is a record of whether or not the rule succeeded, and if not, why it failed. If evaluating the premise of a rule causes the system to trace a new parameter, thereby setting up a new subgoal, the node for this subgoal is the offspring of the node containing the rule that caused the tracing. Figure 18-3 shows part of a representative history tree. In this example, Rule 003 caused the tracing of the parameter CATEGORY, which is used in the premise of this rule.

Other Domain-Independent Knowledge

MYCIN's question-answering ability is limited to describing the system's actions and explaining what facts the system knows. The system also has capabilities for the use of specialized logic. For example, to explain why a particular decision was *not* made, MYCIN recognizes that a reasonable response is to explain what *prevented* the system from using rules that *would have made* that decision. For situations such as this, the necessary logic is built into the appropriate specialist; there is no general representation of knowledge about logic, arithmetic, or set theory. To find out if ORGANISM-1 and ORGANISM-2 have the same identity, for example, it is necessary for the user to ask separately for the identity of each organism, then to compare the answers to these questions.

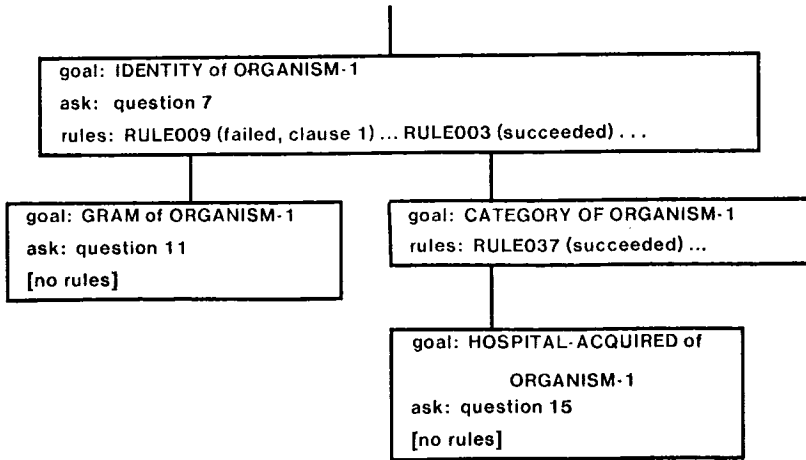


FIGURE 18-3 Portion of a history tree. (Rule 009 is shown in the text; see Figure 18-4 for Rule 003 and Rule 037.)

18.2.2 Scope of MYCIN's Explanation Capability (EC)

Because we wish to allow the user to see how MYCIN makes all its decisions, we have tried to anticipate all types of questions a user might ask and to make every part of the system's knowledge base and reasoning process accessible. The EC consists of several specialists, each capable of giving one type of explanation. These specialists are grouped into three sets: one for explaining what the system is doing at a given time, one for answering questions about the system's static knowledge base, and one for answering questions about the dynamic knowledge base. The first set forms MYCIN's reasoning status checker; the second and third together make up the system's general question answerer.

MYCIN's Reasoning Status Checker (RSC)

Whenever MYCIN asks a question, the user is allowed to examine the current reasoning chain by asking WHY the piece of information being sought by the system is important. As explained above, the system asks a question in order to find out about its current goal. Consider the partial history tree shown in Figure 18-3. HOSPITAL-ACQUIRED is one subgoal, CATEGORY is another at the next level up, and Rule 037 links them. The "reason" for asking if the infection was hospital-acquired, then, is the program's use of Rule 037 in an effort to determine the value of the higher subgoal CATEGORY. The answer to WHY at this point is thus a lucid display of the goals and the rules linking them. Since any WHY could

[preceded by the first 14 questions in the consultation]

15) Is the patient's illness with ORGANISM-1 a hospital-acquired infection?

**** WHY**

[i.e. WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalized?]

[1.0] This will aid in determining the category of ORGANISM-1.

It has already been established that

[1.1] the gram stain of ORGANISM-1 is gramneg, and

[1.2] the morphology of ORGANISM-1 is rod, and

[1.3] the aerobicity of ORGANISM-1 is facultative

Therefore, if

[1.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized

then

there is strongly suggestive evidence (.8) that the category of ORGANISM-1 is enterobacteriaceae

ALSO: there is weakly suggestive evidence (.1) that the identity of ORGANISM-1 is pseudomonas-aeruginosa

[RULE037]

**** WHY**

[i.e. WHY is it important to determine the category of ORGANISM-1?]

[2.0] . . . in order to determine the identity of ORGANISM-1.

It has already been established that

[2.1] this blood culture was taken from a sterile source

Therefore, if

[2.2] this current organism and at least one of the list of members associated with the category of the organism agree with respect to the following properties: air conformation

then

There is strongly suggestive evidence (.9) that each of them is the identity of ORGANISM-1

[RULE003]

**** HOW 1.3**

[i.e. HOW was it established that the aerobicity of ORGANISM-1 is facultative?]

