

29

Extensions to Rules for Explanation and Tutoring

William J. Clancey

As described in Part Eight, the success of MYCIN as a problem solver suggested that the program's knowledge base might be a suitable source of subject material for teaching students. This use of MYCIN was consistent with the design goals that the program's explanations be educational to naive users and that the representation be flexible enough to allow for use of the rules outside of the consultative setting. In theory, the rules acquired from human experts would be understandable and useful to students. The GUIDON program discussed in Chapter 26 was developed to push these assumptions by using the rules in a tutorial interaction with medical students.

In attempting to "transfer back" the experts' knowledge to students through GUIDON, we found that the experts' diagnostic approach and understanding of rules were not explicitly represented. GUIDON cannot justify the rules because MYCIN does not have an encoding of how the concepts in a rule fit together. GUIDON cannot fully articulate MYCIN's problem-solving approach because the structure of the search space and the strategy for traversing it are implicit in the ordering of rule concepts. Thus the seemingly straightforward task of converting a knowledge-based system into a computer-aided instruction program has led to a detailed reexamination of the rule base and the foundations on which rules are constructed, an epistemological study.

In building MYCIN, rule authors did not recognize a need to record the structured way in which they were fitting rule parts together. The rules are more than simple associations between data and hypotheses. Sometimes clause order counts for everything, and different orders can mean differ-

ent things. Also, some rules are present mostly to control the invocation of others. The uniformity of the representation obscures these various functions of clauses and rules. In looking beyond the surface of the rule representation to make explicit the intent of the rule authors, this paper has a purpose similar to Woods' "What's in a Link?" (1975) and Brachman's "What's in a Concept?" (1976). We ask, "What's in a Rule?"

In building GUIDON, we thought that we were simply being "applications engineers" by making use of MYCIN's explanation facility for a tutorial setting. As noted in Chapter 26, it was surprising to find out how little the explanation facility could accomplish for a student. Without a crisp characterization of what we expected an explanation to convey, the program was of questionable tutorial value. On the positive side, the study of these shortcomings led to a radical change in our conception of MYCIN's rules and supplied a new epistemological framework for building expert systems.

In this chapter we provide a review of MYCIN's explanatory capability and an overview of an epistemological framework for enhancing that capability. The following two sections examine in detail the problems of *justifying a rule* and *explaining an approach*, thereby elucidating the *support* and *strategic* aspects of the epistemological framework. Implications for performance of a consultation system and modifiability are considered briefly. Finally, in the last section, the framework is used to analyze other expert systems.

29.1 MYCIN's Explanation Capability

As already mentioned, MYCIN was designed so that it could explain its reasoning to the user. This means that it can describe its reasoning steps: how a request for data is related to a goal, how one goal leads to another, and how a goal is achieved. Figure 29-1 shows a consultation excerpt in which the program demonstrates a line of reasoning (chained goals) that lead backwards from the current question:

Is the patient a compromised host (Rule 343)	→	Is the patient immunosuppressed? ¹
(Rule 395)	→	Is the patient receiving steroids?
(Rule 543)	→	Is <i>E. coli</i> causing the meningitis?

¹Read as "... compromised host (by Rule 343) answers the question 'Is the patient immunosuppressed?' ..."

23) Is J.Smith a compromised host?
**** WHY**

[i.e. WHY is it important to determine whether J.Smith is a compromised host?]

[3.0] This will aid in determining whether J.Smith is immunosuppressed.
 If
 [3.1] J.Smith is not a compromised host
 then
 it is definite (1.0) that J.Smith is not immunosuppressed
 [RULE343]

[back to question 23 . . .]
**** WHY**

[i.e. WHY is it important to determine whether J.Smith is immunosuppressed?]

[4.0] . . . in order to determine whether J.Smith has received corticosteroids.
 If
 [4.1] J.Smith is not immunosuppressed
 then
 it is definite (1.0) that J.Smith has not received corticosteroids.
 [RULE395]

[back to question 23 . . .]
**** WHY**

[i.e. WHY is it important to determine whether J.Smith has received corticosteroids?]

[5.0] . . . in order to determine the organisms causing the infection.
 It has already been established that
 [5.1] the infection which requires therapy is meningitis, and
 [5.2] only circumstantial evidence is available for this case, and
 [5.3] the type of meningitis is bacterial.

Therefore, if
 [5.4] J.Smith is receiving corticosteroids
 then
 there is evidence that the organisms causing the infection are:
 e.coli (.4), klebsiella-pneumoniae (.2), or pseudomonas-aeruginosa (.1)
 [RULE543]

[back to question 23 . . .]
**** NO**

FIGURE 29-1 Consultation excerpt showing explanation capability. (User input in boldface capitals; all other text is printed by MYCIN.)

Figure 29-2 illustrates how, in the questioning session after the consultation, one can inquire further about the program's intermediate reasoning steps, including why it didn't ask about something. These are the explanation capabilities that we sought to exploit in a teaching program.

MYCIN's explanations are entirely in terms of its rules and goals. The question **WHY** means "Why do you want this information?" or "How is this information useful?" and is translated internally as "In what rule does this goal appear, and what goal does the rule conclude about?" Davis, who

developed the explanation facility, pointed out that MYCIN did not have the knowledge to respond to other interpretations of a WHY question (Davis, 1976). He mentioned specifically the lack of rule justifications and planning knowledge addressed in this chapter.

In order to illustrate other meanings for the question WHY in MYCIN, we illustrate the rule set as a network of goals, rules, and hypotheses in Figure 29-3. At the top level are all of the system's *goals* that it might want to pursue to solve a problem (diagnostic and therapeutic decisions). Examples of goals, stated as questions to answer, are "What is the shape of the organism?" and "What organism is causing the meningitis?" At the second level are *hypotheses* or possible choices for each of the goals. Examples of hypotheses are "The organism is a rod." and "*E. coli* is causing the meningitis." At the third level are the *rules* that support each hypothesis. At the fourth level appear the *premises* of these rules, specific hypotheses that must be believed for the rule to apply. For example, for Rule 543 to apply (shown in Figure 29-1) it must be the case that the infection is meningitis, that the meningitis was caused by bacteria, that the patient is receiving steroids, and so on.

A key aspect of MYCIN's interpreter is that, when confronted with a hypothesis in a rule premise that it needs to confirm, it considers all related hypotheses by pursuing the more general goal. For example, attempting to apply Rule 543, the program will consider all rules that conclude about the infection, rather than just those that conclude that the infection is meningitis. Similarly, it will consider all rules that conclude about the kind of meningitis (viral, fungal, TB, or bacterial), rather than just those that hypothesize that the meningitis is bacterial.² These new goals deriving from rules can now be seen conceptually as level 1 goals, and the process recurs.

The links in Figure 29-3 and their ordering are points of flexibility in the rule representation. For example, the author of a rule defines each goal and its specific hypotheses (levels 1 and 2 and 4 and 5). Less trivially, it is the author's choice to define rules that link hypotheses to one another. For example, rules on level 3 link levels 2 and 4. We call the rationale behind this link the *justification of the rule*. GUIDON cannot teach rule justifications because they are not represented in MYCIN. Section 29.3 examines the nature of rule justifications and how a tutoring system can provide them.

²This is not inefficient, given the program's exhaustive search strategy and the fact that the other hypotheses will be referenced by other rules. Note also that some hypotheses, such as "the patient is receiving steroids," are not generalized but are represented as goals directly. Whether or not a hypothesis is represented as a yes/no parameter or as a "value" of a multi-valued parameter (such as "kind of meningitis") is a decision of the author of the rule, deriving from a pattern of hypotheses that he or she wishes to collapse for clarity into a more general goal. By this process of abstraction, a single multi-valued parameter dealing with kinds of surgery would replace individual yes/no parameters that specified "cardiac surgery," "neurosurgery," etc. These organizational decisions have no bearing on system performance, so the knowledge base is somewhat inconsistent in how these choices are made.

**** DID YOU CONSIDER WHETHER THE PATIENT WAS RECEIVING STEROIDS?**

RULE543 could have used:

whether J.Smith has received corticosteroids.

Clause 4 of RULE543 "the patient has received corticosteroids" was already known to be false.

**** HOW DID YOU KNOW THAT THE PATIENT HAD NOT RECEIVED STEROIDS?**

RULE395 was used to conclude that J.Smith has not received corticosteroids.

The last question asked before the conclusion was made was 23.

**** WHAT WAS QUESTION 23?**

Question 23 was asked in order to find out whether J.Smith is a compromised host in an effort to execute RULE343.

FIGURE 29-2 Excerpt from a question/answer session. (User input appears in boldface capitals.)

Next, the author's ordering of hypotheses in a rule's premise will affect the order in which goals are pursued (level 5). The rationale for this choice again lies outside of the rule network. Thus the program cannot explain why it pursues meningitis (goal 5.1 in Figure 29-1) *before* determining that the infection is bacterial (goal 5.3). Section 29.4 examines how this ordering constitutes a strategy and how it can be made explicit.

The order in which rules for a goal are tried (level 3) also affects the order in which hypotheses (and hence subgoals) are pursued (level 5). For example, Rule 535 considers whether the patient is an alcoholic; so if this rule is tried before Rule 543, alcoholism will be considered before steroids. As these goals cause questions to be asked of the user, it is evident that the ordering of questions is also determined by the ordering of rules as well as by the ordering of clauses in the premise of a rule.

Here there is no implicit author rationale, for rule order lies outside of the author's choice; it is fixed, and determined only by the order in which rules were entered into the system. As pointed out above, MYCIN does not *decide* to pursue the hypothesis "bacterial meningitis" before "viral meningitis"—it simply picks up the bag of rules that make some conclusion about "kind of meningitis" and tries them in numeric order. Hence rule order is the answer to the question "Why is one hypothesis considered before another?" And rule order is often the answer to "Why is one question asked before another?" Focusing on a hypothesis and choosing a question to confirm a hypothesis are not necessarily arbitrary in human reasoning. This raises serious questions about using MYCIN for interpreting a student's behavior and teaching him or her how to reason, as discussed in Section 29.4.³

³Meta-rules could have been used for ordering rules, as described in Chapter 28. The present chapter is a rethinking of the whole question.

