

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

### Theses

---

4-8-1987

## Psyxpert: An Expert system for aiding psychiatrists in the diagnosis of psychotic disorders

Mary A. Overby

Follow this and additional works at: <https://repository.rit.edu/theses>

---

### Recommended Citation

Overby, Mary A., "Psyxpert: An Expert system for aiding psychiatrists in the diagnosis of psychotic disorders" (1987). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

ROCHESTER INSTITUTE OF TECHNOLOGY  
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

PSYXPERT:  
AN EXPERT SYSTEM FOR AIDING PSYCHIATRISTS  
IN THE DIAGNOSIS OF PSYCHOTIC DISORDERS

by  
Mary A. Overby

A thesis, submitted to  
The Faculty of the School of Computer Science and Technology,  
in partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science

APPROVED BY: John A. Biles

J.A. Biles

Walter Wolf

Walter Wolf

Peter H. Lutz

Ex Officio

Peter Lutz

Title of thesis:

"Psyxpert: an Expert System For Aiding Psychiatrists in the  
Diagnosis of Psychotic Disorders"

I, Mary A. Overby, hereby grant permission to the  
Wallace Memorial Library, of Rochester Institute of  
Technology, to reproduce my thesis in whole or in part. Any  
reproduction will not be for commercial use or profit.

**Mary A. Overby**

Mary A. Overby  
April 8, 1987

## Abstract

Psyxpert is an expert computer system designed to aid psychiatrists in the diagnosis of mental disorders when psychotic features are the prominent part of the presenting clinical picture. The knowledge base contains psychiatric knowledge in the form of production rules. The system uses a backward-chaining control strategy to guide the consultation. Psyxpert provides a menu-driven user interface and an explanation subsystem. The system uses certainty and importance measures to produce a diagnosis with an attached certainty factor and recommendations for further evaluation or therapy. Psyxpert is written in Virginia Tech HC Prolog and runs on Digital Equipment Corporation's VAX 11/780 under the VMS operating system.

## KEY WORDS

Expert systems   Artificial intelligence   Prolog  
Knowledge engineering   Psychiatric disorders

## COMPUTING REVIEW SUBJECT CODES

Artificial Intelligence	I.2
Medical Application	J.3

## Table of Contents

1.	Introduction	
1.1	Problem Statement .....	1
1.2	Summary .....	1
2.	Background	
2.1	The Expert Systems Field .....	5
2.2	Expert Systems in Medicine .....	14
3.	The Domain: Psychiatry	
3.1	Psychiatric Diagnosis .....	20
3.2	The Mental Status Examination and Report ...	21
3.3	The DSM-III and the DSM-III Casebook .....	26
4.	Prolog .....	33
5.	The Psyxpert System	
5.1	Consultation Overview .....	39
5.2	System Organization .....	49
5.3	Implementation Details .....	51
5.4	An Example Consultation .....	68
6.	Knowledge Acquisition .....	84
7.	Results and Discussion	
7.1	Results .....	89
7.2	Limitations and Restrictions .....	92
7.3	Possible Enhancements .....	94
8.	Conclusions .....	97
	References .....	100
	Bibliography .....	102
	Appendix A -- Standard Prolog vs. HC Prolog .....	A-1
	Appendix B -- Test Case #85 "Miriam and Esther" ....	B-1
	Appendix C -- Log Run .....	C-1
	Appendix D -- Prolog Code Listing .....	D-1

## 1. Introduction

### 1.1 Problem Statement

Psyxpert is an expert computer system prototype designed to aid psychiatrists in the diagnosis of mental disorders when psychotic features are the prominent part of the presenting clinical picture. The knowledge base contains psychiatric knowledge in the form of production rules. The system uses a backward-chaining control strategy to guide the consultation. Psyxpert provides a menu-driven user interface and an explanation subsystem. The system uses certainty and importance measures to produce a diagnosis with an attached certainty factor and recommendations for further evaluation or therapy. Psyxpert is written in Virginia Tech HC Prolog and runs on Digital Equipment Corporation's VAX 11/780 under the VMS operating system.

### 1.2 Summary

Chapter 2 introduces the field of expert systems. The chapter is intended to orient the reader not knowledgeable in Artificial Intelligence or expert systems to the field of expert systems research.

In section 2.1, the roots of expert systems in Artificial Intelligence (AI) are traced. Three methods of representing expert knowledge are described: production rules, semantic networks and frames. The problem of

reasoning with uncertainty and three techniques used to handle it are introduced: Bayes' Theorem, Zadeh's Fuzzy Set Theory, and MYCIN's Certainty Factors. Finally, the process of acquiring expert knowledge is discussed.

In section 2.2, expert systems in the medical domain are introduced. The first four Artificial Intelligence in Medicine (AIM) systems - CASNET, MYCIN, INTERNIST and PIP - are described in terms of method of knowledge representation, control strategy and implementation language. The systems that have grown out of the MYCIN project - EMYCIN, PUFF, CENTAUR, WHEEZE, VM, GUIDON, NEOMYCIN and ONCOCIN - are briefly mentioned to orient the reader to current research.

Chapter 3 presents an introduction to psychiatric diagnosis, the mental status examination, the Diagnostic and Statistical Manual (DSM-III), and the DSM-III Casebook. This chapter is intended to provide a basic background on psychiatry for the reader unfamiliar with the field. In section 3.1, the problems and purposes of diagnosis of mental disorders are presented. In section 3.2, the mental status examination used in this research is described. In section 3.3, the DSM-III and the DSM-III Casebook are introduced, and several mental disorders are described.

Chapter 4 is an introduction to the programming language Prolog. The mathematical theory of Predicate Calculus, on which Prolog is based, is introduced. An example of a simple Prolog program is presented and used to illustrate features of Prolog. Psyxpert was implemented in HC Prolog, a nonstandard version developed at Virginia Polytechnic

Institute and State University. Appendix A describes the differences between standard Prolog and HC Prolog.

Chapter 5 presents the Psyxpert system. In section 5.1, an overview of the consultation is presented to show the reader what a consultation is like. In section 5.2, the system's logical organization is defined in terms of the knowledge base, the user interface and the explanation subsystem. In section 5.3, the implementation details of the system are described. Section 5.3.1 describes the translation of the DSM-III diagnostic criteria into production rules written in Prolog and the system's method of representing and reasoning with uncertainty. Section 5.3.2 and 5.3.3 describe the implementation details of the user interface and the explanation subsystem respectively. In section 5.4, an example consultation is presented to illustrate the features and implementation details of the Psyxpert system.

Chapter 6 discusses the acquisition of knowledge in the Psyxpert system. The three sources of knowledge used to build the Psyxpert system are presented. These sources are the DSM-III diagnostic criteria, the AMSIT mental status report format, and Dr. Daniel Pearson, PhD., M.D. The process of acquiring the knowledge used in the Psyxpert system is described.

Chapter 7 presents the results of the Psyxpert research project and a discussion of the system's limitations and restrictions, as well as, possible enhancements. In section 7.1, the results from 16 test cases are described. In



section 7.2, the limitations and restrictions of Psyxpert are discussed. In section 7.3, the following possible enhancements and the changes necessary to implement them are briefly described: extended domain, knowledge acquisition interface, natural language output, alternate consultation modes, and a data storage and retrieval system.

Chapter 8 presents the conclusions drawn from the results of the Psyxpert research project. Psyxpert is examined from the following positions: as a Psychiatric system, as a system implemented in Prolog, as an academic learning experience and as a clinically effective tool.