The following were used:

[3.1] RULE027 indicated there is strongly suggestive evidence (.8) that the aerobicity of ORGANISM-1 is facultative

[3.2] RULE027 indicated there is weakly suggestive evidence (.2) that the aerobicity of ORGANISM-1 is anaerobic

Since this gave a cumulative CF of (.8) for facultative, and (.2) for anaerobic, it has been established that the aerobicity of ORGANISM-1 is facultative

**** HOW 1.1**

[i.e. HOW was it established that the gram stain of ORGANISM-1 is gramneg?]

You said so [question 11].

FIGURE 18-4 MYCIN's reasoning status checker (user entries follow the double asterisks).

potentially have several interpretations, and we wanted to avoid extensive natural language processing in this component of MYCIN's EC, MYCIN first restates its understanding of the meaning of the user's WHY (Figure 18-4). The higher goal is then presented, followed by the rule. The clauses in the premise of the rule are divided into those already established and those yet to be determined. Finally, since rules may have multiple conclu-

IS BLOOD A STERILE SITE?
 WHAT ARE THE NONSTERILE SITES?
 WHAT ORGANISMS ARE LIKELY TO BE FOUND IN THE THROAT?
 IS BACTEROIDES AEROBIC?
 WHAT METHODS OF COLLECTING SPUTUM CULTURES DO YOU CONSIDER?
 WHAT DOSAGE OF STREPTOMYCIN DO YOU GENERALLY RECOMMEND?
 HOW DO YOU DECIDE THAT AN ORGANISM MIGHT BE STREPTOCOCCUS?
 WHY DO YOU ASK WHETHER THE PATIENT HAS A FEVER OF UNKNOWN ORIGIN?
 WHAT DRUGS WOULD YOU CONSIDER TO TREAT E.COLI?
 HOW DO YOU USE THE SITE OF THE CULTURE TO DECIDE AN ORGANISM'S IDENTITY?

FIGURE 18-5 Sample questions about MYCIN's static knowledge.

sions about different clinical parameters, the relevant conclusion is presented first and all others follow.

As Figure 18-4 illustrates, additional links in the reasoning chain can be examined by repeating the WHY command. For any of the subgoals mentioned in answer to a WHY, the user may ask HOW this goal was (or will be) achieved. MYCIN's reasoning status checker is described in more detail by Shortliffe et al. (1975) and Davis et al. (1977).

MYCIN's General Question Answerer (GQA)

The question-answering part of the system has natural language routines for analyzing the user's input. The system recognizes questions phrased in a number of ways, thereby making the question-answering facility easier to use. Questions about the static knowledge base may deal with judgmental knowledge (e.g., rules used to conclude a certain piece of information) or they may ask about factual knowledge (e.g., entries in tables and lists). Some questions about static knowledge are shown in Figure 18-5.

Perhaps the more important part of the question-answering system is its ability to answer questions about a particular consultation. While some users may be interested in checking the extent of MYCIN's static knowledge, most questions will ask for a justification of, or for the rationale behind, particular decisions that were made during the consultation. Listed in Figure 18-6 are the types of questions about dynamic knowledge that can be handled at present. A few examples of each type are given. The slot <cntxt> indicates some context that was discussed in the consultation; <parm> is some clinical parameter of this context; <rule> is one of the system's decision rules. Before a question can be answered, it must be classified as belonging to one of these groups. As Figure 18-6 illustrates, each question type may be asked in a variety of ways, some specifying the parameter's value, some phrased in the negative, and so forth. MYCIN's natural language processor must classify the questions, then determine what specific clinical parameters, rules, etc., are being referenced.

1. What is <parm> of <cntxt>?

**TO WHAT CLASS DOES ORGANISM-1 BELONG?
IS ORGANISM-1 CORYNEBACTERIUM-NON-DIPHtherIAE?**

2. How do you know the value of <parm> of <cntxt>?

**HOW DO YOU KNOW THAT CULTURE-1 WAS FROM A STERILE SOURCE?
DID YOU CONSIDER THAT ORGANISM-1 MIGHT BE A BACTERIOIDES?
WHY DON'T YOU THINK THAT THE SITE OF CULTURE-1 IS URINE?
WHY DID YOU RULE OUT STREPTOCOCCUS AS A POSSIBILITY FOR ORGANISM-1?**

3. How did you use <parm> of <cntxt>?

**DID YOU CONSIDER THE FACT THAT PATIENT-1 IS A COMPROMISED HOST?
HOW DID YOU USE THE AEROBICITY OF ORGANISM-1?**

4. Why didn't you find out about <parm> of <cntxt>?

**DID YOU FIND OUT ABOUT THE CBC ASSOCIATED WITH CULTURE-1?
WHY DIDN'T YOU NEED TO KNOW WHETHER ORGANISM-1 IS A CONTAMINANT?**

5. What did <rule> tell you about <cntxt>?

**HOW WAS RULE 178 HELPFUL WHEN YOU WERE CONSIDERING ORGANISM-1?
DID RULE 116 TELL YOU ANYTHING ABOUT INFECTION-1?
WHY DIDN'T YOU USE RULE 189 FOR ORGANISM-2?**

FIGURE 18-6 Types of questions about a consultation, with examples.