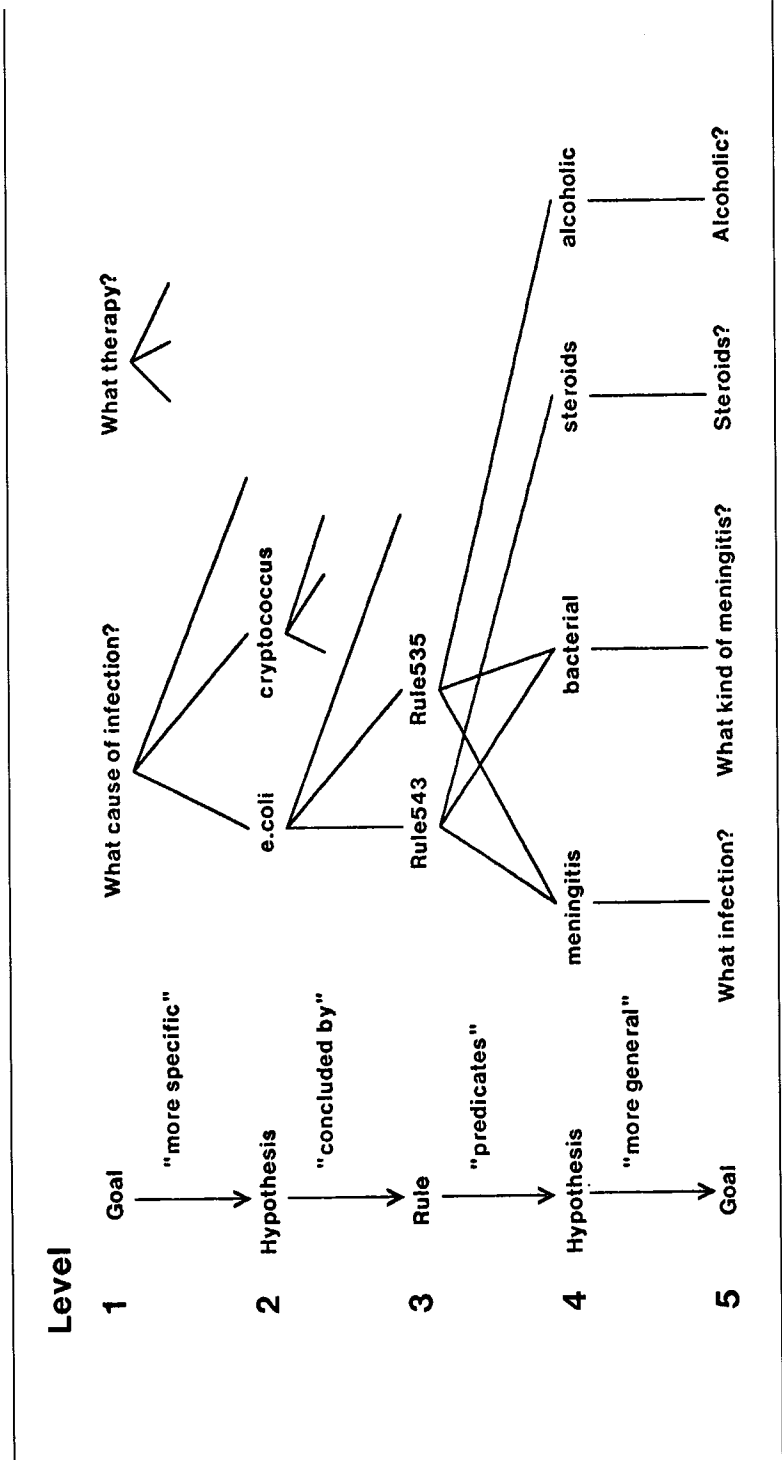


FIGURE 29-3 Rule set shown as a network linking hypotheses and goals.

To summarize, we have used a rule network as a device for illustrating aspects of MYCIN's behavior that it cannot explain. We are especially interested in making explicit the knowledge that lies behind the behavior that is not arbitrary but that cannot be explained because it is implicit in rule design. To do this, we will need some sort of framework for characterizing the knowledge involved, since the rule link itself is not sufficient. An epistemological framework for understanding MYCIN's rules is presented in the next section.

29.2 An Epistemological Framework for Rule-Based Systems

The framework presented in this section stems from an extensive study of MYCIN's rules. It is the basic framework that we have used for understanding physicians' explanations of their reasoning, as well as being a foundation for re-representing the knowledge in MYCIN's rules. As an illustration, we will consider in detail the steroids rule shown again in Figure 29-4.⁴

RULE543

- IF: 1) The infection which requires therapy is meningitis,
2) Only circumstantial evidence is available for this case,
3) The type of the infection is bacterial,
4) The patient is receiving corticosteroids,
THEN: There is evidence that the organisms which might be causing the infection are
e.coli (.4), klebsiella-pneumoniae (.2), or pseudomonas-aeruginosa (.1)
-

FIGURE 29-4 The steroids rule.

Figure 29-5 shows how this diagnostic heuristic is justified and incorporated in a problem-solving approach by relating it to strategic, structural, and support knowledge. Recalling Section 29.1, we use the term *strategy* to refer to a plan by which goals and hypotheses are ordered in problem solving. A decision to determine "cause of the infection" before "therapy to administer" is a strategic decision. Similarly, it is a strategic decision to pursue the hypothesis "*E. coli* is causing meningitis" before "*Cryptococcus* is causing meningitis." And recalling an earlier example, deliberately deciding to ask the user about steroids before alcoholism would be a strategic decision. These decisions all lie above the plane of goals and hypotheses,

⁴The English form of rules stated in this paper has been simplified for readability. Sometimes clauses are omitted. Medical examples are for purposes of illustration only.

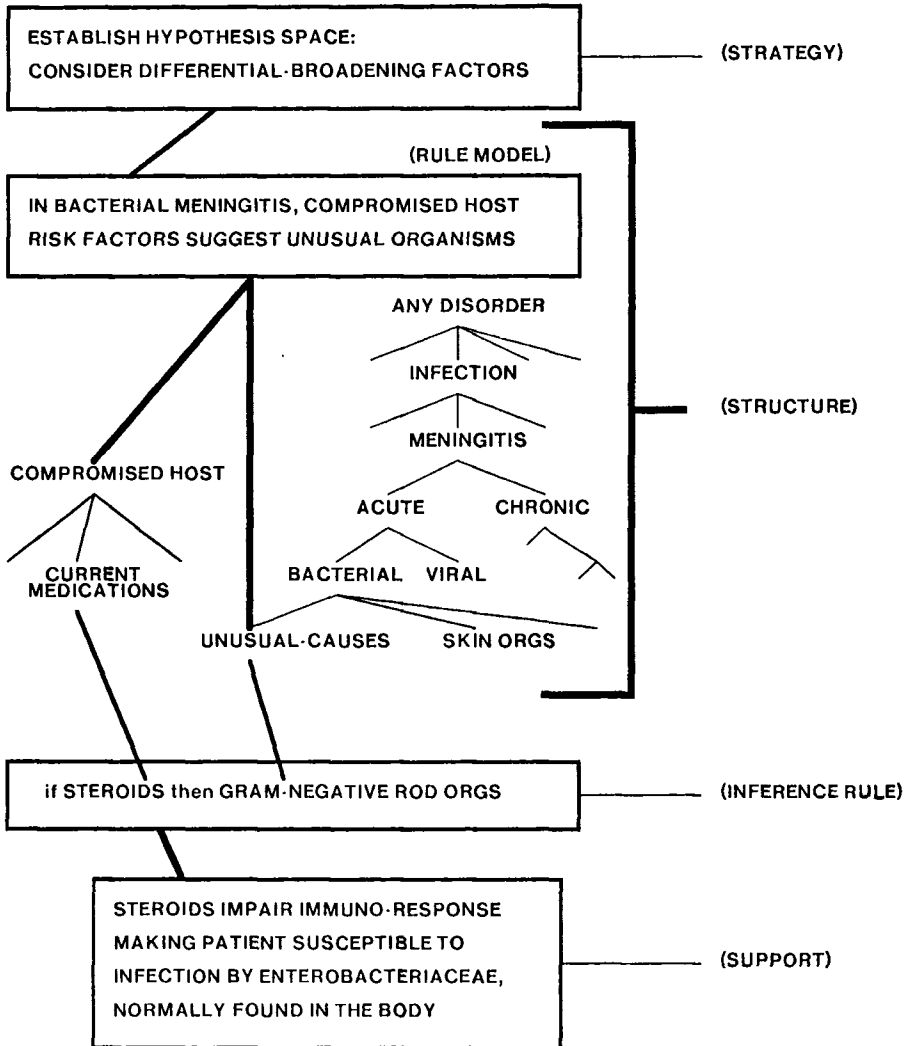


FIGURE 29-5 Augmenting a knowledge source with three kinds of meta-level knowledge: knowledge for indexing, justifying, and invoking a MYCIN rule.

and as discussed later, they can often be stated in domain-independent terms, e.g., “consider differential-broadening factors.”

In order to make contact with the knowledge of the domain, a level of *structural knowledge* is necessary. Structural knowledge consists of abstractions that are used to index the domain knowledge. For example, one can

classify causes of disease into common and unusual causes, for example, of bacterial meningitis. These concepts provide a *handle* by which a strategy can be applied, a means of referencing the domain-specific knowledge. For example, a strategy might specify considering common causes of a disease; the structural knowledge about bacterial meningitis allows this strategy to be instantiated in that context. This conception of structural knowledge follows directly from Davis' technique of *content-directed invocation* of knowledge sources (see Chapter 28). A handle is a means of indirect reference and is the key to abstracting reasoning in domain-independent terms. The discussion here elaborates on the nature of handles and their role in the explanation of reasoning.

The structural knowledge we will be considering is used to index two kinds of hypotheses: *problem features*, which describe the problem at hand (for example, whether or not the patient is receiving steroids is a problem feature); and *diagnoses*, which characterize the cause of the observed problem features. For example, acute meningitis is a diagnosis. In general, problem features appear in the premises of diagnostic rules, and diagnoses appear in the conclusions. Thus organizations of problem features and diagnoses provide two ways of indexing rule associations: one can use a strategy that brings certain diagnoses to mind and consider rules that support those hypotheses; or one can use a strategy that brings certain problem features to mind, gather that information, and draw conclusions (apply rules) in a data-directed way.