## 2. Background

### 2.1 The Expert Systems Field

#### 2.1.1 Overview

Artificial intelligence (AI) is the branch of computer science which deals with the intelligent behavior of computers. Computer science programs that perform tasks requiring intelligent human thought are said to use artificial intelligence [1]. Knowledge engineering is that branch of AI in which a knowledge engineer (someone familiar with expert systems) interacts extensively with a domain expert (someone knowledgeable in a particular field) to build an expert system. An expert system is a high performance computer program that contains extensive knowledge about a narrow, real-world domain. An expert system uses domain specific problem solving strategies or heuristics to reach conclusions [2,3]. An explanation facility is an important component of all expert systems, because it reassures the user of the validity of the chain of inference the system uses to reach its conclusion.

The roots of expert systems in AI can be traced to the 1960's, when researchers were building knowledgeable computer systems. They searched for ways to build general purpose problem solvers [3], but these were inefficient, using too much computer time and space.

Research in the 1970's turned toward solving basic

artificial intelligence problems such as search and knowledge representation. During the late 1970's computer scientists began to realize that knowledge itself was the key to successful implementations. Instead of building computer programs to handle generic situations, computer scientists began to tailor each system to its application.

"Today there has been a shift in paradigm. The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a fashion that permits their effective use and interaction [4]."

The paradigm shift has led to what we have come to know as knowledge-based expert systems [5].

An essential feature of knowledge-based expert systems is the separation of domain knowledge from general problem-solving knowledge. The part of the expert system containing the domain knowledge is known as the knowledge base. The part of the expert system containing the general problem-solving knowledge is known as the cognitive engine or inference engine.

### **2.1.2 Knowledge Representation**

A basic consideration in any knowledge-based expert system is the representation of the knowledge in the knowledge base. The debate between procedural and

declarative representations rages to this day among computer scientists [6]. The basic point of view in the procedural camp is that knowledge consists of information about 'how' we do things. The declarative point of view emphasizes 'what' we know about things. Both are important aspects of knowledge, and the particular application of the knowledge should be the deciding factor. In expert systems, practicality and obtaining the desired results are more important than theoretical arguments. The advantages of the declarative approach are flexibility, modifiability and communicability [6]. Expert systems are built incrementally so flexibility and modifiability are important factors. Expressing knowledge in terms of declarative chunks of information makes the process of encoding an expert's knowledge into a system easier for the developer. This also allows a knowledge acquisition interface to be attached to the system, giving the expert the capability to enter knowledge directly into the system. The explanation subsystem is also easier for the developer to implement and for the expert to understand.

There are several ways to represent declarative knowledge. The production system formalism described by Davis and King [7], semantic nets and frames as described by Minsky [8] will be discussed here, because they have been used most extensively.

A pure production system is composed of production rules (knowledge base), a data base (fact base) and a rule interpreter (cognitive engine). A production rule is a

conditional statement with a left-hand-side (LHS) or antecedent and a right-hand-side (RHS) or consequent. The data base is a collection of symbols. The rule interpreter scans the facts or data base in an attempt to prove one side of the production rule.

Production rules increase the modularity of the chunks of knowledge and make the system easy to modify and extend. The modularity and size of the knowledge chunks also make for an easier implementation of the explanation facility. Production rules make it easy for the expert to describe his or her expertise.

Semantic network is a term used to describe a knowledge representation technique based upon a network of nodes connected by or arcs that describe the relationship between the nodes. Semantic networks are useful in domains in which taxonomies can simplify problem solving. Semantic network representations may also utilize inference rules to enhance their representational power and are very useful for representing complex definitional and relational knowledge. There are several disadvantages to semantic networks: explanations of the system's chain of reasoning are hard to construct; inferences across the semantic network may consume large amounts of computer time; and some types of knowledge cannot be readily expressed as relationships in a natural manner [9]. Woods [10] presents a thorough discussion of semantic networks.

A frame is a data structure used to describe prototypical situations, objects, concepts or events. A

frame is a special type of semantic network in which each node is associated with a set of common attributes called slots. Each slot has restrictions on its contents and may have default values associated with it. For example, in a frame representation the object dog might have slots for breed, color, size, sex and age. Frames allow procedural knowledge to be incorporated with declarative knowledge. Slots may contain procedures that are invoked under certain circumstances. Frames make it difficult to provide explanations of the chain of reasoning, and they may not be a natural way for the expert to express his or her knowledge [3,9]. Minsky [8] presents a thorough discussion of frames.

### 2.1.3 Control Strategies

The cognitive engine (CE) is the part of a knowledge-based expert system that contains the general problem-solving knowledge and is the active part of the expert system. The CE applies the knowledge in the knowledge base to the facts about a given situation to produce a line of reasoning that reaches some conclusion. The CE thus is in control of the expert system.

There are primarily three control strategies that the CE may use: data-driven, goal-driven, or a combination of the two. In a data-driven control strategy the data that are known are applied to the knowledge base to reach a conclusion. In production rule methodologies this is known as forward-chaining. In a goal-driven control strategy the

system applies the knowledge base to the goal to produce new subgoals and continues in this manner until primitive goals that are known to be solvable are reached. In production rule methodologies this is known as backward-chaining. Data-driven and goal-driven strategies may be combined by forward-chaining from the initial conditions and backward-chaining from the goal until a common middle ground is reached [9].

The difference between a forward-chaining and backward-chaining control strategy is in the way the rules and the data are searched [3]. In a forward-chaining system the data in the data base drives the firing of the rules. The data base is matched against the rule base to determine which, if any, rule can fire. The firing of a rule results in the consequent being added to the data base, which may then cause another rule to fire. In a backward-chaining system the consequent, or goal, drives the firing of the rules. An attempt is made to prove the goal by matching the antecedent with the data base. If the antecedent cannot be matched, it becomes the new goal, and an attempt is made to prove it by trying to fire another rule. The procedure repeats recursively until all of the rules in the chain of reasoning are proved, or the system cannot complete the inference and failure results.

#### **2.1.4 Reasoning With Uncertain Knowledge and Information**

Expert systems must be able to reason with information

that may be neither completely true nor completely false. Sometimes information is missing or uncertain; therefore, a method for reasoning with uncertain or incomplete information must be incorporated into an expert system [2].

Early attempts at computer simulation of expert decision making resulted in computer programs that relied on statistical decision theory using Bayes' Theorem to calculate the probability of a given conclusion. An example from the domain of medicine is given below, although Bayes' Theorem has been used in other domains as well.

"The medical diagnostic problem can be viewed as the assignment of probabilities to specific diagnoses after analyzing all relevant data. If the sum of the relevant data (or evidence) is represented by  $e$ , and  $d(i)$  is the  $i$ th diagnosis (or "disease") under consideration, then  $P(d(i)|e)$  is the conditional probability that the patient has disease  $i$  in light of the evidence  $e$ .

Bayes' Theorem is useful in these applications because it allows  $P(d(i)|e)$  to be calculated from the component conditional probabilities:

$$P(d(i)|e) = \frac{P(d(i)) P(e|d(i))}{\sum_j P(d(j)) P(e|d(j))}$$

In this representation of the theorem,  $d(i)$  is one of  $n$  disjoint diagnoses,  $P(d(i))$  is simply the a priori probability that the patient has disease  $i$  before any evidence has been gathered, and  $P(e|d(i))$  is the probability that a patient will have the complex of symptoms and signs represented by  $e$ , given that he or she has disease  $d(i)$ [11]."

Bayes' Theorem requires large amounts of data about diseases and symptoms, which may not be available in new or



less well understood areas of medicine. Furthermore, diseases and their symptoms may change over time making old statistics unreliable, and disease categories themselves may change, reflecting new treatment procedures. All of the aforementioned problems have led scientists to search for new ways to incorporate uncertainty in the decision making process.