18.2.3 Understanding the Question

The main emphasis in the development of MYCIN has been the creation of a system that can provide sound diagnostic and therapeutic advice in the field of infectious diseases. The explanation system was included in the system's original design in order to make the consultation program's decisions acceptable, justifiable, and instructive. Since the question-answering facility was not the primary focus of the research, it is not designed to be a sophisticated natural language understander. Instead, it uses crude techniques, relying strongly on the very specific vocabulary of the domain, to "understand" what information is being requested (Figure 18-7).

The analysis of a question is broken into three phases (Steps 1–3 of Figure 18-7): the first creates a list of *terminal*, or root, *words*; the second determines what type of question is being asked (see the classification of questions above); and the last determines what particular parameters, lists, etc., are relevant to the question. In the first and third steps, the system dictionary is important. The dictionary contains approximately 1400 words that are commonly used in the domain of infectious diseases. It includes all words that are acceptable values for a parameter, common synonyms

1. The question is reduced to a list of terminal words.
 2. Pattern matching classifies the question as a rule-retrieval question, and divides it into a premise part and an action part.
 3. Dictionary properties of the terminal words are used to determine which parameters (and their values) are relevant to each part of the question. These vocabulary clues are listed in the form (<parm> (<values>) weight) where weight is used by the scoring mechanism to determine which parameters should be eliminated from consideration.
 4. After selecting only the most strongly indicated parameters, the final translation tells what rules can answer the question: there are no restrictions on the premise, and the action must contain the parameter CONTAMINANT with any value.
 5. The answer consists of finding all rules that meet these restrictions, and printing those that the user wants to see.
-

FIGURE 18-7 Major steps in understanding a question, finding rules, and printing an answer. See Figure 18-8 for an example.

of these words, and words used elsewhere by the system in describing the parameter (e.g., when translating a rule into English or requesting the value of the parameter).

We now briefly describe how MYCIN achieves each of the five tasks outlined in Figure 18-7. An example analysis is shown in Figure 18-8.

Step 1: Reducing the Question to Terminal Words

Each word in the dictionary has a synonym pointer to its terminal word (terminal words point to themselves). For the purpose of analyzing the question, a nonterminal word is considered to be equivalent to its (terminal) synonym. Terminal words have associated with them a set of properties or descriptors (Table 18-1) that are useful in determining the meaning of a question that uses a terminal word or one of its synonyms. A given word may be modified by more than one of these properties.

The first three properties of terminal words are actually inverse pointers, generated automatically from attributes of the clinical parameters. Specifically, a word receives the “acceptable value” pointer to a clinical parameter (Property 1 in Table 18-1) if it appears in the parameter’s list of acceptable values—a list that is used during the consultation to check the user’s response to a request for the parameter’s value (see EXPECT attribute, Chapter 5).

Also, each clinical parameter, list, and table has an associated list of keywords that are commonly used when talking about that parameter, list, or table. These words are divided according to how sure we can be that a doctor is referring to this parameter, list, or table when the particular word

****WHEN DO YOU DECIDE THAT AN ORGANISM IS A CONTAMINANT?**

- [1] Terminal words: WHEN DO YOU CONCLUDE THAT A ORGANISM IS A CONTAMINANT
- [2] Question type: Rule retrieval
 Premise part: (WHEN DO YOU CONCLUDE)
 Action part: (THAT A ORGANISM IS A CONTAMINANT)
- [3] vocab. clues: (WHENINFECT (ANY) 1) (WHENSTOP (ANY) 1)
 (Premise) (WHENSTART (ANY) 1) (DURATION (ANY) 1)
 vocab. clues: (CONTAMINANT (ANY) 4) (FORM (ANY) 1)
 (Action) (SAMEBUG (ANY) 1) (COVERFOR (ANY) 1)
- [4] Final translation:
 Premise: ANY
 Action: (CONTAMINANT ANY)
- [5] The rules listed below conclude about:
 whether the organism is a contaminant
 6, 31, 351, 39, 41, 42, 44, 347, 49, 106
 Which do you wish to see?
 ** 6

RULE006

IF: 1) The culture was taken from a sterile source, and
 2) It is definite that the identity of the organism
 is one of: staphylococcus-coag-neg bacillus-
 subtilis corynebacterium-non-diphtheriae
 THEN: There is strongly suggestive evidence (.8)
 that the organism is a contaminant

FIGURE 18-8 Sample of MYCIN's analysis of a general question. (User input follows the double asterisks. Steps 1 through 4 are usually not shown to the user. See Figure 18-7 for a description of what is occurring in each of the five steps.)