Figure 29-5 shows how a rule model, or *generalized rule*,⁵ as a form of structural knowledge, enables either data-directed consideration of the steroids rule or hypothesis-directed consideration. Illustrated are partial hierarchies of problem features (compromised host factors) and diagnoses (kinds of infections, meningitis, etc.)—typical forms of structural knowledge. The specific organisms of the steroids rule are replaced by the set “gram-negative rods,” a key hierarchical concept we use for understanding this rule.

Finally, the justification of the steroids rule, a link between the problem feature hypothesis “patient is receiving steroids” and the diagnostic hypothesis “gram-negative rod organisms are causing acute bacterial infectious meningitis,” is based on a causal argument about steroids impairing the body's ability to control organisms that normally reside in the body. While this *support knowledge* is characteristically low-level or narrow in contrast with the strategical justification for considering compromised host risk factors, it still makes interesting contact with structural terms, such as the mention of *Enterobacteriaceae*, which are kinds of gram-negative rod organisms. In the next section, we will consider the nature of rule justifications in more detail, illustrating how structural knowledge enables us to make sense of a rule by tying it to the underlying causal process.

⁵Davis' rule models (Chapter 28), generated automatically, capture patterns, but they do not restate rules more abstractly as we intend here.

29.3 Explaining a Rule

Here we consider the logical bases for rules: what kinds of arguments justify the rules, and what is their relation to a mechanistic model of the domain? We use the terms “explain” and “justify” synonymously, although the sense of “making clear what is not understood” (explain) is intended more than “vindicating, showing to be right or lawful” (justify).

29.3.1 Different Kinds of Justifications

There are four kinds of justifications for MYCIN’s rules: identification, cause, world fact, and domain fact. In order to explain a rule, it is first necessary to know what kind of justification it is based on.

1. Rules that use identifying properties of an object to classify it are called *identification rules*. Most of MYCIN’s rules that use laboratory observations of an unknown are like this: “If the organism is a gram-negative, anaerobic rod, its genus may be bacteroides (.6).” Thus an identification rule is based on the properties of a class.
2. Rules whose premise and action are related by a causal argument are called *causal rules*. The causality can go in either direction in MYCIN rules: “symptom caused by disease” or, more commonly, “prior problem causes disease.” Szolovits and Pauker (1978) suggest that it is possible to subdivide causal rules according to the scientific understanding of the causal link:
 - a. *empirical association* (a correlation for which the process is not understood),
 - b. *complication* (direction of causality is known, but the conditions of the process are not understood), and
 - c. *mechanism* (process is well modeled).

Most of MYCIN’s causal rules represent medical complications that are not easily expressed as anatomical relations and physiological processes. The certainty factors in MYCIN’s causal rules generally represent a mixture of probabilistic and cost/benefit judgment. Rather than simply encoding the strength of association between symptom and cause, a certainty factor also captures how important it is that a diagnosis be considered in therapy selection.

3. Rules that are based on empirical, commonsense knowledge about the world are called *world fact rules*. An example is “If the patient is male, then the patient is not pregnant.” Other examples are based on social patterns of behavior, such as the fact that a young male might be a military recruit and thus be living in a crowded environment where disease spreads readily.

4. *Domain fact rules* link hypotheses on the basis of domain definitions. An example is “If a drug was administered orally and it is poorly absorbed in the GI tract, then the drug was not administered adequately.” By definition, to be administered adequately a drug must be present in the body at high enough dosage levels. By using domain fact rules, the program can relate problem features to one another, reducing the amount of information it has to request from the user.

In summary, a rule link captures class properties, social and domain facts, and probabilistic and cost/benefit judgments. When a definition, property, or world fact is involved, simply saying this provides a reasonable explanation. But causal rules, with their connection to an underlying process of disease, require much more, so we will concentrate on them.

29.3.2 Levels of Explanation—What’s Not in a Rule?

In this section we consider the problem of justifying a causal rule, the tetracycline rule:

“If the patient is less than 8 years old, don’t prescribe tetracycline.”

This rule simply states one of the things that MYCIN needs to know to properly prescribe drugs for youngsters. The rule does not mention the underlying causal process (chelation, or drug deposition in developing bones) and the social ramifications (blackened permanent teeth) on which it is based. From this example, it should be clear that the justifications of MYCIN’s rules lie outside of the rule base. In other words, the record of inference steps that ties premise to action has been left out. A few questions need to be raised here: Did the expert really leave out steps of reasoning? What is a justification for? And what is a good justification?

Frequently, we refer to rules like MYCIN’s as “compiled knowledge.” However, when we ask physicians to justify rules that they believe and follow, they very often can’t explain why the rules are correct. Or their rationalizations are so slow in coming and so tentative that it is clear they are not articulating reasoning steps that are consciously followed. Leaps from data to conclusion are justified because the intermediate steps (like the process of chelation and the social ramifications) generally remain the same from problem to problem. There is no need to step through this knowledge—to express it conditionally in rules. Thus, for the most part, MYCIN’s rules are not compiled in the sense that they represent a deliberate composition of reasoning steps by the rule authors. They are compiled in the sense that they are optimizations that leave out unnecessary steps—evolved patterns of reasoning that cope with the demands of ordinary problems.

If an expert does not think about the reasoning steps that justify a rule, why does a student need to be told about them? One simple reason

tetracycline in youngster
 → chelation of the drug in growing bones
 → teeth discoloration
 → undesirable body change
 → don't administer tetracycline

FIGURE 29-6 Causal knowledge underlying the tetracycline rule.

is so the student can remember the rule. A justification can even serve as memory aid (mnemonic) without being an accurate description of the underlying phenomena. For example, medical students have long been told to think in terms of “bacteria eating glucose” from which they can remember that low CSF (cerebrospinal fluid) glucose is a sign of a bacterial meningitis (as opposed to fungal or viral meningitis). The interpretative rule is learned by analogy to a familiar association (glucose is a food, and bacteria are analogous to larger organisms that eat food). This explanation has been discredited by biological research, but it is still a useful mnemonic.

Given that an accurate causal argument is usually expected, how is a satisfying explanation constructed? To see the difficulty here, observe that, in expanding a rule, there is seemingly no limit to the details that might be included. Imagine expanding the tetracycline rule by introducing three intermediate concepts as shown in Figure 29-6. The choice of intermediate concepts (the grain size of rules) is arbitrary, of course. For example, there is no mention of how the chelation occurs. What are the conditions? What molecules or ions are involved? There are arbitrarily many levels of detail in a causal explanation. To explain a rule, we not only need to know the intermediate steps, we also need to decide which steps in the reasoning need to be explained. Purpose (how deep an understanding is desirable) and prior knowledge are obviously important.

Conceptually, the support knowledge for a causal rule is a tree of rules, where each node is a reasoning step that can theoretically be justified in terms of finer-grained steps. The important thing to remember is that MYCIN is a flat system of rules. It can only state its immediate reasoning steps and cannot explain them on *any* level of detail.

29.3.3 Problem Features, the Hypothesis Taxonomy, and Rule Generalizations

A tree of rules seems unwieldy. Surely most teachers cannot expand on every reasoning step down to the level of the most detailed physical knowledge known. The explanation tree for the tetracycline rule, for example, quickly gets into chemical bonding theory. Explaining a rule (or under-

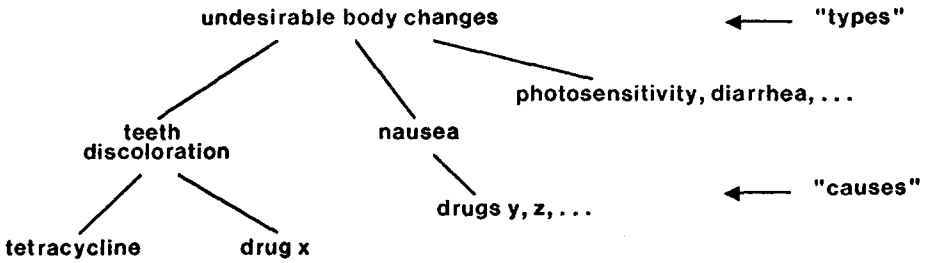


FIGURE 29-7 Problem feature hierarchy for contraindication rules.

standing one) does not require that every detail of causality be considered. Instead, a relatively high level of explanation is generally satisfying—most readers probably feel satisfied by the explanation that tetracycline causes teeth discoloration. This level of satisfaction has something to do with the student's prior knowledge.

For an explanation to be satisfying, it must make contact with already known concepts. We can characterize explanations by studying the kinds of intermediate concepts they use. For example, it is significant that most contraindication rules, reasons for not giving antibiotics, refer to “undesirable body changes.” This pattern is illustrated hierarchically in Figure 29-7. The first level gives types of undesirable changes; the second level gives causes of these types of changes. Notice that this figure contains the last step of the expanded tetracycline rule and a leap from tetracycline to this step. The pattern connecting drugs to the idea of undesirable body changes forms the basis of an expectation for explanations: we will be satisfied if a particular explanation connects to this pattern. In other words, given an effect that we can interpret as an undesirable body change, we will understand why a drug causing that effect should not be given. We might want to know how the effect occurs, but here again, we will rest easy on islands of familiarity, just as we don't feel compelled to ask why people don't want black teeth.

To summarize, key concepts in rule explanations are abstractions that connect to a pattern of reasoning we have encountered before. This suggests that one way to explain a rule, to make contact with a familiar reasoning pattern, is to generalize the rule. We can see this more clearly from the viewpoint of diagnosis, which makes rich use of hierarchical abstractions.

Consider the following fragment from a rule we call the leukopenia rule:

“If a complete blood count is available and the white blood count is less than 2.5 units, then the following bacteria might be

“If . . . age is greater than 17 and the patient is an alcoholic, then . . .”

Here the relation is a social fact; if the patient is not an adult, we assume that he is not an alcoholic. The third relation we observe is a subtype, as in

“If . . . the patient has undergone surgery and the patient has undergone neurosurgery, then . . .”

All screening relations can be expressed as rules, and some are, such as

“If the patient has not undergone surgery, then the patient has not undergone cardiac surgery.”

(stated negatively, as is procedurally useful). The philosophy behind MYCIN’s rule set is inconsistent in this respect; to be economical and to make the relationship between clauses explicit, all screening clauses should be expressed as world fact rules or hierarchies of parameters. Indeed, the age/alcoholic relation suggests that some of the relations are not definitional and should be modified by certainty factors.

