R. X. Latin, G. E. Miles, and J. C. Rettinger
Purdue University, West Lafayette, IN

Expert Systems in Plant Pathology

Computers have become established as valuable tools for plant pathologists since the technological advances of the 1970s. Most of us use a computer or microprocessor for word processing, data acquisition, or data analysis. These computer applications help us to improve the efficiency and quality of our research, teaching, and extension efforts. Other, more sophisticated computer applications in plant pathology are an integral part of the research. Research that involves simulation and modeling of plant disease epidemics and disease forecasting would hardly be possible without computers.

A new generation of computers and computer applications has arisen from research in artificial intelligence, a relatively recent field of computer science concerned with developing smarter and more useful programs that solve problems similar to the way a human would. The field of artificial intelligence encompasses robotics and natural language processing, but the area of expert systems has the most immediate significance for society because of its record of successful applications. Expert systems are a special class of computer programs that emulate the decision-making logic human experts use to solve problems in their particular fields. Expert systems may require computer hardware and software not normally used by plant pathologists. Like computerized disease-forecasting programs, expert systems are decision support resources, but unlike the forecasters, they can be applied to many different kinds of problems.

Our objective in this article is to provide a general introduction to expert systems based on our experience building a system for diagnosing diseases of muskmelon. One of the obstacles we found in describing expert systems was that the lexicon of artificial intelligence is unfamiliar to plant pathologists. We have tried to overcome that by including a glossary of basic terms (Fig. 1) and by using analogies and examples where possible.

How Expert Systems Differ from Other Programs

Insight into the special nature of expert systems can be gained from a comparison of the elements of expert systems and conventional programs. The most fundamental difference is that conventional programs deal with data, whereas expert systems deal with knowledge. Data are facts that are observed directly or derived by experimentation and calculation. Computer data bases store the facts and numerical data retrieved later by the user for interpretation. Knowledge implies an awareness or understanding gained through experience or study. A knowledge base goes beyond the mere storage of facts to the point of interpreting the data and relating stored information to new facts in order to provide advice. Symptom observations represent data commonly used to diagnose plant diseases. For example, data collected from unhealthy muskmelon plants may include such observations as clustering of affected plants in a low area of the field, wilting, necrotic and water-soaked lesions on lower stems, gummy exudate within the necrotic areas, and visible necrosis within the vascular system. A person knowledgeable about muskmelon diseases would reason that the combined data represent symptoms of Fusarium wilt. Similarly, an expert system would be able to apply reasoning processes in the interpretation of observations and facts.

One of the main components of an expert system is the knowledge base. It contains all of the domain-specific facts and knowledge. Knowledge is represented (i.e., structured in the program) so that the problem is solved efficiently. There are several formal methods for knowledge representation (11), but the most widely used is the rule-based method. In a rule-based system, knowledge is represented as IF-THEN statements (rules). The IF

Purdue University Agricultural Experiment Station Journal Paper No. 11,070.

portion of a rule contains one or more conditional clauses consisting of parameters and their values. A parameter is a variable that helps describe the nature of the problem. For example, a system designed to identify plant disorders is likely to include the plant symptom as a parameter, and its value may be leaf spot, wilt, canker, or some other type of plant symptom. The THEN portion of the rule consists of a single conclusion that is drawn after all of the conditional statements are satisfied. This method of representation is most appropriate when knowledge results from compiled experience in solving certain types of problems. The system operates by checking the rules against information about the current situation. The current information may be obtained by direct observation or by consultation with a client. When the IF portion of a rule is true in the current situation, the action specified by the THEN portion is performed, i.e., the rule is executed.

For example, Figure 2 presents the rule for diagnosing Alternaria leaf blight and a photograph of the symptoms representing the current data about the disease. During a consultation, the computer prompts the user with parameters (plant symptom type, leaf lesion color, etc.) and associated values (wilt, leaf spot, brown, yellow, etc.) that the user selects to describe the current data. The computer then compares the current data with the IF portion of the rule. When the current data match the IF portion of the rule, the rule is executed and the conclusion is reached. Execution of a rule also can add new data to the knowledge base. In Figure 2A, for example, the conclusion that the disease in question is Alternaria leaf blight represents new information synthesized by the system after the conditional clauses were analyzed. This is essential information if the disease is to be managed effectively.