TABLE 18-1 Properties of Terminal Words

1. The word is an acceptable value for some clinical parameter(s).
2. The word *always* implicates a certain clinical parameter, system list, or table (e.g., the word "identity" always implicates the parameter IDENTITY, which means the identity of an organism).
3. The word *might* implicate a certain parameter, system list, or table (e.g., the word "positive" might implicate the parameter NUMPOS, which means the number of positive cultures in a series).
4. The word is part of a phrase that can be thought of as a single word (examples of such phrases are "transtracheal aspiration," "how long," and "not sterile").

is used in a question. It is from this list that terminal words' "implication" pointers (Properties 2 and 3 in Table 18-1) are generated.

During the first phase of parsing, each word in the original text is replaced by its terminal word. For words not found in the dictionary, the system uses Winograd's root-extraction algorithm (Winograd, 1972) to see if the word's lexical root is in the dictionary (e.g., the root of "decision" is "decide"). If so, the word is replaced by the terminal word for its root. Words still unrecognized after root extraction are left unchanged.

The resulting list of terminal and unrecognized words is then passed to a function that recognizes phrases. Using Property 4 (Table 18-1), the function identifies a phrase and replaces it with a single synonymous terminal word (whose dictionary properties may be important in determining the meaning of the question).

Step 2: Classifying the Question

The next step is to classify the question so that the program can tell which specialist should answer it. Since all questions about the consultation must be about some specific context, the system requires that the name of the context (e.g., ORGANISM-1) be stated explicitly. This provides an easy mechanism to separate general questions about the knowledge base from questions about a particular consultation.

Further classification is done through a pattern-matching approach similar to that used by Colby et al. (1974). The list of words created by the first phase is tested against a number of patterns (about 50 at present). Each pattern has a list of actions to be taken if the pattern is matched. These actions set flags that indicate what type of question was asked. In the case of questions about judgmental knowledge (called *rule-retrieval questions*), pattern matching also divides the question into the part referring to the rule's premise and the part referring to its action. For example, in "How do you decide that an organism is streptococcus?" there is no premise part, and the action part is "an organism is streptococcus"; in "Do you ever use the site of the culture to determine an organism's identity?" the premise part is "the site of the culture" and the action part is "an organism's identity."

Steps 3 and 4: Determining What Pieces of Knowledge Are Relevant

The classification of a question guides its further analysis. Each question type has an associated template with blanks to be filled in from the question. The different blanks and the techniques for filling them in are listed in Table 18-2. With the question correctly classified, the general question

TABLE 18-2 Mechanisms for Analyzing a Question

Slot	<i>Analysis cues for filling a slot</i>
<cntxt>	The context must be mentioned by name, e.g., ORGANISM-2.
<rule>	Either a rule's name (RULE047) will be mentioned or the word "rule" will appear, together with the rule's number (47).
<value>	One of the terminal words in the question has a dictionary property indicating that it is a legal value for the parameter (Property 1, Table 18-1), e.g., THROAT is a legal value for the parameter SITE.
<parm>	All of the words in the list are examined to see if they implicate any clinical parameters. Strong implications come from words with properties showing that the word is an acceptable value of the parameter, or that the word <i>always</i> implicates that parameter (Properties 1 and 2, Table 18-1). Weak implications come from words with properties showing that they <i>might</i> implicate the parameter (Property 3, Table 18-1). The system uses an empirical scoring mechanism for picking out only the most likely parameters. Associated with certain parameters are words or patterns that <i>must</i> appear in the question in order for the parameter to be implicated. This scheme allows the system to distinguish among related parameters that may be implicated by the same keywords in the first pass. For example, the word "PMN" implicates parameters CSFPOLY (the percent of PMN's in the CSF) and PMN (the percent of PMN's in the complete blood count). These are distinguished by requiring that the word "CSF" be present in a question in order for CSFPOLY to be implicated.
<list>	System lists are indicated in a manner similar to that for parameters, except that scoring is not done. Lists, like parameters, may have associated patterns that must be present in the question. Furthermore, lists have properties telling which other system lists are their subsets. If a question implicates both a list and a subset of that list, the more general (larger) list is discarded. As an example, the question "Which drugs are aminoglycosides?" implicates two lists: the list of all drugs, and the list of drugs that are aminoglycosides. The system only considers the more specific list of aminoglycosides when answering the question.
<table>	Tables are indicated in a manner similar to that for lists except that an entry in the table must also be present in the question. For example, the word "organism" may indicate two tables: one containing a classification of organisms, and the other containing normal flora of various portals. The question "What organisms are considered to be subtypes of <i>Pseudomonas</i> ?" will correctly implicate the former table, and "What are the organisms likely to be found in the throat?" will implicate the latter, because PSEUDOMONAS is in the first table and THROAT is in the second.

answerer can tell which specialist should answer it. Filling in all blanks in the template gives the specialist all the information needed to find the answer.