Viewed as a semantic network representation, MYCIN’s rules are links without labels. Even when rules explicitly link problem features, the *kind of relation* is not represented because MYCIN’s rule language does not allow the link to be labeled. For example, a rule could state “If no CBC was taken, then WBC is not available,” but MYCIN allows no way of saying that WBC is a *component of* CBC. Finally, when one problem feature serves as a redefinition of another, such as the relation between leukopenia and WBC, the more abstract problem feature tends to be left out altogether. “Leukopenia” is not a MYCIN parameter; the rule mentions WBC directly, another manifestation of knowledge compilation. *For purposes of explanation, we argue that problem features, their relations, and the nature of the link should be explicit.*

Returning to Figure 29-8, the action concepts, or *diagnostic hypotheses* shown on the right-hand side, are part of a large hierarchy of causes that the problem solver will cite in the final diagnosis. The links in this diagnosis space generally specify refinement of cause, although in our example they strictly designate subclasses. Generally, problem features are abstractions of patient states indicated by the observable symptoms, while the diagnosis space is made up of abstractions of causal processes that produce the symptoms. Paralleling our observations about rule problem features, we note that the relations among diagnostic hypotheses are not represented in MYCIN—nowhere in the knowledge base does it explicitly state that *E. coli* is a bacterium.

Now suppose that the knowledge in Figure 29-8 were available, how would this help us to explain the leukopenia rule? The idea is that we first

restate the rule on a higher level. We point out that a low WBC indicates leukopenia, which is a form of immunosuppression, thus tying the rule to the familiar pattern that implicates gram-negative rods and *Enterobacteriaceae*. This is directly analogous to pointing out that tetracycline causes teeth discoloration, which is a form of undesirable body change, suggesting that the drug should not be given.

By re-representing Figure 29-8 linearly, we see that it is an expansion of the original rule:

WBC < 2.5 → leukopenia
 → immunosuppression
 → compromised host
 → infection by organisms found in body
 → gram-negative rods and *Enterobacteriaceae*
 → *E. coli*, *Pseudomonas*, and *Klebsiella*

The expansion marches up the problem feature hierarchy and then back down the hierarchy of diagnoses. The links of this expansion involve causality composed with identification, subtype, and subset relations. By the hierarchical relationships, a rule on one level “explains” the rule below it. For example, the rule on level 3 provides the detail that links immunosuppression to the gram-negative rods. By generalizing, we have made a connection to familiar concepts.

Tabular rules provide an interesting special case. The CSF protein rule shown in Figure 29-9 appears to be quite formidable. Graphing this rule as shown in Figure 29-10, we find a relatively simple relation that an expert states as “If the protein value is less than 40, I think of viral infections; if it is more than 100, I think of bacterial, fungal, or TB.” This is the first level of generalization, the principle that is implicit in the rule. The second level elicited from the expert is “If the protein value is low, I think of an

RULE500 (The CSF Protein Rule)

- IF: 1) The infection which requires therapy is meningitis,
 2) A lumbar puncture has been performed on the patient, and
 3) The CSF protein is known
- THEN: The type of the infection is as follows:
 If the CSF protein is:
- a) less than 41 then: not bacterial (.5), viral (.7), not fungal (.6), not tb (.5);
 - b) between 41 and 100 then: bacterial (.1), viral (.4), fungal (.1);
 - c) between 100 and 200 then: bacterial (.3), fungal (.3), tb (.3);
 - d) between 200 and 300 then: bacterial (.4), not viral (.5), fungal (.4), tb (.4);
 - e) greater or equal to 300 then: bacterial (.4), not viral (.6), fungal (.4), tb (.4);
-

FIGURE 29-9 The CSF protein rule.

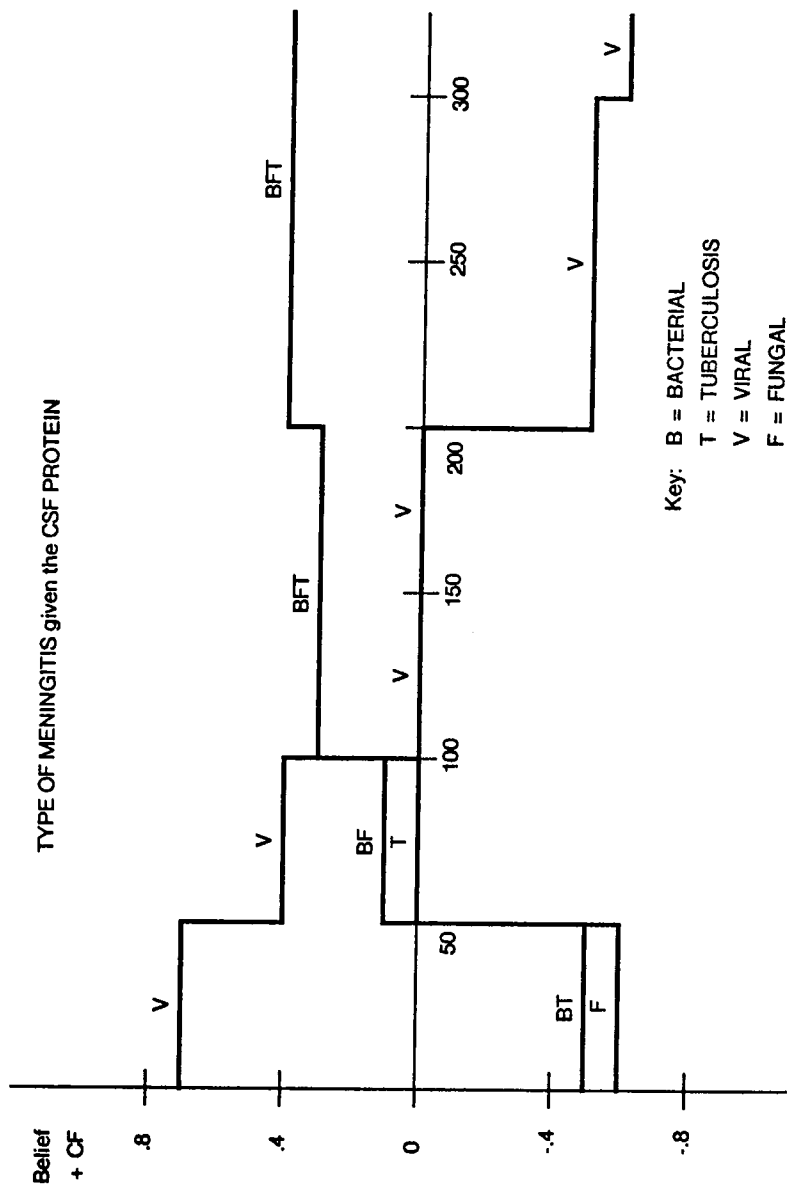


FIGURE 29-10 Graph of the conclusions made by the CSF protein rule (Figure 29-9).

acute process; if it is high, I think of a severe or long-term process.”⁶ Then, at the highest level, the expert states, “An infection in the meninges stimulates protein production.” So in moving up abstraction hierarchies on both the premise and action sides of the rule (acute and chronic are subtypes of infection), we arrive at a mnemonic, just like “bacteria eat glucose.” Abstractions of both the observations and the conclusions are important for understanding the rule.

We might be surprised that explanations of rules provide levels of *detail* by referring to more *general* concepts. We are accustomed to the fact that principled theoretical explanations of, say, chemical phenomena, refer to atomic properties, finer-grained levels of causality. Why should a rule explanation refer to concepts like “compromised host” or “organisms normally found in the body”? The reason is that in trying to understand a rule like the steroids rule, we are first trying to relate it to our understanding of what an infection is at a high, almost metaphorical level. In fact, there are lower-level “molecular” details of the mechanism that could be explained, for example, how steroids actually change the immunological system. But our initial focus as understanders is at the top level—to link the problem feature (steroids) to the global process of meningitis infection. We ask, “What makes it happen? What role do steroids play in the infectious meningitis process?”

The concept of “compromised host” is a label for a poorly understood causal pattern that has value because we can relate it to our understanding of the infection process. It enables us to relate the steroids or WBC evidence to the familiar metaphor in which infection is a war that is fought by the body against invading organisms.

“If a patient is compromised, his or her defenses are down; he or she is vulnerable to attack.”

In general, causal rules argue that some kind of process has occurred. We expect a top-level explanation of a causal rule to relate the premise of the rule to our most general idea of the process being explained. This provides a constraint for how the rule should be generalized, the subject of the next section.

29.3.4 Tying an Explanation to a Causal Model

MYCIN’s diagnostic rules are arguments that a process has occurred in a particular way. There are many kinds of infections, which have different characteristics, but bacterial infections tend to follow the same script: entry of an organism into the body, passage of the organism to the site of infec-

⁶Bacterial meningitis is a severe, acute (short-term) problem, while fungal and TB meningitis are problems of long (chronic) duration.

tion, reproduction of the organism, and causation of observable symptoms. An explanation of a rule that concludes that an organism is causing an infection must demonstrate that this *generic process* has occurred. In short, this is the level of abstraction that the explanation must connect to.

A program was written to demonstrate this idea. The data parameters in MYCIN's 40 diagnostic rules for bacterial meningitis are restated as one or more of the steps of the infectious process script. This restatement is then printed as the explanation of the rule. For example, the program's explanation of the rule linking alcoholism to *Diplococcus* meningitis is:

The fact that the patient is an alcoholic *allows access of organisms from the throat and mouth to lungs* (by reaspiration of secretions).

The fact that the patient is an alcoholic *means that the patient is a compromised host, and so susceptible to infection.*

Words in italics in the first sentence constitute the pattern of "portal and passage." We find that the premise of a rule generally supplies evidence for only a single step of the causal process; the other steps must be inferred by default. For example, the alcoholic rule argues for passage of the *Diplococcus* to the lungs. The person reading this explanation must know that *Diplococcus* is normally found in the mouth and throat of any person and that it proceeds from the lungs to the meninges by the blood. The organism finds conditions favorable for growth because the patient is compromised, as stated in the explanation. In contrast, the leukopenia rule only argues for the patient being a compromised host, so the organisms are the default organisms, those already in the body, which can proceed to the site of infection.⁷

These explanations say which steps are enabled by the data. They place the patient on the path of an infection, so to speak, and leave it to the understander to fill in the other steps with knowledge of how the body normally works. This is why physicians generally refer to the premise data as "predisposing factors." To be understood, a rule must be related to the prior steps in a causal process, the general concepts that explain many rules.