Zadeh's Fuzzy Set Theory [12] uses mathematical set theory notation to quantify concepts that are imprecise. Fuzzy logic uses laws of inference to reason with fuzzy sets. For example the statement, 'X is a large number' may be characterized by the fuzzy set  $X = (0, 10)$  with the possibility .1,  $X = (10, 10,000)$  with the possibility .2 and  $X > (10,000)$  with the possibility .7 [2]. Zadeh's work is primarily theoretical, but Duda, Gaschnig and Hart incorporate aspects of Fuzzy Set Theory in their Prospector Consultant System for Mineral Exploration [13].

Researchers have attempted to find ways to more closely mimic the way in which experts actually make decisions. Experts use combinations of judgmental knowledge "chunks" based upon past experiences and general principles from their domain of expertise to reach conclusions. The MYCIN system (a medical consultation system for diagnosis and treatment of infectious diseases) uses a concept of certainty factors (CF's) to gather evidence for or against a hypothesis [11,14]. Certainty factors are based on confirmation theory but were developed outside of theoretical bounds and have resulted in a practical implementation of reasoning with

uncertainty.

A certainty factor is a number between -1 and +1 that reflects the degree of belief in a hypothesis. When the CF = +1, the hypothesis is known to be correct; when the CF = -1, the hypothesis is known to be incorrect, and when the CF = 0, there is no evidence confirming or disconfirming the hypothesis. Certainty factors are used in two ways. They are used to represent the belief in the evidence that has been accumulated, and they are used in the decision rules themselves to represent the importance of a fact to a given hypothesis. Certainty Factors are calculated for each hypothesis generated and used to choose among competing hypotheses.

#### **2.1.5 Knowledge Acquisition**

Knowledge acquisition is the process of assimilating knowledge from a specific domain into a form that can be used by a system's reasoning mechanism to arrive at a solution to a problem. Domain knowledge may come from diverse sources, such as experts themselves or textbooks and journals. The process of translating knowledge from the source to the program may be performed by the knowledge engineer or, in limited cases, by a program [2]. Some systems may provide a knowledge acquisition interface or intelligent editing program through which the expert may enter his or her knowledge directly into the system.

Acquiring knowledge from an expert in the field is a

major undertaking filled with problems and hurdles. The success of a project depends upon how well the expert is able to verbalize his or her knowledge to the knowledge engineer. The expert's experience and attitudes about computers can affect the project also.

Acquiring knowledge from textbooks and journals has the advantage that the information has already been "verbalized". On the other hand, textbooks and journals cannot give feedback on the solutions resulting from the rules that were written using their knowledge. An expert must be available to provide feedback on the computer's solution.

The knowledge acquisition process proceeds incrementally from a rapidly-designed prototype to a system that performs at an acceptable level. Knowledge in an expert system is added and tested a little at a time to insure accurate performance. The expert is consulted about all results and modifications. The system may need to be repeatedly revised, redesigned and retested until an acceptable level of performance is achieved.

## **2.2 Expert Systems in Medicine**

Medical consultation is an important area of artificial intelligence research.

"Expert medical consultation is a scarce, expensive, yet critical component of any health care system. Making the knowledge and expertise of human experts more widely available through computer consultation systems has been recognized as an important mechanism for improving

the access to high quality health care  
[15]."

In the 1970's the Biotechnology Resources Program of the Division of Research Resources of the National Institute of Health began to support research into the use of artificial intelligence in medicine (AIM). The first four AIM systems developed, CASNET, MYCIN, INTERNIST and PIP, differed from earlier statistical approaches to medical reasoning. These systems share the central features of expert systems such as a separation of domain-specific knowledge from control strategies, a facility for explanation and a method for reasoning with uncertainty. Yet these systems operate in different domains of medicine and use different knowledge representation techniques, methods of reasoning with uncertainty and control strategies.

CASNET, implemented in FORTRAN and developed at Rutgers University, is a consultation system that diagnoses and prescribes therapy for glaucoma. The system represents knowledge about the development of glaucoma through the use of a particular type of semantic network known as a causal association network. CASNET uses a data-driven control strategy to guide the consultation and weighted inferential links to reason with uncertainty. Explanations are provided by references to medical literature. CASNET has been tested with many cases and was found to perform at an expert level [15,3].

MYCIN, implemented in LISP and developed at Stanford University, assists physicians in the diagnosis and treatment

of infectious disease. The system uses production rules to represent domain knowledge, a backward-chaining control strategy, and a model of inexact reasoning to handle uncertain or incomplete information. The system provides explanations about the system's line of reasoning, which is made easier because of MYCIN's emphasis on the modular nature of its knowledge.

"There has been a formal evaluation of the MYCIN system by a number of independent consultants, which demonstrated that the program performed at a level comparable to experts [15]."

INTERNIST, implemented in LISP and developed at the University of Pittsburgh, assists the physician in making diagnoses in the area of internal medicine. Internist, renamed CADUCEUS, uses a semantic network to represent its knowledge of more than 500 diseases, making it one of the largest medical expert systems developed. The system uses both data-driven and goal-driven control strategies to guide the consultation. CADUCEUS uses uncertainty weights associated with the network links and expressed on a scale from 1 to 5 for minimum to maximum confirmation of its hypotheses. CADUCEUS' explanation facilities are minimal. The system has been tested with complex cases from clinical case reports in the major journals [15,2] and has proved quite successful.

PIP (Present Illness Program), implemented in CONNIVER and developed at MIT, contains knowledge about kidney disease. PIP uses frames to represent knowledge. Initially,

PIP uses a data-driven control strategy to suggest hypotheses that are then used to select questions that discriminate between them. The frames contain certainty rules that calculate a score for each hypothesis representing how closely the patient's findings match the hypothesized disease. The hypotheses are then ranked according to their scores. PIP is an experimental system, and testing has revealed problems in its lines of reasoning [15].

New research is being carried out based upon lessons learned from these early systems. General purpose expert system tools or shells have been built from early expert systems. These general purpose shells contain the inference engine and support facilities with the domain specific knowledge removed. EXPERT and EMYCIN are examples of general purpose expert systems shells. EXPERT is an outgrowth of CASNET and has been used to build systems in rheumatology, neurophthalmology and endocrinology. EMYCIN, Essential MYCIN, has been used to construct a wide range of systems in medicine as well as in engineering, tax and estate planning, geology and software development [16].

Many medical consultation systems have grown out of the MYCIN and EMYCIN projects. These systems have developed from a variety of motivations. Some systems vary the domain while holding the architecture constant; others hold the domain constant while varying the architecture. Other systems address issues in expert systems research such as monitoring over time, intelligent computer-aided instruction and the integration of computer consultants into clinical

settings [11].

PUFF is a system that interprets measurements of pulmonary function and produces a diagnostic report on the presence and severity of pulmonary disease. PUFF is implemented in EMYCIN and has reached the stage of a production prototype, a system operating in the user environment that is fast, reliable and efficient [3]. PUFF is being used at the Pacific Medical Center in San Francisco, and in a 144-case evaluation there was a 91% rate of agreement between the system's diagnoses and the diagnoses of the physiologist [17].

CENTAUR and WHEEZE are systems that operate in the same domain as PUFF, but their architectures are different. CENTAUR incorporates a frame based knowledge representation scheme with production rules [18,19]. The system uses frames, called prototypes, to characterize the typical features of each pulmonary disease. The use of prototypes in combination with production rules has shown the following advantages: (1) the system's line of reasoning more closely follows the physicians's line of reasoning. (2) The order in which questions are asked can be controlled by making the knowledge explicit in a special slot within the frame. (3) Irrelevant questions are not asked and only hypotheses suggested by the initial data are explored. (4) Inconsistent or erroneous data that cannot be accounted for are indicated [11].