Step 5: Answering the Question

Corresponding to each question type, there are a number of possible answer templates. For example, for questions of the form “How do you know the value of <parm> of <cntxt>?” two of the answer templates are

I used <rule> to conclude that <parm> of <cntxt> is <value>.
 This gave a cumulative CF of <certainty factor>.
 The last question asked before the conclusion was made
 was <question number>.

In answer to question <question number> you said that <parm> of
 <cntxt> is <value>.

The specialist for answering questions of a given type has to check the history tree or the system’s knowledge base in order to determine which of the answer templates is appropriate for a particular question. Some blanks in the answer template are filled in by the same items as are used to fill blanks in the question template. The remainder are filled by the specialist with the information that answers the question. In the above example, the slots <parm>, <cntxt>, and possibly <value> would be filled in from the question, and the other slots would be filled from the history tree.

18.2.4 Questions About MYCIN’s Static Knowledge

General questions about MYCIN’s factual knowledge (Figure 18-9) are the easiest to answer. The specialist that provides answers to such questions must look up the desired information in the system’s static knowledge base. Generally, an answer consists of simply displaying this information. The most processing occurs when checking the retrieved information against some item in the question for equality or set membership.

As was discussed in the previous section, the specialist that answers questions about judgmental knowledge is more complicated (Figure 18-8). Answering such questions (Figure 18-10) involves built-in knowledge about the rule set, plus the ability to “read” the rules. By the time the question has been analyzed, the specialist knows which parameters must appear in the premise and which must appear in the action of any rule that answers the question. Values may be specified for any of the parameters. To answer the question, the rule-retrieval specialist must first find every rule whose

****IS BLOOD A STERILE SITE?**

Yes. Blood is one of those sites that are normally sterile.

**** WHAT ARE THE NONSTERILE SITES?**

Those sites that are normally nonsterile are: CERVIX,
CUTANEOUS-ULCER, LOCHIA, NOSE, SKIN, STOOL, THROAT, URETHRA,
VAGINA.

**** WHAT ORGANISMS ARE LIKELY TO BE FOUND IN THE THROAT?**

The likely pathogens associated with throat are: STREPTOCOCCUS-
PNEUMONIAE, STREPTOCOCCUS-GROUP-A, NEISSERIA-MENINGITIDIS.

**** IS BACTEROIDES AEROBIC?**

No, the aerobicity of bacteroides is anaerobic.

**** WHAT DOSAGE OF STREPTOMYCIN DO YOU GENERALLY RECOMMEND?**

The suggested dosage and route of streptomycin is 7.5 mg/kg
q12h IM (or IV).

**** HOW DO YOU TREAT MENINGOCOCCAL BACTEREMIA?**

For treatment of NEISSERIA-MENINGITIDIS in PRIMARY-BACTEREMIA

Select therapy from among the following drugs (ranked according
to relative efficacy, assuming uniform sensitivity):

PENICILLIN	(.91)
AMPICILLIN	(.9)
SULFISOXAZOLE	(.8)
CHLORAMPHENICOL	(.7)
ERYTHROMYCIN	(.5)
TETRACYCLINE	(.4)

FIGURE 18-9 Questions about factual knowledge.

premise and action satisfy these constraints. It accordingly needs to know that there are two special lists associated with each parameter: one containing every rule that uses the parameter in its premise, and the other containing every rule that concludes about the parameter in its action (LOOKAHEAD and UPDATED-BY, see Chapter 5). Using these lists for the various parameters mentioned in the question, the specialist can find those rules that might answer the question. If no values were specified, the job is done and the relevant rules can be displayed without further analysis; otherwise, it is necessary to read each of the rules in the list and to eliminate those that do not mention the correct values for the parameter.

The rule-retrieval specialist also makes use of a piece of MYCIN's knowledge discussed in Chapter 9. The system contains models of its own knowledge (called *rule models*), which are used primarily during acquisition of new medical knowledge from an expert (Davis, 1976). These models, however, can be put to many uses—one is to explain general patterns in decision making. The rule models are abstract descriptions of subsets of rules and are generated automatically by reading the rules. For example, the model for IDENT IS PSEUDOMONAS tells what features are common to the majority of rules that conclude that the identity of an organism is *Pseudomonas*.