The process of explanation is a bit more complicated in that causal relations may exist between clauses in the rule. We have already seen that one clause may screen another on the basis of world facts, multicomponent test relations, and the subtype relation. The program described here knows these relations and "subtracts off" screening clauses from the rule. Moreover, as discussed in Section 29.4, some clauses describe the context in which the rule applies. These, too, are made explicit for the explanation program and subtracted off. In the vast majority of MYCIN rules, only one premise clause remains, and this is related to the process of infection in the way described above.

⁷As physicians would expect, alcoholism also causes infection by gram-negative rods and *Enterobacteriaceae*. We have omitted these for simplicity. However, this example illustrates that a MYCIN rule can have multiple conclusions reached by different causal paths.

When more than one clause remains after the screening and contextual clauses have been removed, our study shows that a causal connection exists between the remaining clauses. We can always isolate one piece of evidence that the rule is about (for example, WBC in the leukopenia rule); we call this the *key factor* of the rule. We call the remaining clauses *restriction clauses*.⁸ There are three kinds of relations between a restriction clause and a key factor:

- *A confirmed diagnosis explains a symptom.* For example, a petechial rash would normally be evidence for *Neisseria*, but if the patient has leukemia, it may be the disease causing the rash. Therefore, the rule states, “If the patient has a petechial rash (the key factor) and does not have leukemia (the restriction clause), then *Neisseria* may be causing the meningitis.”
- *Two symptoms in combination suggest a different diagnosis than one taken alone.* For example, when both purpuric and petechial rashes occur, then a virus is a more likely cause than *Neisseria*. Therefore, the petechial rule also includes the restriction clause “the patient does not have a purpuric rash.”
- *Weak circumstantial evidence is made irrelevant by strong circumstantial evidence.* For example, a head injury so strongly predisposes a patient to infection by skin organisms that the age of the patient, a weak circumstantial factor, is made irrelevant.

In summary, to explain a causal rule, a teacher must know the purposes of the clauses and connect the rule to abstractions in the relevant process script.

29.3.5 The Relation of Medical Heuristics to Principles

It might be argued that we must go to so much trouble to explain MYCIN's rules because they are written on the wrong level. Now that we have a “theory” for which intermediate parameters to include (“portal,” “pathway,” etc.), why don't we simply rewrite the rules?

The medical knowledge we are trying to codify is really on two levels of detail: (1) principles or generalizations, and (2) empirical details or specializations. MYCIN's rules are empirical. Cleaning them up by representing problem feature relationships explicitly would give us the same set of rules at a higher level. But what would happen if process concepts were incorporated in completely new reasoning steps, for example, if the rule set related problem features to hypotheses about the pathway the organism took through the body? It turns out that reasoning backwards in terms of

⁸Restriction clauses are easy to detect when examining the rule set because they are usually stated negatively.

a causal model is not always appropriate. As we discovered when explaining the rules, not all of the causal steps of the process can be directly confirmed; we can only assume that they have occurred. For example, rather than providing diagnostic clues, the concept of “portal of entry and passage” is very often deduced from the diagnosis itself.

According to this view, principles are good for summarizing arguments, and good to fall back on when you’ve lost grasp on the problem, but they don’t *drive the process* of medical reasoning. Specifically, (1) *if a symptom needs to be explained (is highly unusual), we ask what could cause it* (“Strep-viridans? It is normally found in the mouth. How did it get to the heart? Has the patient had dental work recently?”); (2) *to “prove” that the diagnosis is correct (after it has been constructed), we use a causal argument* (“He has pneumonia; the bacteria obviously got into the blood from the lungs.”). Thus causal knowledge can be used to provide feedback that everything fits.

It may be difficult or impossible to expect a set of diagnostic rules both to serve as concise, “clincher” methods for efficiently getting to the right data and still to represent a model of disease. Put another way, a student may need the model if he or she is to understand new associations between disease and manifestations, but will be an inefficient problem solver if he or she always attempts to convert that model directly to a subgoal structure for solving ordinary problems. Szolovits and Pauker (1978) point out that these “first principles” used by a student are “compiled out” of an expert’s reasoning.

In meningitis diagnosis, the problem is to manage a broad, if not incoherent, hypothesis set, rather than to pursue a single causal path. The underlying theory recedes to the background, and the expert tends to approach the problem simply in terms of weak associations between observed data and bottom-line conclusions. This may have promoted a rule-writing style that discouraged introducing intermediate concepts such as leukopenia, even where they might have been appropriate.

29.4 Teaching Problem-Solving Strategy

A *strategy* is an approach for solving a problem, a plan for ordering methods so that a goal is reached. It is well accepted that strategic knowledge must be conveyed in teaching diagnostic problem solving. As Brown and Goldstein (1977) say:

Without explicit awareness of the largely tacit planning and strategic knowledge inherent in each domain, it is difficult for a person to “make sense of” many sequences of behavior as described by a story, a set of instructions, a problem solution, a complex system, etc. . . . The teacher should articulate

for that domain the higher-order planning knowledge and strategic knowledge for formulating and revising hypotheses about what something means.

Strategic knowledge is general, much like the principles of mechanism we discussed earlier; both relate to processes that have structure. Thus it is not sufficient to show a student only MYCIN's solution, the *surface structure* of the program; we must explain why the rules are invoked in a particular order.

Here it is clear how *teaching* how to do something differs from merely *explaining* how something was done: we want the student to be able to replicate what he or she observes, to solve similar problems independently. This is why mnemonics are useful when justifying a rule. Regarding strategy, we must again address human foibles and preference: we must teach a strategy that a human can follow.

The main points of this section are:

- MYCIN's strategy is different from a physician's strategy;
- MYCIN's strategic knowledge is embedded in the rules, indistinguishable from screening and problem feature clauses;
- A domain-independent representation of strategy is useful for teaching and for purposes of economy.

29.4.1 Surface and Deep Structure of MYCIN

A person trying to understand a MYCIN consultation observes that pieces of data are requested by the program as shown in Figure 29-11. Conceptually, these questions are terminals hanging below an AND node in a subgoal tree, as shown in Figure 29-12. Following the terminology of Brown and Goldstein (1977), a rule node is a *method* for achieving a goal (e.g., "organisms that might be causing the infection") by asking questions or pursuing a set of subgoals to achieve. Therefore, the tree of rules and subgoals is part of a deep-structured trace that they postulate is constructed when the understander makes sense of the surface problem solution.

It is not sufficient for a student to know all of the possible methods he or she can bring to bear on a problem. He or she generally needs a *plan* for solving the problem, that is, needs schemata for kinds of problems that can be tackled using different *approaches* or lines of reasoning. A plan sets up a rational sequence of methods that might get you closer to the solution, but without guarantees.

The hypothetico-deductive strategy used in medical problem solving constitutes a plan for focusing on hypotheses and selecting confirmatory questions (Elstein et al., 1978). However, the methods selected in Figure 29-12 (Rules 511 through 578) have been applied in a fixed, arbitrary order—not planned by the rule author. MYCIN has no "deep structure"

- 31) Has Pt538 ever undergone any type of surgery?
** YES
- 32) Does Pt538 have a history of NEUROSURGERY?
** NO
- 33) Does Pt538 live in a crowded environment?
** NO
- 34) Do you suspect recent alcoholic history in Pt538?
** NO
- 35) Is meningitis a hospital-acquired infection?
** YES
- 36) Is Pt538's clinical history consistent with EPIGLOTTITIS?
** NO
- 37) Is Pt538's clinical history consistent with OTITIS-MEDIA?
** NO
- 38) Has Pt538 ever undergone splenectomy?
** NO
- 39) Is Pt538 a burn patient?
** YES

FIGURE 29-11 Excerpt from a MYCIN consultation showing requests for relevant data.

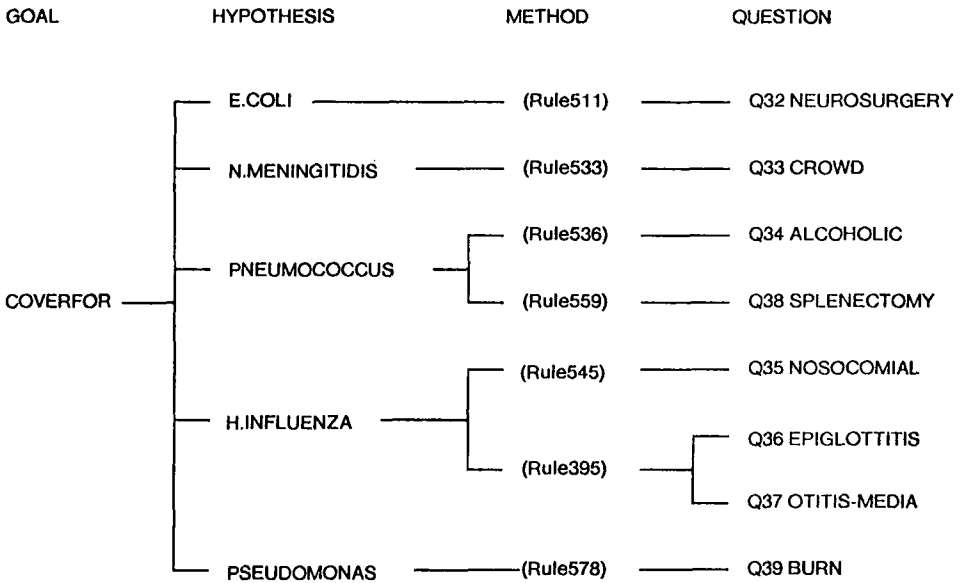


FIGURE 29-12 Portion of the AND/OR tree corresponding to the questions shown in Figure 29-11 (reorganized according to the hypothesis each rule supports).