WHEEZE carries the work on CENTAUR one step further by using a uniform knowledge representation structure and a

varied control strategy. All PUFF's production rules have been translated into frame structures. The system uses an agenda mechanism to combine goal-driven and data-driven control strategies. WHEEZE's diagnostic performance is comparable to PUFF's and thus shows that the frame representation is capable of expressing all of the domain knowledge previously expressed in production rules. Furthermore, the combination of goal-driven and data-driven control strategies more closely mimics the diagnostic behavior of the expert physician [11].

VM, Ventilator Manager, is an on-line system used in a hospital's intensive care unit to monitor patients who need mechanical assistance with breathing. VM addresses the issue of interpretation of multisensor data over time and uses an extension of MYCIN's production rule methodology [11,20]. The GUIDON and NEOMYCIN systems address the issue of intelligent computer-aided instruction by extending or rewriting MYCIN's knowledge base of production rules [11,21].

The latest work to come out of the MYCIN research is the ONCOCIN system, which assists physicians in treating and managing cancer patients undergoing chemotherapy. The system uses MYCIN-like production rules and a combination data-driven/goal-driven control strategy. The impetus behind the ONCOCIN project was to design and build a system that would be totally integrated into a clinical setting. This goal has been partially achieved, and expectations are that total integration will be achieved in the future [11].



### 3. The Domain: Psychiatry

#### 3.1 Psychiatric Diagnosis

Psychiatry is the branch of medicine that deals with the origin, diagnosis, prevention and treatment of mental disorders. A mental disorder is an illness with mental or behavioral manifestations and/or impairment in functioning due to social, psychologic, genetic, physical/chemical or biological disturbance [22]. There are a few mental disorders in which the etiology (cause) and the pathophysiology (mechanism) are known, but for most mental disorders there is no general concensus, and a variety of theories have been proposed. Therefore, the diagnosis of mental disorders is essentially a classification problem [23].

Diagnosis in psychiatry serves four main purposes. First, a diagnosis of mental disorders is made in an attempt to communicate more reliably and effectively about certain classes of problems. Secondly, diagnosis implies an understanding of a pattern of illness suggesting a specific treatment and prognosis (outcome). Thirdly, diagnostic categories are important for research, because uniformity may help uncover common causes or patterns [24]. Fourthly, diagnostic categories are necessary to compare the usefulness of types of treatment.

Diagnosis involves organizing, into syndromes, a set of symptoms and signs drawn from the patient's history, the

physical examination, and the mental status examination. A symptom is a subjective complaint made by the patient, and a sign is a piece of objective physical evidence uncovered by the examiner. A syndrome is a set of symptoms and signs that occur in a recognizable pattern. A disorder is a set of symptoms and signs that is more specific than a syndrome in terms of the course of the illness, the history before the illness and the family pattern of illness. The same syndrome can occur in different disorders. A disorder is different from a disease, in which the etiology and pathophysiology are known [24].

### **3.2 The Mental Status Examination and Report**

A mental status examination is the process of estimating psychological and behavioral function by observing the patient, eliciting his description of self and formally questioning him [22]. The mental status examination is used to assess the patient's current orientation, attention, feeling states, thought patterns and specific cognitive skills [24]. The mental status of the patient is reported in a series of narrative statements. The mental status report used at the University of Texas Health Science Center at San Antonio is known as the AMSIT. The AMSIT is a systematic report, which is written after the interview is completed, and it is not to be used to guide the interview explicitly. AMSIT is an acronym for the following descriptive categories: A stands for appearance; M for mood and affect; S for

sensorium; I for intelligence and T for thought processes. A section is written by the examiner concerning each descriptive category.

In the appearance section, the clinician describes observations about the patient's general appearance, behavior and speech.

In the second section the patient's mood and affect are described. Mood refers to a sustained or dominant emotion. A statement should be made concerning which of the following seven possibilities best describes the patient's mood: severely depressed; moderately depressed; mildly depressed; euthymic (normal); mildly elated; moderately elated or severely elated.

Affect refers to an immediately expressed and observed emotion. During the course of the interview a patient's affect should fluctuate with what the patient is currently saying. When describing affect, the range, intensity, lability and appropriateness are described. The range refers to the extent to which both emotional highs and lows are present in the interview. The patient's range of affect may be described by one of the following seven possibilities: markedly decreased; moderately decreased; mildly decreased; normal; mildly increased; moderately increased and markedly increased. The intensity of affect is the amplitude of emotional expression and can be described by one of the seven possibilities used to describe the range of affect. Lability of affect is the tendency to show rapidly fluctuating emotional states. A statement indicating whether or not the

patient's affect is labile should be included.

Appropriateness of affect refers to whether the emotion at a particular moment is the one expected for the patient's currently expressed thought. A statement indicating whether the patient's affect is appropriate at all times during the interview should be included.

In the section under sensorium the patient's orientation, memory and calculating ability are described. The sensorium may be clear or impaired. If spontaneous comments fail to reveal whether or not orientation and memory are intact, these functions should be tested. Orientation may be tested by asking the patient to state the month, day, year and current location. Memory functions include immediate recall, recent memory, and remote memory. The examiner can test immediate recall and recent memory by naming five unrelated objects, asking the patient to repeat them back immediately and then again ten to fifteen minutes later. The examiner may use a test called serial seven subtractions to evaluate calculating ability. The patient is asked to subtract seven from 100 and seven from the answer and so on until six to eight subtractions have been completed. The patient must have had the skill previously. This also tests intellectual function. Impairment in sensorium may suggest an organic mental disorder.

In the next section the patient's intellectual function is described as above average, average or below average. The patient's intellectual function may be estimated based upon spontaneous comments, tests of general information,

observation of the patient's vocabulary, and the complexity of concepts expressed. The examiner should ask the patient questions that are consistent with the patient's life experiences to test the patient's general fund of information.

In the final section the patient's thought processes are described. This section is important in recognizing the presence of a psychosis, because psychotic patients have characteristic disturbances of thought. There are ten variables that describe the patient's thought processes: coherence; logic; goal-directedness; associations; perceptions; delusions; content; judgement; abstracting ability and insight.

The first four variables concern the patient's form of thought or how the patient's thoughts are organized and expressed. Coherence is the tendency of thoughts to stick together well enough to make sense. Logic concerns whether or not conclusions are based upon rational reasoning. Goal-directedness refers to the ability to express ideas about a subject or goal without being sidetracked by unimportant details. The patient's pattern of associations refer to the way in which the patient moves from one idea to the next. Disordered patterns of associations include flight of ideas, in which the patient rapidly jumps from one thought to the next; loose associations, in which there seems to be no connection between ideas; and blocking, in which the patient stops in mid-sentence as if the ideas were suddenly stopped.

The next three variables: perceptions, delusions and content, refer to the patient's content of thought.

Perception is the conscious mental registration of sensory stimulation. False perceptions include: hallucinations, which are auditory, visual or other sensory experiences in the absence of external stimuli; ideas of reference, in which the patient interprets casual incidents and external events as having a direct reference to himself; illusions, which are misinterpretations of a real external stimulus; depersonalization, which is a feeling of unreality or strangeness concerning one's self; and distortions of body image, in which the patient has an erroneous internal picture of his body or its parts. Delusions are false beliefs that are not amenable to rational explanation or objective contradictory evidence. The content variable deals with other central themes of thought.

Finally , the patient's judgement, abstracting ability and insight are described. Judgement, the ability to make appropriate decisions, should be tested based upon the patient's life experiences. Abstracting ability is the ability to think or perform symbolically and may be impaired in various mental disorders. Abstracting ability may be tested by having the patient explain proverbs which use symbolic imagery. Insight is a correct understanding by the patient of the nature and extent of any mental, behavioral or emotional problems that he may have [25].