Heuristics vs. Algorithms

The use of rules alludes to another difference between expert systems and conventional programs. Conventional programs operate according to algorithms, whereas expert systems employ heuristics. Algorithms are formal procedures designed to produce correct or optimal solutions. They can be procedures for solving mathematical problems but are not limited to numerical data. For example, the algorithm for plant disease diagnosis is Koch's postulates. Confirmation of Koch's postulates for a certain isolate establishes the etiology of the disease. Performance of the algorithm establishes proof, or a guarantee, that the isolate in question is indeed pathogenic. Employing heuristics usually results in the correct conclusion but cannot offer the same guarantee as the algorithm. The expert system rules actually are heuristics, i.e., they embody judgmental knowledge, rules of thumb, or simplifications used by experts for lack of an appropriate algorithm. Figure 3 compares the diagnostic algorithm with the heuristic rule for diagnosing bacterial wilt of muskmelon.

algorithm	A formal procedure guaranteed to produce a correct or optimal solution.
artificial intelligence	A subfield of computer science that studies how machines might behave like people.
domain expert	A person with great experience and proficiency at solving problems in a certain field.
expert systems	Computer programs that emulate the problem solving behavior of a human expert.
heuristics	A problem solving technique that employs simplifications and rules of thumb to achieve appropriate solutions.
inference mechanism	That part of a knowledge-based system that contains the general problem solving logic (also called inference engine).
knowledge base	A collection of facts and rules that comprise all relevant information in a certain domain.
knowledge engineer	A person who designs and builds the expert system.
knowledge representation	The process of structuring information and knowledge about a problem in a way that the problem will be solved efficiently.
parameter	A variable that helps describe the nature of the problem. For example, in a plant disease diagnostic system, the "plant symptom type" would be a likely parameter.
rule	A formal way of specifying a directive expressed as IF "condition" THEN "action".

Fig. 1. Glossary of terms.

The Inference Mechanism

The other major component of an expert system, the inference mechanism (or inference engine), is the part containing general problem-solving logic. The concept of the inference mechanism often causes confusion because there is no simple, general way to characterize the mechanism and because many possible methodologies can be followed in constructing it. Fundamental to the power and structure of expert systems is the separation of knowledge (expertise) from the general problem-solving logic. Separate maintenance of these components is somewhat analogous to the separation of a computer program from its data. A program designed to calculate the mean of a column of numbers can be used as an example. By maintaining the data separate from the program, more data can be added, data can be changed, and different data sets can be substituted for the original without affecting the operation of the program. Likewise, in an expert system, rules can be edited, removed, and added without interrupting the logic that enables the knowledge base to be thoroughly searched for the solutions to a problem.

In our rule-based system, the inference mechanism searches through the knowledge base, applies rules to the current data, and accumulates new information. Figure 4 shows how the inference mechanism operates in the context of powdery mildew diagnosis and control. The two objectives are to diagnose the disorder and to recommend a control action. The left side of Figure 4 illustrates the symptoms of powdery mildew and lists several rules for diagnosing different diseases. The inference mechanism searches through the IF portion of the rules until it finds a match with the current data. The matched rule then executes and concludes that the disease is powdery mildew. This conclusion then becomes new information to be added to the store of current facts. To achieve the second objective, the inference mechanism searches all rules that deal with control in the THEN portion of the rule (middle section of Figure 4). The mechanism executes the rule whose IF portion matches the current data (powdery mildew). The conclusion is to recommend the application of fungicide X (right side of Figure 4).

A

IF the plant symptom is a leaf spot,
and the shape of the leaf spot is circular,
and the color of the leaf spot is brown,
and concentric circles occur within the brown areas;

THEN the disease is Alternaria leaf blight.

Fig. 2. (A) The IF-THEN rule used for diagnosis of Alternaria leaf blight of muskmelon and (B) symptoms of the disease.