RULE092 (The Goal Rule)

- IF: 1) Gather information about cultures taken from the patient and therapy he is receiving,
 2) Determine if the organisms growing on cultures require therapy
 3) Consider circumstantial evidence for additional organisms that therapy should cover
 THEN: Determine the best therapy recommendation

RULE535 (The Alcoholic Rule)

- IF: 1) The infection which requires therapy is meningitis,
 2) Only circumstantial evidence is available for this case,
 3) The type of meningitis is bacterial,
 4) The age of the patient is greater than 17 years, and
 5) The patient is an alcoholic,
 THEN: There is evidence that the organisms which might be causing the infection are *diplococcus-pneumoniae* (.3) or *e.coli* (.2)

FIGURE 29-13 The goal rule and the alcoholic rule.

plan at this level; the program is simply applying rules (methods) exhaustively. This lack of similarity to human reasoning severely limits the usefulness of the system for teaching problem solving.

However, MYCIN does have a problem-solving strategy above the level of rule application, namely the control knowledge that causes it to pursue a goal at a certain point in the diagnosis. We can see this by examining how rules interact in backward chaining. Figure 29-13 shows the goal rule and a rule that it indirectly invokes. In order to evaluate the third clause of the goal rule, MYCIN tries each of the COVERFOR rules; the alcoholic rule is one of these (see also Figure 29-12). We call the goal rule a *task rule* to distinguish it from inference rules. Clause order counts here; this is more a procedure than a logical conjunction. The first three clauses of the alcoholic rule, the *context clauses*, also control the order in which goals are pursued, just as is true for a task rule. We can represent this hidden structure of goals by a tree which we call the *inference structure* of the rule base (produced by "hanging" the rule set from the goal rule). Figure 29-14 illustrates part of MYCIN's inference structure.^{9,10}

The program's strategy comes to light when we list these goals in the order in which the depth-first interpreter makes a final decision about them. For example, since at least one rule that concludes "significant" (goal 4 in Figure 29-14) mentions "contaminant" (goal 3), MYCIN applies *all* of the "contaminant" rules before making a final decision about "significant." Analyzing the entire rule set in a similar way gives us the ordering (shown in Figure 29-14):

⁹Some definitions of terms used in the following discussion: TREATFOR = organisms to be treated, based on direct laboratory observation; COVERFOR = organisms to be treated, based on circumstantial evidence; SIGNIFICANT = this organism merits therapeutic attention, based on the patient's degree of sickness and validity of culture results; CONTAMINANT = the finding of this organism is spurious; it was probably introduced during sampling from the cultured site of the body, as a blood culture might include skin organisms.

¹⁰We leave out the goals REGIMEN and TREATFOR because they are just placeholders for task rules, like subroutine names.

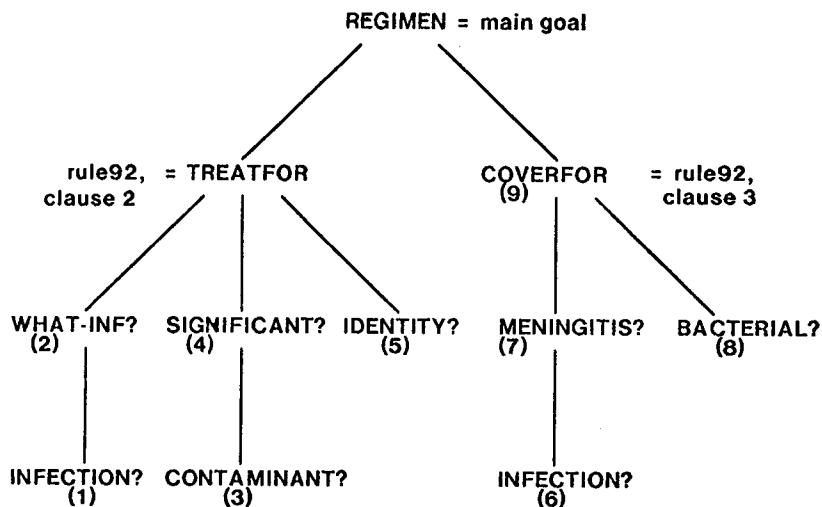


FIGURE 29-14 Portion of MYCIN's inference structure. (Numbers give the order in which nonplaceholder goals are achieved by the depth-first interpreter.)

1. Is there an infection?
2. Is it bacteremia, cystitis, or meningitis?
3. Are there any contaminated cultures?
4. Are there any good cultures with significant growth?
5. Is the organism identity known?
6. Is there an infection? (already done in Step 1)
7. Does the patient have meningitis? (already done in Step 2)
8. Is it bacterial?
9. Are there specific bacteria to cover for?

MYCIN's diagnostic plan is in two parts, and both proceed by top-down refinement. This demonstrates that a combination of structural knowledge (the taxonomy of the diagnosis space—infection, meningitis, bacterial, *Diplococcus* . . .) and strategic knowledge (traversing the taxonomy from the top down) is procedurally embedded in the rules. In other words, we could write a program that interpreted an explicit, declarative representation of the diagnosis taxonomy and domain-independent form of the strategy to bring about the same effect.

At this level, MYCIN's diagnostic strategy is not a complete model of how physicians think, but it could be useful to a student. As the quote from Brown and Goldstein would indicate, and as has been confirmed in GUIDON research, teachers do articulate both the structure of the problem

META-RULE002

IF: 1) The infection is pelvic-abscess, and
2) There are rules which mention in their premise enterobacteriaceae, and
3) There are rules which mention in their premise gram-positive rods,
THEN: There is suggestive evidence (.4) that the former should be done before the latter

FIGURE 29-15 A MYCIN meta-rule.

space and the nature of the search strategy to students. This means that we need to represent explicitly the fact that the diagnosis space is hierarchical and to represent strategies in a domain-independent form. If a strategy is not in domain-independent form, it can be taught by examples, but not explained.

29.4.2 Representing Strategic Knowledge in Meta-Rules

How might we represent domain-independent strategic knowledge in a rule-based system? In the context of the MYCIN system, Davis pursued the representation of strategic knowledge by using *meta-rules* to order and prune methods (Chapter 28). These meta-rules are invoked just before the object-level rules are applied to achieve a goal. An example of an infectious disease meta-rule is shown in Figure 29-15 (see Figure 28-12 for other examples). Observe that this is a *strategy* for pursuing a goal. In particular, this meta-rule might be associated with the goal "identity of the organism." It will be invoked to order the rules for every subgoal in the search tree below this goal; in this simple way, the rule sets up a line of reasoning. This mechanism causes some goals to be pursued before others, orders the questions asked by the system, and hence changes the surface structure of the consultation.

Although meta-rules like this can capture and implement strategic knowledge about a domain, they have their deficiencies. Like the performance rules we have examined, Davis's domain-dependent examples of meta-rules leave out knowledge important for explanation. Not only do they leave out the domain-specific support knowledge that justifies the rules, they leave out the domain-independent strategic principles that GUIDON should teach. In short, meta-rules provide the *mechanism* for controlling the use of rules, but not the domain-independent *language* for making the strategy explicit.

The implicit strategic principle that lies behind Meta-Rule 002 is that common causes of a disorder should be considered first. The structural knowledge that ties this strategy to the object-level diagnostic rules is an explicit partitioning of the diagnosis space taxonomy, indicating that the group of organisms called *Enterobacteriaceae* are more likely than gram-

positive rod organisms to cause pelvic infections. This is what we want to teach the student. One can imagine different common causes for different infection types, requiring different meta-rules. But if all meta-rules are as specific as Meta-Rule 002, principles will be compiled into many rules redundantly and the teaching points will be lost.

What does a domain-independent meta-rule look like, and how is it interfaced with the object-level rules? To explore this question, we have reconfigured the MYCIN rule base into a new system, called NEOMYCIN (Clancey and Letsinger, 1981). Briefly, meta-rules are organized hierarchically (again!) into tasks, such as “group and refine the hypothesis space.” These rules manage a changing hypothesis list by applying different kinds of knowledge sources, as appropriate. Knowledge sources are essentially the object-level rules, indexed in the taxonomy of the diagnosis space by a domain-independent structural language.

For example, one meta-rule for achieving the task of pursuing a hypothesis is “If there are unusual causes, then pursue them.”¹¹ Suppose that the current hypothesis is “bacterial meningitis.” The program will use the structural label “unusual causes” to retrieve the nodes “gram-negative rods,” “enterobacteriaceae,” and “listeria,” add them to the hypothesis list, and pursue them in turn. When there are no “unusual causes” indicated, the meta-rule simply does not apply. Pursuing gram-negative rods, the program will find that leukopenia is a relevant factor, but will first ask if the patient is a compromised host (Figure 29-8), modeling a physician’s efficient casting of wider questions.

Other terms in the structural language used by NEOMYCIN’s domain-independent meta-rules are

1. process features, such as *extent* and *location* of disease;
2. the *enabling step* of a causal process;
3. *subtype*;
4. *cause*;
5. *trigger association*;
6. *problem feature screen*; and structural properties of the taxonomy, such as *sibling*.

In effect, the layer of structural knowledge allows us to separate out *what* the heuristic is from *how* it will be used. How domain-specific heuristics like MYCIN’s rules should be properly integrated with procedural, strategic knowledge is an issue at the heart of the old “declarative/procedural

¹¹This rule appears after the rule for considering common causes, and the ordering is marked as strategically significant. Domain-independent meta-rules have justifications, organization, and strategies for using them. Their justification refers to properties of the search space and the processor’s capabilities.

controversy” (Winograd, 1975). We conclude here that, for purposes of teaching, the hierarchies of problem features and of the diagnosis space should be represented explicitly, providing a useful means for indexing the heuristics by both premise and action. A structural language of cause, class, and process can connect this domain-specific knowledge to domain-independent meta-rules, the strategy for problem solving.

29.4.3 Self-Referencing Rules

Self-referencing rules provide an interesting special example of how problem-solving strategies can be embedded in MYCIN’s rules. A rule is self-referencing if the goal concluded by the action is also mentioned in the premise. An example is the aerobicity rule shown in Figure 29-16.¹²

RULE086

IF: 1) The aerobicity of the organism is not known, and
2) The culture was obtained more than 2 days ago,

THEN: There is evidence that the aerobicity of the organism is obligate-aerob (.5) or facultative (.5)

FIGURE 29-16 The aerobicity rule.