### 3.3 The DSM-III and the DSM-III Casebook

The standard diagnostic psychiatric text is the Diagnostic and Statistical Manual of Mental Disorders, Third Edition, (DSM-III), of the American Psychiatric Association (APA) [26]. The DSM-III attempts to classify psychopathology into a series of disorders in a purely descriptive fashion and makes no attempt to account for how the disturbance came about, so it may be used by clinicians of various theoretical orientation [26]. The DSM-I, which appeared in 1952, used terms that reflected theoretical positions concerning etiology, and the DSM-II was written to rectify this. The DSM-III was introduced to update the DSM-II with current knowledge regarding mental disorders and to introduce the concepts of diagnostic criteria and multiaxial evaluation.

The DSM-III was compiled by a task force composed of psychiatric professionals appointed by the APA in 1974. The task force sought the advice of experts in each specific area of psychiatry and worked with the APA and other professional organizations. Committees were formed to make comments and suggestions to the task force, and accommodations were made, when possible, to satisfy all groups concerned. Drafts of the DSM-III were first presented to the Annual Meeting of the APA in 1975 and at all subsequent meetings. A special conference was held in 1976 to examine the "DSM-III in Midstream," and additional diagnostic categories were added. The 1977 draft and successive drafts of the DSM-III were made available to the profession for review. The final draft was

approved in May 1979 at the Annual Meeting of the APA, and in June 1979 the Reference Committee and the Board of Trustees gave their approval.

Two year field trials were conducted during the development process to identify problems, try solutions and demonstrate clinical acceptability and usefulness. The field trials resulted in several changes, but the final results showed a favorable response to the DSM-III by a majority of the participants. The diagnostic reliability of the DSM-III was tested by having pairs of clinicians make independent diagnoses of several hundred patients. These results indicated a greater reliability than found in the DSM-II and are presented in the appendix of the DSM-III.

The DSM-III describes each disorder in terms of essential features, associated features, age at onset, course, impairment, complications, predisposing factors, prevalence, sex ratio, familial pattern, differential diagnosis and diagnostic criteria. The diagnostic criteria are included to provide the clinician with explicit guidelines for making diagnoses. In some cases disorders may be divided into specific types with additional diagnostic criteria.

The mental disorders are grouped into seventeen diagnostic categories, which are hierarchically organized into classes based upon shared clinical features. The hierarchical arrangement, in which a disorder high in the hierarchy may have features found in disorders lower in the hierarchy but not the reverse, can be represented with



decision trees, found in Appendix A of the DSM-III.

The research presented here deals with a diagnostic class of mental disorders that share a common psychotic feature. Psychosis may be defined as a gross impairment in reality testing. Reality testing is the (patient's) ability to evaluate the external world objectively and to differentiate adequately between it and the (patient's) internal world [22]. Disorders in which there are major psychotic features are not grouped together in one diagnostic category, but are found in the following diagnostic categories: Disorders Usually First Evident in Infancy, Childhood or Adolescence; Organic Mental Disorders; Schizophrenic Disorders; Paranoid Disorders; Psychotic Disorders Not Elsewhere Classified and Affective Disorders. The topic of our research is further narrowed by excluding Disorders Usually First Evident in Infancy, Childhood or Adolescence; Organic Mental Disorders, and some of the Affective Disorders. Thus our research concentrates on diagnosis of the following seven disorders: Schizophrenic Disorder; Paranoid Disorder; Psychotic Disorders Not Elsewhere Classified, which are Schizophreniform Disorder, Brief Reactive Psychosis, Schizoaffective Disorder and Atypical Psychosis; and Major Affective Disorder with Psychotic Features.

The essential features of Schizophrenic Disorder include the presence of certain psychotic features or characteristic symptoms involving multiple psychological thought processes, deterioration from previous level of functioning, onset

before age 45 and a duration of at least six months. Some characteristic symptoms include disturbances in the content of thought such as delusions or hallucinations; disturbances in the form of thought, such as loose, blocked or incoherent associations; and disturbances of affect, such as moderately or markedly decreased range and intensity or inappropriateness.

The essential features of Paranoid Disorder are persecutory or jealous delusions not due to any other mental disorder. A persecutory delusion is one in which the central theme is that a person or group is being attacked, harassed, cheated, persecuted or conspired against. A jealous delusion is one in which a person falsely believes that his or her sexual partner is unfaithful.

The essential features of Schizophreniform Disorder are identical to the essential features of Schizophrenic Disorder except that the duration is less than six months but greater than two weeks. Schizophreniform Disorder is not classified as a Schizophrenic Disorder because the prognosis is better and the familial pattern is different.

The essential feature of Brief Reactive Psychosis is a sudden onset of the psychotic symptoms following a severe psychological or environmental stressor. The duration is at least two hours but no more than two weeks with a return to a previous level of functioning.

Schizoaffective Disorder is a very ambiguous disorder and has been used in many different ways in the past. The DSM-III uses it to describe disorders in which a diagnosis

cannot be made between Schizophrenic Disorder, Schizophreniform Disorder, and Major Affective Disorder with Psychotic Features.

Atypical Psychosis is a disorder in which there are psychotic symptoms such as delusions, hallucinations, incoherence, loose associations, markedly illogical thinking, or behavior that is grossly disorganized or catatonic (muscular rigidity or inflexibility); yet the criteria for any specific mental disorder cannot be met.

The essential feature of Major Affective Disorder with Psychotic Features is a mood disturbance not due to any other physical or mental disorder. The mood disturbance may involve depression and/or elation. The patient exhibits psychotic features that are brief in relation to the mood disturbance or are present only with the mood disturbance.

The DSM-III Case Book is a companion text to the DSM-III for those wishing to become familiar with the DSM-III classification system [27]. The Case Book contains brief descriptions of real patients, a discussion of each case and a diagnosis of each case based upon the DSM-III classification scheme. The cases are drawn from the authors' own experiences, from the experiences of well known experts in particular areas of diagnosis and treatment, and from significant historical cases. The cases were reviewed by experts in the classification of diseases (nosologists), and their input on the diagnostic assessments was used to confirm or modify the diagnoses made. Appendix A of the DSM-III Case Book contains a condensed version of the decision tree for

the differential diagnosis of disorders with psychotic features (figure 1).