If a model exists describing the rules about which the question is ask-

**** WHY DO YOU ASK WHETHER THE PATIENT HAS A FEVER OF UNKNOWN ORIGIN?**

The rules listed below use:

whether the patient has a true fever of unknown origin

RULE180

- IF: 1) The site of the culture is blood, and
 2) The number of cultures in the series including the culture is greater than or equal to 1, and
 3) The number of cultures in this series which were positive for the organism is less than or equal to 1, and
 4) The patient has a true fever of unknown origin, and
 5) Cardiac-surgery is a relevant item from the history of the patient

Then: There is suggestive evidence (.6) that the infection is infective-endocarditis

**** WHAT DOES GU MANIPULATION TELL YOU ABOUT THE IDENTITY OF AN ORGANISM?**

The rules listed below use:

whether the patient has had a genito-urinary manipulative procedure

to conclude about:

the identity of the organism

156, 163, 190

Which do you wish to see?

**** 156**

RULE156

- IF: 1) The site of the culture is blood, and
 2) The gram stain of the organism is gramneg, and
 3) The morphology of the organism is rod, and
 4) The portal of entry of the organism is urine, and
 5) The patient has not had a genito-urinary manipulative procedure, and
 6) Cystitis is not a problem for which the patient has been treated

THEN: There is suggestive evidence (.6) that the identity of the organism is e.coli

FIGURE 18-10 Rule-retrieval questions.

ing, the rule-retrieval specialist incorporates this model's information into its answer (Figure 18-11). Thus the question-answering facility is able to give some information about strategies for achieving some of the system's goals, as well as giving the individual rules that use the strategies.

18.2.5 Consultation-Specific Questions

Some of the simplest questions about a specific consultation are those that inquire about the value of a particular parameter (Figure 18-12). The specialist that answers these questions must know how to retrieve this in-

**** HOW DO YOU DECIDE THAT AN ORGANISM MIGHT BE PSEUDOMONAS AERUGINOSA?**

Rules which conclude that the identity of the organism is pseudomonas-aeruginosa generally use one or more of the following pieces of information:

- the site of the culture
- the gram stain of the organism
- the morphology of the organism

Furthermore, the following relationships hold:

The gram stain of the organism, and the morphology of the organism tend to appear together in these rules.

RULE184, RULE116, RULE047, RULE085, RULE040 conclude that the identity of the organism is pseudomonas-aeruginosa.

Which of these do you wish to see?

**** 184**

RULE184

- IF: 1) The category of the organism is not known, and
 2) The gram stain of the organism is gramneg, and
 3) The morphology of the organism is rod, and
 4) The aerobicity of the organism is facul

THEN: There is weakly suggestive evidence (.1) that the identity of the organism is pseudomonas-aeruginosa

FIGURE 18-11 Response that uses rule models.

formation. In some cases it will also be necessary to make comparisons to see if a value specified in the question is one of the values that was inferred for that parameter.

Answering other types of consultation-specific questions involves the use of knowledge about the control structure:

- how the system acquires information,
 - why the system tries to find out about parameters,
 - how information is used once it is acquired,
 - what causes a rule to be tried, and
 - what causes a rule to fail.
-

**** TO WHAT CLASS DOES ORGANISM-1 BELONG?**

The category of ORGANISM-1 is enterobacteriaceae (.9)

**** IS ORGANISM-1 A STREPTOCOCCUS?**

No.

The identity of ORGANISM-1 is e.coli (.74) bacteroides (.7)

klebsiella-pneumoniae (.4) pseudomonas-aeruginosa (.22)

enterobacter (.21) proteus-non-mirabilis (.19) serratia (.03)

salmonella (.02)

FIGURE 18-12 Questions about a parameter's value.

**** HOW DO YOU KNOW THAT CULTURE-1 WAS FROM A STERILE SOURCE?**

I used rule189 to conclude that this blood culture was taken from a sterile source. This gave a cumulative CF of (1.0). The last question asked before the conclusion was made was 7.

**** DID YOU CONSIDER BACTEROIDES AS A POSSIBILITY FOR ORGANISM-1?**

Yes. I used rule095 to conclude that the identity of ORGANISM-1 is bacteroides. This gave a cumulative CF of (.7). The last question asked before the conclusion was made was 20.

**** HOW DO YOU KNOW THAT ORGANISM-1 IS E.COLI?**

The following rules made conclusions about whether the identity of ORGANISM-1 is e.coli

Rule	cumulative certainty		last question asked before conclusion was made
	Yes	No	
RULE021	(.47)		20
RULE084	(.55)		22
RULE003	(.74)		24

In answer to question 9 you said that the identity of ORGANISM-1 is e.coli (.3)

FIGURE 18-13 Questions regarding how a conclusion was made.

The specialist for answering questions like “How do you know the value of <parm> of <cntxt>?” knows that the value of a parameter can come from two sources: it can be deduced by rules, or the user can give it in response to a question. The history tree will show which (possibly both) of these sources provided the particular information mentioned in the question (Figure 18-13).

If the question is phrased in the negative, it is necessary first to find all the ways the conclusion *could* have been made (this is a simple task of rule retrieval), then to explain why it *wasn't* made in this consultation (Figure 18-14). The specialist for answering these questions must know what situations can prevent conclusions from being made. The second question in Figure 18-14 illustrates how the answer to one question might cause another question to be asked.