This rule is tried only after all of the non-self-referencing rules have been applied. The cumulative conclusion of the non-self-referencing rules is held aside, then the self-referencing rules are tried, using in each rule the tentative conclusion. Thus the first clause of Rule 86 will be true only if none of the standard rules made a conclusion. The effect is to reconsider a tentative conclusion. When the original conclusion is changed by the self-referencing rules, this is a form of nonmonotonic reasoning (Winograd, 1980). We can restate MYCIN’s self-referencing rules in domain-independent terms:

- *If nothing has been observed, consider situations that have no visible manifestations.* For example, the aerobicity rule: “If no organism is growing in the culture, it may be an organism that takes a long time to grow (obligate-aerob and facultative organisms).”
- The self-referencing mechanism makes it possible to state this rule without requiring a long premise that is logically exclusive from the remainder of the rule set.

¹²Aerobicity refers to whether an organism can grow in the presence of oxygen. A facultative organism can grow with or without oxygen; an anaerobic organism cannot grow with oxygen present; and an obligate-aerob is aerobic only in a certain stage of growth. Note that the rule is self-referencing in that aerobicity is mentioned in both the premise and the conclusion.

- *If unable to make a deduction, assume the most probable situation.* For example: “If the gram stain is unknown and the organism is a coccus, then assume that it is gram-positive.”
- *If there is evidence for two hypotheses, A and B, that tend to be confused, then rule out B.* For example: “If there is evidence for TB and fungal, and you have hard data for fungal, rule out TB.”

Like Meta-Rule 002, self-referencing rules provide a useful mechanism for controlling the use of knowledge, but they leave out both the domain-dependent justification and the general, domain-independent reasoning strategy of which they are examples. These rules illustrate that strategy involves more than a search plan; it also takes in principles for reasoning about evidence. It is not clear that a teacher needs to state these principles explicitly to a student. They tend to be either common sense or almost impossible to think about independently of an example. Nevertheless, they are yet another example of strategic knowledge that is implicit in MYCIN's rules.

29.5 Implications for Modifiability and Performance

MYCIN achieved good problem-solving performance even without having to reason about the structural, strategic, and support knowledge we have been considering. However, there are situations in which knowledge of justification and strategy allows one to be a more flexible problem solver, to cope with novel situations, in ways that MYCIN cannot. Knowing the basis of a rule allows you to know when not to apply it, or how to modify it for special circumstances. For example, knowing that tetracycline won't kill the young patient but the infection might, you may have to dismiss social ramifications and prescribe the drug. You can deliberately break the rule because you understand the assumptions underlying it.

There will also be problems that cannot be diagnosed using MYCIN's rules. For example, several years ago *Coccidioides* meningitis strangely appeared in the San Francisco Bay Area. We would say that this “violates all the rules.” To explain what was happening, one has to reason about the underlying mechanisms. The organisms were traveling from the San Joaquin Valley to the Bay Area by “freak southeastern winds,” as the newspapers reported. The basic mechanism of disease was not violated, but this time the patients didn't have to travel to the Valley to come in contact with the disease. A human expert can understand this because he or she can fit the new situation to the model. Examples like these make us realize that

AI systems like MYCIN can only perform some of the functions of an expert.

Regarding modifiability, the process of reconfiguring MYCIN's rules in NEOMYCIN's terms required many hours of consultation with the original rule authors in order to unravel the rules. As shown in this paper, the lack of recorded principles for using the representation makes it difficult to interpret the purposes of clauses and rules. The strategy and overall design of the program have to be deduced by drawing diagrams like Figure 29-14. Imagine the difficulty any physician new to MYCIN would have modifying the CSF protein table (Figure 29-9); clearly, he or she would first need an explanation from the program of why it is correct.

We also need a principled representation to avoid a problem we call *concept broadening*. When intermediate problem abstractions are omitted, use of goals becomes generalized and weakened. This happened in MYCIN as the meaning of "significance" grew to include both "evidence of infection" and "noncontaminated cultures." As long as the rule author makes an association between the data and some parameter he or she wants to influence, it doesn't matter for correct performance that the rule is vague. But vague rules are difficult to understand and modify.

A rule base is built and extended like any other program. Extensive documentation and a well-structured design are essential, as in any engineering endeavor. The framework of knowledge types and purposes that we have described would constitute a "typed" rule language that could make it easier for an expert to organize his or her thoughts. On the other hand, we must realize that this meta-level analysis may impose an extra burden by turning the expert into a taxonomist of his or her own knowledge—a task that may require considerable assistance, patience, and tools.

29.6 Application of the Framework to Other Systems

To illustrate further the idea of the strategy, structure, and support framework and to demonstrate its usefulness for explaining how a program reasons, several knowledge-based programs are described below in terms of the framework. For generality, we will call inference associations such as MYCIN's rules *knowledge sources* (KS's). We will not be concerned here with the representational notation used in a program, whether it be frames, production rules, or something else. Instead, we are trying to establish an understanding of the knowledge contained in the system: what kinds of inferences are made at the KS level, how these KS's are structured explicitly in the system, and how this structure is used by strategies for invoking KS's. This is described in Table 29-1.

TABLE 29-1 Examples of Various Types of Knowledge Structures in Several AI Systems

System	Domain	KS example	Strategy	Structure	Support
DENDRAL (Buchanan and Feigen- baum, 1978)	Chemistry, mass spectro- metry analysis	Identification rules relating functional groups to spectral peaks	Aggregation heuris- tics build superatoms and generate all plausible interstitial structures	Family trees of func- tional groups (ke- tones, ethers, etc.)	Molecular chemistry
HEARSAY II (Erman et al., 1980)	Speech un- derstanding	Hypothesizing words from syllable level	Policy KS's control hypothesizing words to generate thresh- olds (data-directed)	Hierarchy of inter- pretation levels with links to KS's	Grammar identifica- tion properties of phonemes, syllables, and words
AM (Lenat, 1976)	Concept for- mation, math- ematical dis- covery	Rules to create con- cepts and fill in facets	Activity heuristics propose tasks (prior- ity agenda focuses heuristics)	Hierarchy of heuris- tics associated with most general concept/ context to which they apply	Theory of interest- ingness, chiefly based on generalizing and specializing
MOLGEN (Stefik, 1979)	Molecular ge- netics, experi- ment plan- ning	Specific lab tech- niques: input objects → molecular changes and byproducts	Determine differ- ences, sketch plan, refine steps (message passing)	Hierarchy of labora- tory operation types (used by refinement design operator)	Processes of molecu- lar biology
CENTAUR (Aikins, 1983)	Medical diag- nosis, pulmo- nary function	Disease component → evidence for proto- type	Hypothesis-directed, top-down refinement (agenda)	Hierarchy of disease prototypes	Disease patterns, bio- logical processes
NEOMYCIN (Clancey and Letsinger, 1981)	Medical diag- nosis, diseases causing neu- rological symptoms	Data → evidence for disease process or causal state/category	Grouping and refin- ing list of hypotheses (meta-rules focus pursuit)	Multiple hierarchies of etiological pro- cesses	Disease patterns, bio- logical processes

29.6.1 The Character of Structural Knowledge

One product of this study is a characterization of different ways of structuring KS's for different strategic purposes. In all cases, the effect of the structural knowledge is to provide a handle for separating out what the KS is from when it is to be applied.¹³

The different ways of structuring KS's are summarized here according to the processing rationale:

- *Organize KS's hierarchically by hypothesis for consistency in data-directed interpretation.* In DENDRAL, if a functional group is ruled out, more specific members of the family are not considered during forward-directed, preliminary interpretation of spectral peaks. Without this organization of KS's, earlier versions of DENDRAL could generate a subgroup as a plausible interpretation while ruling out a more general form of the subgroup, as if to say "This is an ethyl ketone but not a ketone." (Buchanan et al., 1970).
- *Organize KS's hierarchically by hypothesis to eliminate redundant effort in hypothesis-directed refinement.* In DENDRAL, the family trees prevent the exhaustive structure generator from generating subgroups whose more general forms have been ruled out. The same principle is basic to most medical diagnosis systems that organize diagnoses in a taxonomy and use a top-down refinement strategy, such as CENTAUR and NEOMYCIN.
- *Organize KS's by multiple hypothesis hierarchies for efficient grouping (hypothesis-space splitting).* Besides using the hierarchy of generic disease processes (infectious, cancerous, toxic, traumatic, psychosomatic, etc.), NEOMYCIN groups the same diseases by multiple hierarchies according to disease process features (organ system involved, spread in the system, progression over time, etc.). When hypotheses are under consideration that do not fall into one confirmed subtree of the primary etiological hierarchy, the group and differentiate strategy is invoked to find a process feature dimension along which two or more current hypotheses differ. A question will then be asked, or a hypothesis pursued, to differentiate among the hypotheses on this dimension.
- *Organize KS's for each hypothesis on the basis of how KS data relates to the hypothesis, for focusing on problem features.* In NEOMYCIN, additional relations make explicit special kinds of connections between data and hypotheses, such as "this problem feature is the enabling causal step for this diagnostic process," and meta-rules order the selection of questions (invocation of KS's) by indexing them indirectly through these relations. For example, "If an enabling causal step is known for the hypothesis to be confirmed, try to confirm that problem feature." The meta-rules that

¹³In this section, the term *hypothesis* generally refers to a diagnostic or explanatory interpretation made by a KS (in terms of some model), although it can also be a hypothesis that a particular problem feature is present.

reference these different relations (“enabling step,” “trigger,” “most likely manifestation”) are ordered arbitrarily. Meta-meta-rules don’t order the meta-rules because we currently have no theoretical basis for relating the first-order relations to one another.

- *Organize KS’s into data/hypothesis levels for opportunistic triggering at multiple levels of interpretation.* HEARSAY’s blackboard levels (sentence, word sequence, word, etc.) organize KS’s by the level of analysis they use for data, each level supplying data for the hypothesis level above it. When new results are posted on a given level, KS’s that “care about” that level of analysis are polled to see if they should be given processing time. *Policy KS’s* give coherence to this opportunistic invocation by affecting which levels will be given preference. CRYSLIS (Engelmore and Terry, 1979) (a program that constructs a three-dimensional crystal structure interpretation of x-ray crystallographic data) takes the idea a step further by having multiple planes of blackboards; one abstracts problem features, and the other abstracts interpretations.
- *Organize KS’s into a task hierarchy for planning.* In MOLGEN, laboratory operators are referenced indirectly through tasks that are steps in an abstract plan. For example, the planning level *design decision* to refine the abstract plan step MERGE is accomplished by indexing laboratory operators by the MERGE task (e.g., MERGE could be refined to using a ligase to connect DNA structures, mixing solutions, or causing a vector to be absorbed by an organism). Thus tasks in planning are analogous to hypotheses in interpretation problems.
- *Organize KS’s into a context specialization hierarchy for determining task relevance.* In AM, relevant heuristics for a task are inherited from all concepts that appear above it in the specialization hierarchy. Thus AM goes a step beyond most other systems by showing that policy KS’s must be selected on the basis of the kind of problem being solved. Lenat’s work suggests that this might be simply a hierarchical relationship among kinds of problems.