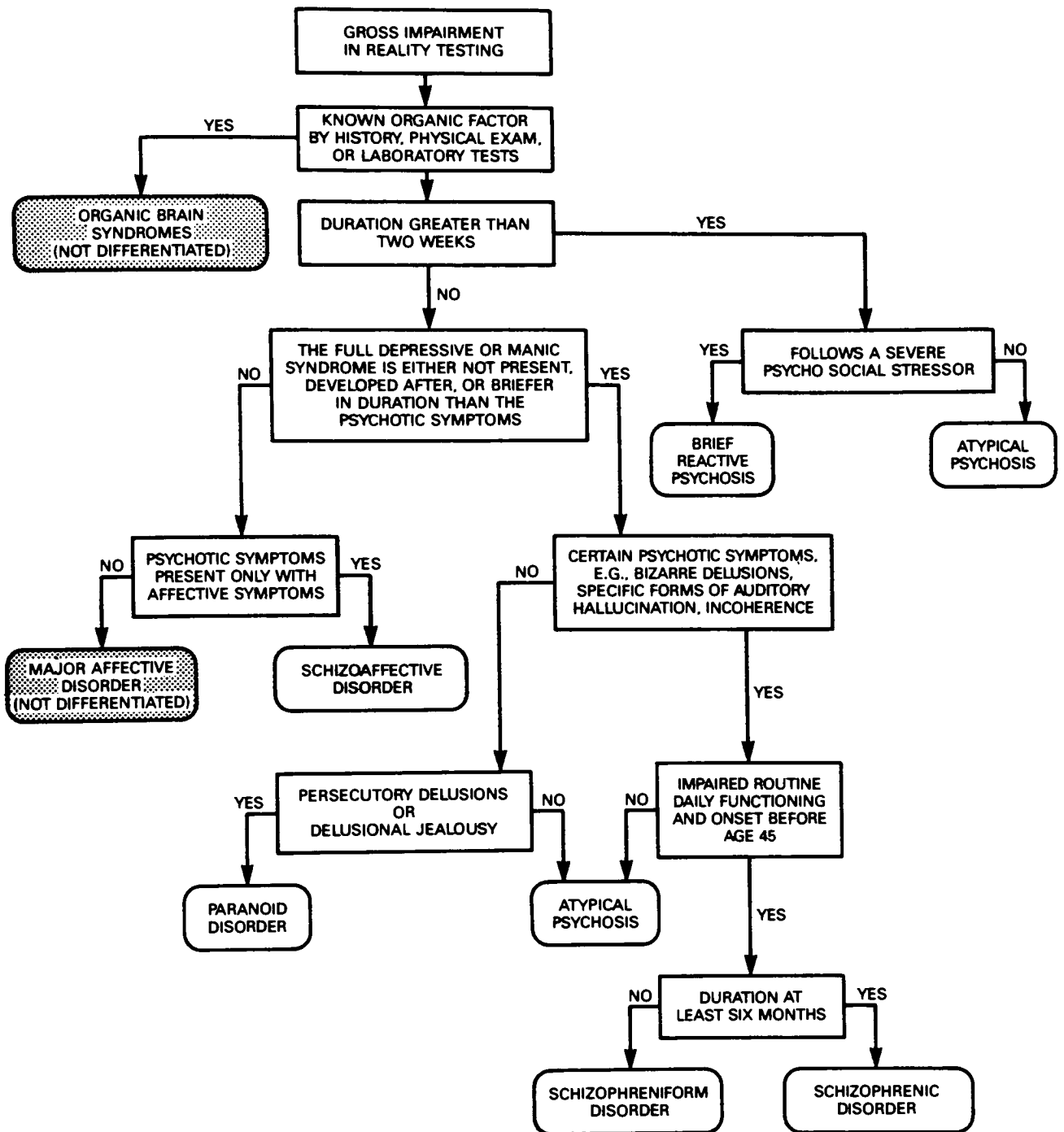


Figure 1. Differential Diagnosis of Psychotic Features.

Note: This diagram is a modified version of the decision tree appearing in Appendix A of the DSM-III [9].

#### 4. Prolog

Prolog is a symbol manipulation language designed to represent and manipulate complex concepts. Prolog was proposed by Alain Colmerauer, in about 1970 and uses logic-based methods to specify tasks. Prolog stands for "programming in logic". Logic is a way of representing arguments to test their validity [28].

Prolog is based upon the formal mathematical theory of Predicate Calculus used in classical logic to express propositions, the relations between propositions and rules to infer new propositions [29]. Theorem proving uses rules expressed in predicate calculus to infer new propositions or theorems from those already known (axioms). Resolution is an inference rule that uses a pattern matching strategy, known as unification, to prove theorems from axioms in a clausal format. In practice it has been necessary to restrict the clausal format to a particular type of clause, the Horn clause [28].

A logic-based program, such as Prolog, consists of a set of Horn clauses of the form:

`<consequent> :- <antecedent1> <antecedent2>...<antecedentn>`

Where the antecedents are predicates, of the form  $P(\text{term1 term2} \dots \text{termn})$ , that can be tested for their truth value, and the consequent is a predicate that is true if its antecedents can be proved true [3]. Not all clauses have antecedents, and a clause without an antecedent is a fact and is always true. A clause without a consequent is a goal, and

only one is allowed at a time. Figure 2 shows a simple Prolog program as described in Clocksin and Mellish [28].

The resolution principle is used to infer new propositions. It states that if the predicate and the number of arguments appear on both the left hand side of one clause and the right hand side of another clause then the clause obtained by fitting together the two clauses follows from them [28]. The process of matching the goal against the consequence is known as unification. Two symbolic constants in the same argument position match if they are equal. A variable may match anything, but once it is matched, or instantiated, within a rule, all occurrences of that variable take on the matched value. An anonymous variable or wild card can match anything and will not retain the matched value. The wild card allows matching of many different patterns.

In Prolog, flow of control does not proceed sequentially from line-to-line (or clause to clause) as in other programming languages. To execute a Prolog program a goal is supplied, and the interpreter, the mechanism that analyzes the code, applies the resolution rule to the stored clauses and matches the goal against the consequent. The interpreter works recursively, setting up antecedents as subgoals and backtracking in case of failure. If all of the subgoals are satisfied, the goal is proved and the substitutions found constitute the answer [1].

---

```
male(george).                ; george is a male
parent(mary,shannon).        ; mary is the parent of
                                ; shannon
parent(george,shannon).      ; george is the parent
                                ; of shannon
father(X,Y) :- parent(X,Y), male(X). ; X is the father of Y
                                ; if X is the parent of
                                ; Y and X is a male
```

**Figure 2. -- Simple Prolog Program**

---



For example, given the simple program (figure 2) and the goal:

```
:- male(X).
```

The goal is unified with the fact, male(george), X becomes instantiated to george and the answer, X = george, is returned. Given the same program and the goal, father(george,Z), the answer, Z = shannon is returned.

When failure occurs, Prolog provides an automatic backtracking mechanism which attempts to prove alternate solutions in the following way. Given

```
<goal1> :- <subgoal1> <subgoal2> <subgoal3>.
```

if the goal currently being proven can be unified with goal1 then subgoal1, subgoal2 and subgoal3 are proved in that order. If subgoal1 is proved true all variables within the rule become instantiated (take on values). If the same variables occur in subgoal2 they take on the matched value and an attempt is made to prove subgoal2. If subgoal2 fails the system backs up to subgoal1, all variables within the rule are uninstantiated and an attempt is made to prove subgoal1 in another way. If subgoal1 is proven again, all variables within the rule become reinstantiated and subgoal2 is attempted again. If subgoal2 fails repeatedly, all possible ways of proving subgoal1 are attempted. If subgoal2 is proven, an attempt is made to prove subgoal3. If subgoal3 is proven, the entire production rule goal1 is proven. If subgoal3 fails, the system backtracks, searching for a way to make subgoal3 succeed. If subgoal3 never succeeds, goal1 fails. The backtracking algorithm performs a

depth-first search as opposed to a breadth-first search, because Prolog attempts to satisfy each subgoal in all possible ways before trying an alternate subgoal.

For example given the simple program (figure 2) and the goal:

```
:- father(F,Z).
```

the goal is unified with the father rule making  $F = X$  and  $Z = Y$ . The first subgoal,  $\text{parent}(X,Y)$ , is unified with  $\text{parent}(\text{mary},\text{shannon})$ , making  $F = X = \text{mary}$  and  $Z = Y = \text{shannon}$ . The second subgoal,  $\text{male}(X)$  where  $X = \text{mary}$ , cannot be unified with any clause so the second subgoal fails. Backtracking occurs, the variables become uninstantiated and the first subgoal is retried. This time  $\text{parent}(X,Y)$  is unified with  $\text{parent}(\text{george},\text{shannon})$ , making  $F = X = \text{george}$  and  $Z = Y = \text{shannon}$ . The second subgoal,  $\text{male}(X)$  where  $X = \text{george}$ , is unified with the fact,  $\text{male}(\text{george})$ , and the whole goal succeeds returning  $F = \text{george}$  and  $Z = \text{shannon}$ .

This discussion has been based on the "standard" version of Prolog as described by Clocksin and Mellish [28]. The Psyxpert system is implemented in a nonstandard version, HC Prolog, developed at Virginia Polytechnic Institute and State University [30]. The syntactical differences between the two versions of Prolog are described in Appendix A of this paper.