B

The pathogen must be associated with diseased plants.
The pathogen must be isolated, identified, and grown in pure culture on nutrient media.
Healthy plants must be inoculated with the pathogen from the pure culture.
The symptoms expressed must be the same as those observed initially.
The pathogen must be reisolated from diseased plants.

C

IF the plant symptom is wilt,
and the wilt is rapid,
and no yellow tissue is associated with wilted leaves,
and bacterial streaming can be demonstrated from freshly cut stems,

THEN the disease is bacterial wilt.

Fig. 3.(A) Symptoms of bacterial wilt of muskmelon, (B) the algorithm for diagnosing bacterial wilt, and (C) the heuristics for diagnosing bacterial wilt.

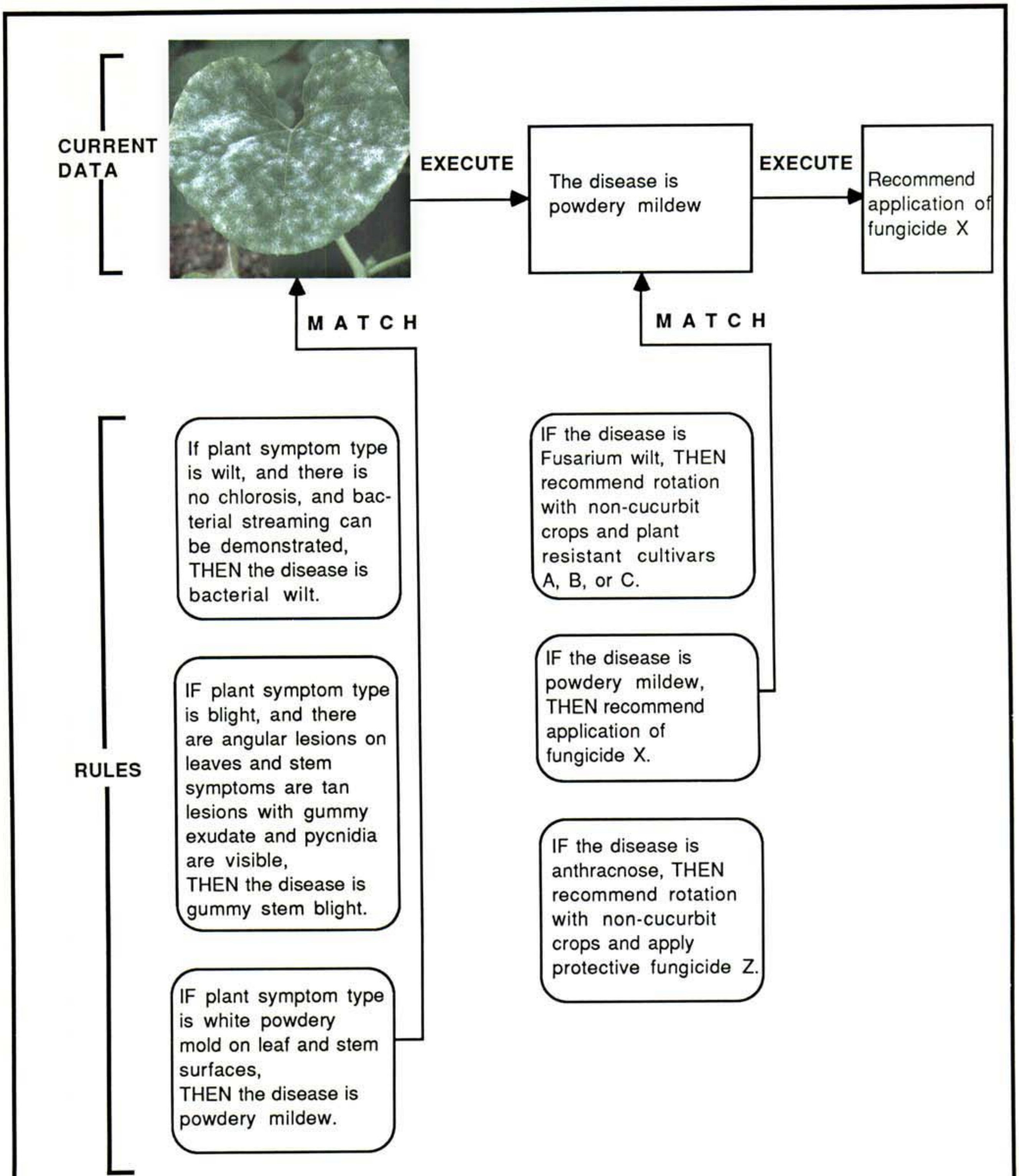


Fig. 4. A simplified example of how the inference mechanism operates in the diagnosis of powdery mildew of muskmelon and recommendation of a mildew control action.

The essence of the inference mechanism is in its capacity to search and execute the rules. Because it is maintained separately from the knowledge base, new rules can be added without interfering with the operation of the system. The type of inference method where the IF portion of a rule is matched against current data to establish new information and a pathway to the goal of the system is called forward chaining. Another method, called backward chaining, starts with its conclusion and then tries to establish the facts it needs to prove that conclusion. The nature of the problem determines the type of inference method preferred.

Explanations, Incomplete Data

Another characteristic inherent to expert system software is the ability to explain its actions. After all, who will trust the diagnosis and recommended therapy of an expert who cannot explain the basis on which they were made? In our rule-based system, each time a rule is used the system records which rule it was and what was being proved or deduced by it. A special command menu that appears on the screen of the computer monitor offers a "how" selection that can be used to ask the system how it arrived at a certain conclusion. Such explanations are available because the knowledge base is separate from the inference mechanism and because knowledge base rules are in an IF-THEN form (i.e., given the THEN part, the system always knows what the IF part is). A conventional program normally cannot explain what rules it used, and the programming tasks to make it do so would be overwhelming.