The above characterizations of different organizations for knowledge are a first step toward a vocabulary or language for talking about indirect reference of KS’s. It is clear that strategy and structure are intimately related; to make this clearer, we return to the earlier topic of explanation.

Teaching a strategy might boil down to saying “think in terms of such-and-such a structural vocabulary in order to get this strategical task done”—where the vocabulary is the indexing scheme for calling KS’s to mind. So we might say, “Think in terms of families of functional subgroups in order to rule out interpretations of the spectral peaks.” Or, “Consider process features when diseases of different etiologies are possible.” That is, teaching a strategy involves in part the *teaching of a perspective for relating KS’s hierarchically* (e.g., “families of functional subgroups” or “disease process features”) and then *showing how these relations provide leverage for managing a large amount of data or a large number of hypotheses.* The explanation

of the sought-after leverage must be in terms of some task for carrying the problem forward, thus tying the structuring scheme to the overall process of what the problem solver is trying to do. Thus we say “to rule out interpretations” or “to narrow down the problem to one etiological process” or (recalling Figure 29-4) “to broaden the spectrum of possibilities.” In this way, we give the student a meta-rule that specifies what kind of vocabulary to consider for a given strategical task.

Davis’ study of meta-rules (Chapter 28) suggested a need for a vocabulary of meta-rule knowledge. His examples suggested just a few conceptual primitives for describing refinement (ordering and utility of KS’s) and a few primitives for describing object-level knowledge (KS input and output). All of the strategies in our examples deal with ordering and utility criteria for KS’s; so we have nothing to add there. All of the examples given here reference KS’s by the data they act upon, the hypotheses they support or the tasks they accomplish, except for AM, which references KS’s by their scope or domain of applicability. What is novel about the analysis here is the focus on *relations* among hypotheses and among data.

From our domain-independent perspective, strategical knowledge selects KS’s on the basis of the causal, subtype, process, or scoping relation they bear to hypotheses or data currently thought to be relevant to the problem at hand. Thus our meta-rules make statements like these:

1. “Consider KS’s that would demonstrate a prior cause for the best hypothesis.”
2. “Don’t consider KS’s that are subtypes of ruled-out hypotheses.”
3. “Consider KS’s that abstract known data.”
4. “Consider KS’s that distinguish between two competing kinds of processes.”
5. “Consider KS’s relevant to the current problem domain.”

To summarize, the structural knowledge we have been studying consists of relations that hierarchically abstract data and hypotheses. These relations constitute the vocabulary by which domain-independent meta-rules invoke KS’s. The key to our analysis is our insistence on domain-independent statement of meta-rules—a motivation deriving from our interest in explanation and teaching.

29.6.2 Explicitness of Strategical Knowledge

Another consideration for explanation is whether or not the strategy for invoking KS’s is explicit. To some extent, system designers are not generally interested in representing high-level strategies that are always in effect and never reasoned about by the program. Instead, they are satisfied if their system can be programmed in the primitives of their representation language to bring about the high-level effect they are seeking. For example,

top-down refinement is “compiled into” CENTAUR’s hierarchy itself by the control steps that specify on each level what to do next (e.g., “After confirming obstructive airways disease, determine the subtype of obstructive airways disease.”). By separating control steps from disease inferences, Aikins improved the explanation facility, one of the goals of CENTAUR. However, the rationale for these control steps is not represented—it is just as implicit as it was in PUFF’s contextual clauses. In contrast, NEOMYCIN’s “explore and refine” task clearly implements top-down refinement through domain-independent meta-rules. However, these meta-rules are ordered to give preference to siblings before descendents—an example of an implicit strategy.

One common way of selecting KS’s is on the basis of numerical measures of priority, utility, interestingness, etc. For example, CENTAUR, like many medical programs, will first request the data that give the most weight for the disease under consideration. Thus the weight given to a KS is another form of indexing by which a strategy can be applied. If we wish to explain these weights, we should ideally replace them by descriptors that “generate” them, and then have the strategy give preference to KS’s having certain descriptors. NEOMYCIN’s meta-rules for requesting data (described above) are a step in this direction.

MOLGEN’s “least-commitment” meta-strategy is a good example of implicit encoding by priority assignment. The ordering of tasks specified by least commitment is “Look first for differences, then use them to sketch out an abstract plan, and finally refine that plan. . . .” This ordering of tasks is implicit in the numerical priorities that Stefik has assigned to the design operators in MOLGEN. Therefore, an explanation system for MOLGEN could not explain the least-commitment strategy but could only say that the program performed one task before another because the priority was higher for the former.

29.6.3 Absence of Support Knowledge

We have little to say about support knowledge in these systems because none of them represents it. That is, the causal or mathematical models, statistical studies, or world knowledge that justifies the KS’s is not used during reasoning. As discussed in Section 29.5, this limitation calls into question the problem-solving flexibility or “creativity” of these programs. In any case, the knowledge is not available for explanation.

29.6.4 Summary

The strategy/structure/support framework can be applied to any knowledge-based system by asking certain questions: What are the KS’s in the system, i.e., what kinds of recognition or construction operations are per-

formed? How are the KS's labeled or organized, by data/constraint or by hypothesis/operation? Is this indexing used by the interpreter or by explicit strategic KS's, or is it just an aid for the knowledge engineer? What theoretical considerations justify the KS's? Is this knowledge represented? With this kind of analysis, it should be clear how the knowledge represented needs to be augmented or decomposed if an explanation facility is to be built for the system. Quite possibly, as in MYCIN, the representational notation will need to be modified as well.

29.7 Conclusions

The production rule formalism is often chosen by expert system designers because it is thought to provide a perspicuous, modular representation. But we have discovered that there are points of flexibility in the representation that can be easily exploited to embed structural and strategic knowledge in task rules, context clauses, and screening clauses. Arguing from a teacher's perspective, we showed that hierarchies of problem features and diagnoses, in addition to a domain-independent statement of strategy, are useful to justify rules and teach approaches for using them. Also, when a rule is causal, satisfactory explanations generalize the rule in terms of an underlying process model. This same knowledge should be made explicit for purposes of explanation, ease of modification, and potential improvement of problem-solving ability.

Characterizing knowledge in three categories, we concluded that MYCIN's rules were used as a programming language to embed strategic and structural principles. However, while context and screening clauses are devices that don't precisely capture the paths of expert reasoning, the basic connection between data and hypothesis is a psychologically valid association. As such, the "core rules" represent the experts' knowledge of causal processes in proceduralized form. Their knowledge is not necessarily compiled into this form, but may be compiled with respect to causal models that may be incomplete or never even learned. For this reason, support knowledge needs to be represented in a form that is somewhat redundant to the diagnostic associations, while structure and strategy can be directly factored out and represented declaratively.

The lessons of this study apply to other knowledge-based programs, including programs that do not use the production rule representation. The first moral is that one cannot simply slap an interactive front end onto a good AI program and expect to have an adequate teaching system. Similarly, an explanation system may have to do more than just read back reasoning steps and recognize questions: it may be useful to abstract the

reasoning steps, relating them to domain models and problem-solving strategies.

Other knowledge bases could be studied as artifacts to evaluate the expressiveness of their representation. Is the design of the inference structure explicit? Can it be reasoned about and used for explanation? Where are the choice points in the representation and what principles for their use have not been represented explicitly? For rule-based systems one should ask: What is the purpose of each clause in the rule and why are clauses ordered this way? Why is this link between premise and conclusion justified? Under what circumstances does this association come to mind?

Finally, future knowledge engineering efforts in which the knowledge of experts is codified could benefit from an epistemology that distinguishes KS's from meta-level knowledge of three kinds—strategy, structure, and support knowledge. Relative to that framework, then, it makes sense to ask about the appropriateness of representing knowledge using rules, units, or other notations. When the system fails to behave properly, changes to *either* the epistemology or the rules should be entertained. In fact, this is a cyclic process in which changes are made to the rules that subtly tear at the framework, and after incorporating a series of changes, a new, better epistemology and revised notation can be arrived at. (For example, a single MYCIN rule might seem awkward, but a pattern such as 40 rules having the same first 3 clauses suggests some underlying structure to the knowledge.) Thus a methodology for converging on an adequate epistemology comes in part from constant cycling and reexamining of the entire *system* of rules.

The epistemology that evolved from attempts to reconfigure MYCIN's rules is NEOMYCIN's etiological taxonomy, multiple disease process hierarchies, data that trigger hypotheses, etc., plus the domain-independent task hierarchy of meta-rules. In our use of terms like "problem feature," we have moved very far from MYCIN's too abstract concept of "clinical parameter," which did not distinguish between data and hypotheses. Our epistemology provides an improved basis for interpreting expert reasoning, a valuable foundation for knowledge engineering, as echoed by Swanson et al. (1977):

Three aspects of the expert's adaptation are especially important to the design of decision support systems: the generative role of basic principles of pathophysiology, the hierarchical structure of disease knowledge, and the heuristics used in coping with information processing demands.

These categories of knowledge provide a framework for understanding an expert. We ask, "What *kind* of knowledge is the expert describing?" This framework enables us to focus our questions so that we can separate out detailed descriptions of the expert's causal model from both the associations that link symptom to disorder and the strategies for using this knowledge.

29.8 Postscript: How the Rule Formalism Helped

Despite some apparent shortcomings of MYCIN's rule formalism noted in this chapter and throughout the book, we must remember that the program has been influential because it works well. The uniformity of representation has been an important asset. With knowledge being so easy to encode, it was perhaps the simple parameterization of the problem that made MYCIN successful. The program could be built and tested quickly at a time when little was known about building expert systems. Finally, the explicit codification of medical knowledge, now taken for granted in expert systems, allows examination of, and improvement upon, the knowledge structures.