Unification and instantiation give Prolog the ability to perform complex pattern-matching tasks. The search through the antecedents to prove the consequent looks much like

backward-chaining in a production system. Automatic resolution theorem proving and a built-in relational data base characterize Prolog as an expert system building language. In summary, the pattern-matching capability, control scheme, theorem proving ability and built-in relational data base make Prolog a good choice for the implementation of an expert system.

## 5. The Psyxpert System

### 5.1 Consultation Overview

The Psyxpert system is a consultation aide to be used by a psychiatric specialist in the differential diagnosis of mental disorders when psychotic features are the prominent part of the presenting clinical picture. Psyxpert can aide psychiatrists by consistently classifying patients based upon the diagnostic criteria and classification scheme from the Diagnostic and Statistical Manual of Mental Disorders (DSM-III) [26]. The decision tree for differential diagnosis of psychotic features in Appendix A. of the DSM-III Casebook [27] was used to provide a framework for the diagnostic process (figure 1).

The user interface prompts the user with questions and accepts the user's answers, which are asserted with part of the question as facts in the data base, where the inference engine can make use of them. The user interface also passes requests for explanations from the user to the explanation subsystem and gives the explanations produced by the explanation subsystem to the user. The user interface also prints the diagnostic report generated by the knowledge base and the inference engine.

Psyxpert uses a menu-driven user interface to question the user by displaying questions with a numbered list of possible answers (figure 3). The user simply types the number of the answer chosen, which eliminates spelling errors

---

Rank the patient's current mood along the following  
7-point depression-elation continuum:

- 1)severely\_depressed
- 2)moderately\_depressed
- 3)mildly\_depressed
- 4)euthymic
- 5)mildly\_elated
- 6)moderately\_elated
- 7)severely\_elated

Does the patient show evidence of auditory experiences  
in the absence of an external stimulus?

- 1)yes
- 2)no
- 3)unsure

**Figure 3. -- Sample questions**

---

and simplifies the detection of illegal answers. The user also may respond to any question with a keyword that causes the system to react in a particular way. The four keywords are '?', 'how', 'why' and 'stop'. The system was constructed so that more keywords and keyword functions could be added.

The user interface checks the response to see if it is a keyword. If the response is not a keyword, the user interface checks to make sure it corresponds to a menu selection. If it does not, 'Illegal Answer !!!' is printed, and the question with the menu of answers is redisplayed. If the response is a legal choice, it becomes an index into a list of answers used to print the menu, and the answer is asserted into the data base for future reference.

If the response is the keyword 'stop', the user interface finishes the consultation by asking the user either to end the consultation, rerun the consultation or return to the operating system (figure 4). Ending the consultation turns control over to the top-level of the HC Prolog interpreter. The entire knowledge base remains intact, and the user may enter Prolog commands to examine the contents of the knowledge base. Rerunning the consultation deletes the facts in the data base but leaves the rule base intact. Psyxpert is then invoked automatically, and a new consultation may be started. Returning to the operating system causes the consultation to be terminated, the HC Prolog interpreter to be exited and the operating system prompt to be displayed.

---

**The system asks--**

Do you want to :

1)end\_the\_consultation

2)rerun\_Psyxpert

3)exit\_the\_interpreter

**The user responds--**

3

**The system responds--**

Returning to the operating system.

**Figure 4. -- Finishing the Consultation**

---

If the response is an explanation keyword, such as '?', 'why', or 'how', the explanation subsystem is invoked. Psyxpert supports four types of explanations. The user may type the keyword '?', and the system will give an explanation of the current question (figure 5). The user may type the keyword 'how', and the system gives ways the user may establish an answer to the question (figure 6). Typing the keyword 'why' causes the system to give the information it is trying to determine or the goal it is trying to satisfy along with a list of known key facts. At the end of the consultation the system may print an explanation of the diagnosis to the screen. The explanation subsystem will be covered in detail in the next section.

The consultation is begun after the psychiatric specialist has taken the patient's history, performed a physical examination and conducted a mental status examination. At the beginning of the consultation the user may create a log file of the current consultation session. The user interface prompts the user for a file name which must conform to the operating system's naming convention. The VMS operating system requires a file descriptor suffix to be appended to the file name, and Psyxpert requires file names to be enclosed in quotation marks. For example, "run.out" is a legal file name. At the end of the consultation the file is automatically closed.

The user interface then asks if the user needs help. If the user responds affirmatively, the user interface prints the help files on the screen. There are three help files



---

**The system asks--**

Rate the patient's sensorium:

1)clear

2)impaired

3)unsure

**The user responds--**

?

**The system responds--**

The sensorium describes the condition of a subject relative to the subject's mental clarity and is evaluated in terms of orientation, memory and calculating ability.

**Figure 5. -- Explanation example**

---

---

**The system asks--**

Rate the patient's sensorium:

1)clear

2)impaired

3)unsure

**The user responds--**

how

**The system responds--**

HOW?

Orientation may be evaluated by asking the subject the month, date, year and place. Memory may be tested by asking the subject to recall recent events.

Calculating ability may be tested by asking the subject to perform serial seven subtractions (the subject must have had the skill previously).

**Figure 6. -- Explanation example**

---

that contain information about running the Psyxpert system. The user may see one, two or all three of the files (figure 7).

Psyxpert asks the user a set of initial questions concerning the patient's mood, affect, sensorium, intellectual function and thought processes. The answers to these questions are the facts upon which all subsequent inference procedures are based.

The system then begins the diagnostic phase. At any point the system may ask additional questions to gather information it cannot infer. The first step is to confirm or disconfirm the existence of organic brain disorder. In the case of organic brain disorder the system does not diagnose the specific type. In the absence of organic brain disorder, the system attempts to confirm or disconfirm the presence of psychosis. If psychosis is confirmed, the system attempts to match the known facts with one of the seven psychotic disorders.

When the diagnostic phase of the consultation is complete, the user interface prints a diagnostic report to the screen containing the patient's name, the patient's age, the diagnosis, the confidence factor and Psyxpert's recommendations for either therapy or further evaluation (figure 8). The user interface asks if the user wishes to see an explanation of the diagnosis, in which case the diagnostic explanation is printed to the screen. The system then ends the consultation by giving the user the option of rerunning the Psyxpert system, exiting the Psyxpert system

Would you like help to run the Psyxpert system?

- 1) yes
- 2) no

1

The following information is intended to help the user run the Psyxpert system. This system is designed to be used by a qualified psychiatrist as a consultation aide in the diagnosis of psychiatric disorders in accordance with the DSM-III classification scheme. The system is not to be used in place of a psychiatrist.

The system is to be used after the psychiatrist has taken a history, performed a physical examination, and given a mental status examination. The psychiatrist should answer the questions based upon these findings. The answers to most questions must be selected from a menu by entering the number corresponding to the answer that in the psychiatrist's judgement best fits the facts. The patient's name must be entered enclosed in parenthesis.

If you would like to see more type "yes".  
yes

During the consultation, the explanation facility provides the following types of explanations:

- 1) If the psychiatrist does not understand a question, entering "?" will provide a detailed explanation of the question and the question will be asked again.
- 2) If the psychiatrist wants to know why a question was asked, typing "why" will give the goal the system is trying to satisfy and the question will be asked again.
- 3) If the psychiatrist wants to know how the answer to a question may be determined, typing "how" will give ways to determine the answer and the question will be asked again.

If you would like to see more type "yes".  
yes

After a diagnosis is given, the explanation facility provides the following explanation:

- 1) The psychiatrist is asked if he/she would like an explanation of the diagnosis, if yes is chosen an explanation of the diagnosis with the facts is given.