A final distinguishing feature of expert systems that underscores their utility and special nature is that they can operate with unknown or incomplete data. In response to a prompt for new information, a user can answer "don't know" and still be able to produce a conclusion. The conclusion is likely to be uncertain, but certainty values (100%, 90%, 50%, etc.) based on the best estimate of the human expert can be programmed into the system. Incomplete data usually "do not compute" on conventional systems.

Engineers and Experts, Hardware and Software

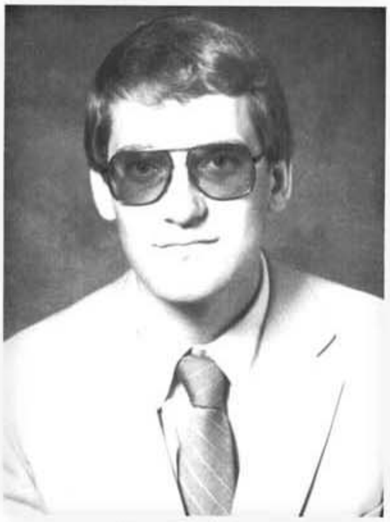
Expert systems are built by skilled individuals with experience in computer programming (knowledge engineers) and other individuals proficient at solving problems within a narrow domain (domain experts). Building an expert system requires frequent and extensive interviews between the domain expert and the knowledge engineer to structure and articulate the domain knowledge into rules. The process includes continual testing of the program and expansion of the knowledge base to include new rules

R. X. Latin

Dr. Latin is an associate professor of plant pathology at Purdue University and has extension and research responsibilities for vegetable crop diseases. He earned his B.S. degree from Waynesburg College and his M.S. and Ph.D. degrees from Pennsylvania State University. His research interests involve epidemiology and control of vegetable diseases.

G. E. Miles

Dr. Miles is an associate professor of agricultural engineering at Purdue University. He earned his B.S. and M.S. degrees from Mississippi State University and his Ph.D. degree from Purdue University. His research interests include development of expert system applications and decision support systems for horticultural crops.



J. C. Rettinger

Mr. Rettinger is a graduate research instructor in agricultural engineering at Purdue University. He received his B.S. degree from Purdue and currently is working toward his M.S. degree in the area of expert system applications in crop production.

for previously overlooked cases. Completion of the project is likely to take months, perhaps years, because the emphasis is on knowledge that has not been formally documented in a public forum and validation is very involved. The focus is on knowledge acquired through experience—rules of thumb (heuristics) that enable a human expert to extrapolate and make educated guesses when necessary and to recognize promising approaches to a problem and deal effectively with incomplete data. This type of knowledge makes an expert system a unique and powerful tool to help solve problems conventional programs are unable to address.