The specialist for answering questions of the form “How did you use <parm> of <cntxt>?” needs to know not only how to find the specific rules that might use a parameter, but also how a parameter can cause a rule to fail and how one parameter can prevent another from being used. The history tree can be checked to see which of the relevant rules used the parameter, which failed because of the parameter, and which failed for some other reason, preventing the parameter from being used (Figure 18-15).

For questions of the form “Why didn't you find out about <parm> of <cntxt>?” general knowledge of MYCIN's control structure tells the conditions under which it would have been necessary to find out some piece of information. The record of the consultation can be used to de-

**** WHY DID YOU RULE OUT STREPTOCOCCUS AS A POSSIBILITY FOR ORGANISM-1?**

The following rules could have been used to determine that the identity of ORGANISM-1 is streptococcus: RULE033. However, none of these succeeded in the context of ORGANISM-1. If you would like an explanation for why any of these rules failed, please enter their numbers:

**** 33**

Clause 2 of RULE033 ["the morphology of the organism is coccus"] was already known to be false for ORGANISM-1, so the rule was never tried.

**** WHY DON'T YOU THINK THAT THE MORPHOLOGY OF ORGANISM-1 IS COCCUS?**

It is definite that the morphology of ORGANISM-1 is rod. Knowing this with certainty rules out all other values for the morphology of ORGANISM-1, including coccus.

FIGURE 18-14 Questions regarding why a conclusion was not made.

**** HOW DID YOU USE THE AEROBICITY OF ORGANISM-1?**

The aerobicity of ORGANISM-1 was used in the following rules: RULE003, RULE005, RULE084.

The aerobicity of ORGANISM-1 caused the following rules to fail: RULE035, RULE051, RULE052, RULE053, RULE111.

The aerobicity of ORGANISM-1 also would have been used in: RULE037, RULE050, RULE058, RULE086, RULE110, RULE184, RULE203, RULE204, RULE205. However, none of these succeeded in the context of ORGANISM-1. If you would like an explanation for why any of these rules failed, please enter their numbers:

**** 37**

RULE037 was tried in the context of ORGANISM-1, but it failed due to clause 1 ["the category of the organism is not known"].

**** DID YOU CONSIDER THE FACT THAT PATIENT-1 IS NOT A COMPROMISED HOST?**

The fact that Fred Jones is not a compromised host caused the following rules to fail: RULE085, RULE106.

The fact that Fred Jones is not a compromised host also would have been used in: RULE109. However, none of these succeeded in the context of PATIENT-1. If you would like an explanation for why any of these rules failed, please enter their numbers:

**** NONE**

FIGURE 18-15 Questions regarding how information was used.

termine why these conditions never arose for the particular parameter in question (Figure 18-16). Figure 18-16 also illustrates that MYCIN's general question answerer allows a user to get as much information as is desired. The first answer given was not really complete in itself, but it led the user to ask another question to get more information. Then another question was asked to determine why clause 1 of Rule 159 was false. The answers to the first two questions both mentioned rules, which could be printed if the user wanted to examine them. The special command PR (Print Rule) is for the user's convenience. It requires no natural language processing and thus can be understood and answered immediately ("What is Rule 109?" or "Print Rule 109" also would be understood).

In questions that ask about the application of a rule to a context there are three possibilities: the rule told us something about the context; the rule failed when applied to that context; or the rule was never tried in that context. The history tree tells which of these is the case. Furthermore, if a rule succeeded, there is a record of all the conclusions it made, and if it failed, the reason for failure is recorded. As Figure 18-17 illustrates, rules are only applicable in certain contexts. If the rule could not apply to the context mentioned in the question but might be relevant when applied to some other context (which is an ancestor or descendent in the context tree), then the question is answered as if it had mentioned the relevant context.

There are interactions among the different types of questions, but they weren't mentioned earlier for the sake of clarity. Sometimes a question may indicate an incorrect assumption about the state of MYCIN's knowledge. For example, a question could be "Why don't you think that ORGANISM-1 is *E. coli*?" when, in fact, the system has concluded that the organism is *E. coli*. To answer this question, the explanation system would explain how it *did* make such a decision, despite the phrasing of the question. Similarly, when asked how the system used (or concluded about) some parameter that was never traced, the system explains why it didn't need to find out about the parameter, regardless of how the question was phrased.

18.3 Conclusions

Consultation systems form a class of artificial intelligence programs that can provide useful solutions to real-world problems. The utility of such systems, however, depends on their acceptability to human users. One feature that can increase their acceptability is a mechanism whereby they can explain or justify their advice.

The development of an explanation mechanism for a consultation system is very much related to the problems of representing knowledge and of making use of different sources of knowledge. Since the use of inference rules provides a unified way to represent modular pieces of knowledge,

**** DID YOU FIND OUT ABOUT THE CBC ASSOCIATED WITH CULTURE-1?**

During the preceding consultation, I never had reason to try to find out whether the results of the CBC from the time period of this blood culture are available. It would have been necessary to know this information in order to use any of the following: RULE109. However, none of these succeeded in the context of this blood culture. If you would like an explanation for why any of these rules failed, please enter their numbers:

**** 109**

RULE109 would be relevant to this blood culture when applied to: ORGANISM-1. I never had reason to try to apply RULE109 to ORGANISM-1. I would have needed to use RULE109 in order to find out whether ORGANISM-1 is a contaminant. However, during the preceding consultation, I never had reason to try to find out whether ORGANISM-1 is a contaminant.

**** WHY DIDN'T YOU NEED TO KNOW WHETHER ORGANISM-1 IS A CONTAMINANT?**

During the preceding consultation, I never had reason to try to find out whether ORGANISM-1 is a contaminant. It would have been necessary to know this information in order to use any of the following: RULE006, RULE007, RULE106, RULE108, RULE109, RULE159. However, none of these succeeded in the context of ORGANISM-1. If you would like an explanation for why any of these rules failed, please enter their numbers:

**** 159**

RULE159 was tried in the context of ORGANISM-1, but it failed due to clause 1 ["it is suspected that the identity of the organism is corynebacterium-non-diphtheriae"]

**** IS ORGANISM-1 CORYNEBACTERIUM-NON-DIPHtherIAE?**

No. The identity of ORGANISM-1 is e.coli (.74) bacteroides (.7) klebsiella-pneumoniae (.4) pseudomonas-aeruginosa (.22) enterobacter (.21) proteus-non-mirabilis (.19) serratia (.03) salmonella (.02)

**** PR 109**

RULE109

IF: 1) The culture was taken from a sterile source, and
 2) It is not known whether the organism is a contaminant, and
 3) The patient is not a compromised host, and
 4) The patient was febrile due to the infection, and
 5) The results of the CBC from the time period of the culture are available, and
 6) A - The white count from the CBC is greater than 10.5, or
 B - The percent of PMN's in the CBC at the time the culture was obtained is greater than 78, or
 C - The percent of WBC's which were bands in the CBC at the time the culture was obtained is greater than 10

THEN: There is strongly suggestive evidence (.8) that the organism is not a contaminant

FIGURE 18-16 Questions about why a parameter was not traced.

**** DID RULE 116 TELL YOU ANYTHING ABOUT INFECTION-1?**

RULE116 would be relevant to the primary-bacteremia when applied to: ORGANISM-1.

Clause 5 of RULE116 ["the infection is meningitis"] was already known to be false for ORGANISM-1, so the rule was never tried.

**** HOW WAS RULE 178 HELPFUL WHEN YOU WERE CONSIDERING ORGANISM-1?**

RULE178 would be relevant to ORGANISM-1 when applied to: CULTURE-1.

I never had reason to try to apply RULE178 to this blood culture. I would have needed to use RULE178 in order to find out whether this blood culture was taken from a sterile source. However, I was already certain that this blood culture was taken from a sterile source.

**** WHAT DID RULE 295 TELL YOU ABOUT ORGANISM-1?**

I used RULE295 to conclude that the identity of ORGANISM-1 is hemophilus-influenzae. This gave a cumulative CF of (.25). The last question asked before the conclusion was made was 36.

**** WHY DIDN'T YOU USE RULE 112 TO FIND OUT ABOUT ORGANISM-1?**

RULE112 was not executed because it would have caused circular reasoning when applied to ORGANISM-1. Would you like to see the chain of rules and parameters which makes up this circle?

**** YES**

I wanted to know about the identity of ORGANISM-1 because I try to find out the identity of the organism for all current organisms of the patient.

To find out about the identity of ORGANISM-1, I tried to use RULE021. Before I could use RULE021, I needed to know about a prior organism with possibly the same identity as ORGANISM-1.

To find out about a prior organism with possibly the same identity as ORGANISM-1, I tried to use RULE005. Before I could use RULE005, I needed to know about the aerobicity of ORGANISM-1.

To find out about the aerobicity of ORGANISM-1, I tried to use RULE031. Before I could use RULE031, I needed to know about the category of ORGANISM-1.

To find out about the category of ORGANISM-1, I tried to use RULE112. Before I could use RULE112, I needed to know about the identity of ORGANISM-1.

But this is the unknown parameter I sought originally.

FIGURE 18-17 Questions regarding the application of rules.

the task of designing an explanation capability is simplified for rule-based consultation systems. The example of MYCIN shows how this can be done and illustrates further that a system designed for a single domain with a small, technical vocabulary can give comprehensive answers to a wide range of questions without sophisticated natural language processing.