Expert systems are programmed in traditional programming languages such as BASIC, C, and FORTRAN or in languages developed for artificial intelligence applications such as LISP and PROLOG. Traditional languages are disadvantaged in that the types of data they can manipulate are limited to numbers, logical values, and character strings. Their capacity to maintain sufficient memory to handle new or additional data is also limited. LISP and PROLOG are not bound by such constraints and offer additional convenient debugging features and functions to easily check that each procedure performs the intended task and generates the correct result.

Another approach to expert system development is through the use of knowledge engineering languages (also called shells) such as EMYCIN and EXPERT. These are skeletal systems with the inference mechanism already in place. Only the domain-specific knowledge (rules and parameters and their values) is encoded by the knowledge engineers. Skeletal systems greatly facilitate the development of an expert system but lack the flexibility to tailor the system to specific problems. Such systems generally operate on an IBM-compatible personal computer with 640K of RAM and a disk drive. They are affordable developmental tools and greatly facilitate the process of building the expert system.

Applications in Plant Pathology

Many kinds of problems in plant pathology are appropriate for solution by expert systems. These problems generally lie in the area of plant disease management but are not limited to the more applied tasks. Regardless of their nature, all problems appropriate for expert system solution share two features: They normally are thought to require a human expert for their solution and they occur repeatedly. A recurring need for a system or program is the common thread running through all computer applications. Expert system development and computer programming are expensive, and the application must be employed often by many users to justify the cost.

Disease diagnostic problems are likely targets for developers of expert systems because they require a human expert and occur repeatedly. The clinical expertise essential for accurate diagnosis is acquired through exposure to numerous specimens, i.e., the type of knowledge readily employed in an expert system. An advantage expert systems have over descriptive texts and bulletins is elimination of random searching for possible solutions. Expert systems also may include subtle diagnostic keys not found in textbooks and give more weight to some observations than others. One of the first and most widely used expert systems, MYCIN, is designed to diagnose and prescribe treatment for human bacterial blood infections (8). In plant pathology, diagnostic systems will be especially useful for rapid diagnosis of disorders of a crop with a high unit value and for which several control tactics are available. The foundation of our prototype is a system for identifying disorders of muskmelon, a valuable crop threatened by many diseases each year. Expert systems also can be used as training and educational tools. Michalski et al (6) developed PLANT/ds to provide consultation on the diagnosis of soybean diseases. Their rule-based system is part of a general decision support program for managing crop diseases and insect damage.

Disease-forecasting problems also are candidates for expert system solution. The SEPTORIA expert system developed by Sands et al (7) uses disease projections and environmental data to estimate yield losses caused by Septoria nodorum blotch. The yield loss information then is used to determine whether a fungicide application is justified. In fields related to plant pathology, expert systems are utilized as decision support systems for grain marketing (9) and for comprehensive cotton crop management (4). These are only a few of the expert systems in operation today (2,3,5,10,11).

Who Will Use Expert Systems?

One of the most exciting features of expert system development is the availability of this very sophisticated computer technology for immediate practical use by the entire agricultural community. There is a great need for reliable decision support systems for many phases of crop management, including disease diagnosis and control. Land-grant universities need to look only as far as their extension efforts in the areas of crop and pest management to find useful applications for expert systems.

Extension specialists in plant pathology once were relied upon to diagnose and prescribe controls for crop disorders at individual farms. Reductions in the number of these crop diagnostic experts and the concomitant need to distribute information via printed bulletins and mass media have curtailed their field activities at a time when the need for their expertise is growing because of rising production costs and shrinking profit margins. The reduction in on-farm consultations by specialists creates a niche that may eventually be filled by county agents or private crop consultants, but it will take years for those indivduals to acquire the problem-solving experience of a particular crop specialist. Commercial growers will benefit from expert systems substituted for the travel-limited specialist. Immediate attention to their problems by expert systems will avoid costly delays as well as the frustrations associated with multiple phone calls to elusive specialists. Extension specialists will reclaim time once spent traveling to pursue other extension activities and conduct developmental research in areas important to production agriculture. Because the knowledge base can easily be amended to include new information generated by the research, the system can be kept as current as the human expert.

Expert systems applied to disease management problems will be useful to any facet of agriculture that deals with disease control products or service. Such systems are expected to be attractive to an agrichemical industry experiencing staff reductions. Sales and development personnel are being required to represent types of products for which they have little or no education or experience. An expert system operable on a personal computer will provide the problem-solving support needed by these representatives. Traditional agribusinesses already use and promote computer software customized for applications in production agriculture. These conventional programs provide the user with information. Expert systems can offer experience!

The educational utility of expert systems cannot be overstated. With repeated use of the system, the user becomes familiar with the organization of rules and information in the knowledge base. Also, because the system can explain how certain conclusions were drawn, the logic employed by the expert is impressed upon the user, bringing performance closer to that of an expert. The systems can be used to demonstrate the epidemiologic basis for pest management recommendations or to teach students how to structure information for diagnosing diseases. The possibilities for application of expert systems in the classroom are limited only by our imaginations.

Perspective

Although expert systems hold promise for various applications in plant pathology, they are not for everyone. Expectations raised for their development should be tempered by the realities of their cost. A problem addressed by an expert system should be truly meaningful, and the solutions offered by the system must be useful to justify the cost. The systems also must work! The reliability of expert systems developed with certain tools is under scrutiny (1). Solutions must be sufficiently accurate so the user will have confidence in decisions made by the system. This should be obvious, but a quick accounting of the number of idle computerized disease-forecasting programs shows that the need for low-risk performance was not a high priority in their development. Expert systems must perform so well that users will purchase them. Income from the systems will be needed to underwrite their continued development and maintenance costs.

The potential for application of expert systems in plant disease management is almost unlimited. These sophisticated programs that capture the judgmental knowledge of a human expert can serve all sectors of the agricultural community that influence disease control decisions. A final benefit of expert system development is gained by the system builders. Programming our knowledge forces us to define our problem-solving behavior clearly. This can be an enlightening experience that also forces us to put our theories to the test.

Literature Cited

1. Denning, P. J. 1986. Expert systems: Artificial intelligence vs. real intelligence. Am. Biotechnol. Lab. 4:6-8.
2. Duda, R. O., and Shortliffe, E. H. 1983. Expert systems research. Science 220:261-268.
3. Huggins, L. F., Barrett, J. R., and Jones, D. D. 1986. Expert systems: Concepts and opportunities. Agric. Eng. 67:21-23.
4. McKinion, J. M., and Lemmon, H. E. 1985. Expert systems for agriculture. Comput. Electron. Agric. 1:31-40.
5. Michaelsen, R. H., Michie, D., and Boulanger, A. 1985. The technology of expert systems. Byte 10:303-312.
6. Michalski, R. S., Davis, J. H., Bisht, V. S., and Sinclair, J. B. 1983. A computer-based advisory system for diagnosing soybean diseases in Illinois. Plant Dis. 67:459-463.
7. Sands, D. C., Sharp, E. L., Scharen, A. L., Slater, L. S., and Caristi, J. 1986. An expert system for predicting crop disease epidemics. (Abstr.) Phytopathology 76:1083.
8. Shortliffe, E. H. 1976. Computer Based Medical Consultation, MYCIN. Elsevier, New York. 264 pp.
9. Thieme, R. H., Uhrig, J. W., Peart, R. M., Whittaker, A. D., and Barrett, J. R. 1987. Expert system techniques applied to grain marketing analysis. Comput. Electron. Agric. 1:299-308.
10. Thompson, B. A., and Thompson, W. A. 1985. Inside an expert system. Byte 10:315-330.
11. Waterman, D. A. 1985. A Guide to Expert Systems. Addison-Wesley Publishing Company, Inc., Reading, MA. 419 pp.