The psychiatrist is then given the options of:

- 1) ending the consultation,
- 2) rerunning the consultation or
- 3) exiting from the interpreter and returning to the operating system command level.

At any time during the consultation the psychiatrist may abort the consultation by typing "stop" in response to a question.

Begin the consultation.

---

DIAGNOSTIC REPORT:

Patient : Jones

Age : 23

Diagnosis : SCHIZOPHRENIC\_DISORDER

Confidence factor : 2

Recommendation :

Treatment of the Schizophrenic disorder.

---

**Figure 8. — Diagnostic Report**

---

and returning to the Prolog interpreter, or exiting the Psyxpert system and returning to the operating system as described above.

## 5.2 System Organization

Psyxpert is made up of several files that contain production rules written in Prolog. These files are loaded into the Prolog interpreter when a consultation is started. Logically the files may be grouped into three general categories of functions: the knowledge base, the user interface and the explanation subsystem (figure 9). The inference engine is built-in as part of the Prolog interpreter, and it uses the rules and data in the knowledge base to make inferences that add new facts to the data base or produce a diagnosis to be passed to the user interface.

The knowledge base consists of several files that contain two types of knowledge, rules and data. The rule base is made up of files that consist of production rules containing psychiatric knowledge. The data base contains facts that are derived and asserted during the consultation. The rule base may be expanded with knowledge about other mental disorders to increase Psyxpert's diagnostic capabilities.

The user interface provides interactive communication between the user and the system and serves as the communication port between the system and the external world [9]. The user interface maintains the initiative by asking

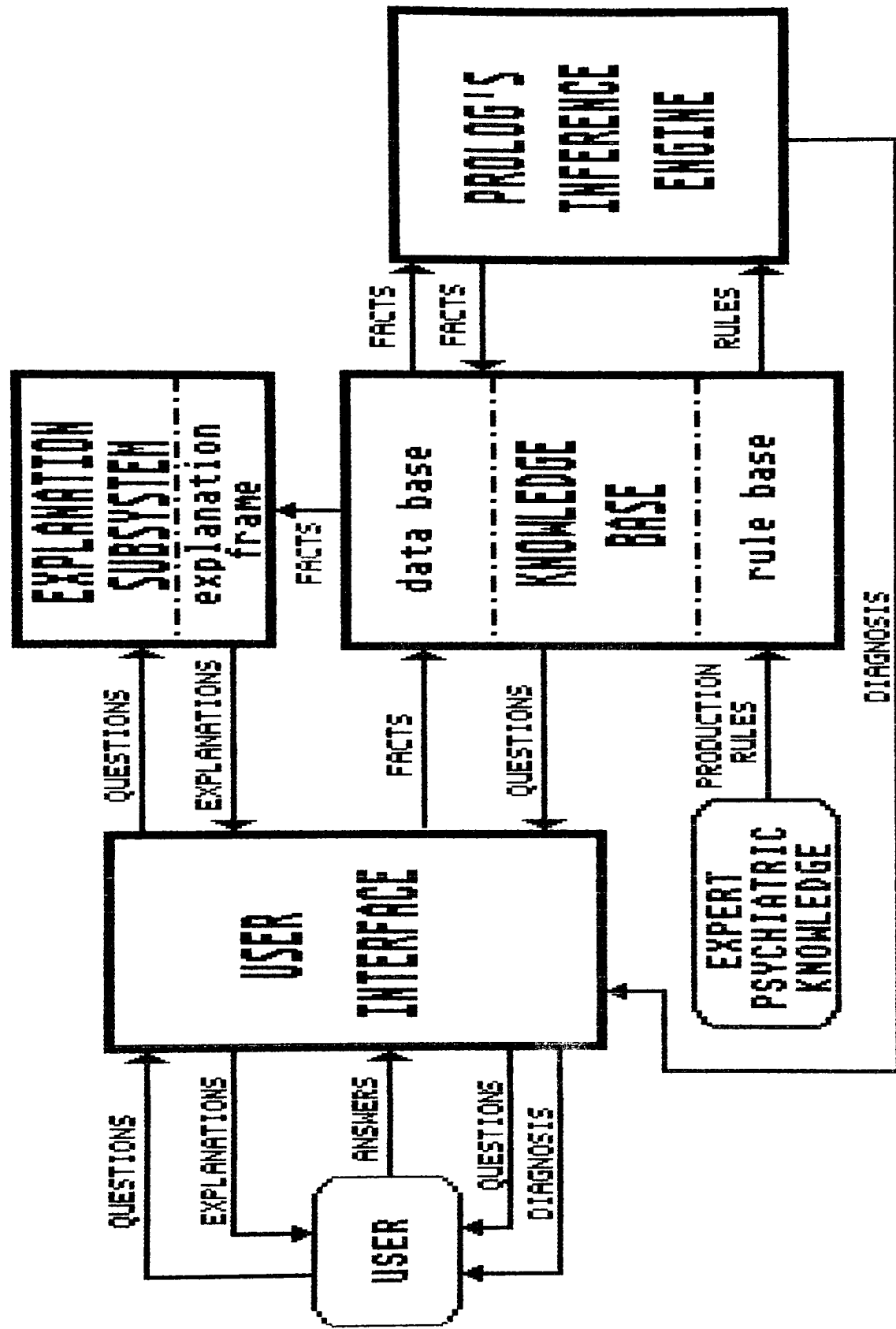


Figure 9. -- Psyxpert Organizational Diagram

the user questions so that the input may be anticipated. The system assumes the user is familiar with psychiatric jargon and medical terminology when addressing the user.

The explanation subsystem consists of files containing rules and a data structure known as the explanation frame. The rules interpret the explanation keywords and explain the system's questions. The explanation frame contains key facts from the knowledge base. The explanation subsystem can be invoked by the user in response to any question the system asks by typing an explanation keyword.

### 5.3 Implementation Details

#### 5.3.1 Knowledge Representation

The diagnostic criteria for mental disorders with a major psychotic feature used in the decision tree in Appendix A of the DSM-III Casebook (figure 1) are translated into production rules written in Prolog. The following discussion explains the representation and will be based upon the decision tree, the diagnose rule in file "diagnose.p" and the organic rule in file "organic.p" listed in Appendix D (D-5, D-17).

The control knowledge concerning when to look for what, implicit in the decision tree, is translated into seven diagnose rules written to control the order in which each disorder rule is tried. The first premise of each diagnose rule attempts to establish one of the categories of mental



illness in which there may be a major psychotic factor, organic disorder, psychotic disorder or mood disorder. After a category is established, an attempt is made to diagnose a particular disorder.

The first diagnose rule attempts to confirm or deny the possibility of an organic disorder. The second box in the decision tree contains the information needed to determine the presence or absence of an organic disorder and will be described in detail below. The `diagnose_organic` rule is included and could be expanded in the future to diagnose the specific type of organic disorder. At this time the diagnostic phase of the consultation is complete if the possibility of an organic disorder is confirmed.

The next four diagnose rules attempt to confirm or deny the possibility of a psychotic disorder. The first rule is satisfied if psychosis is proven with a certainty factor of 2, for total certainty, and the psychotic disorder is further diagnosed. The second is satisfied if psychosis is proven with a certainty factor of 1, for less than total certainty; and in this case an attempt is made to establish a mood disturbance that would account for the symptoms but that is not further diagnosed at this time. The third is satisfied if psychosis is proven with less than total certainty in the absence of mood disturbance and the psychotic disorder is further diagnosed. The fourth is satisfied if psychosis is proven with either certainty factor but the psychotic disorder cannot be further diagnosed.

The knowledge defining a major psychotic feature

contained in the first box of the decision tree is translated into fourteen psychotic rules in file "psychotic.p" (D-21). Each of these four rules attempts to confirm the presence or absence of a major psychotic feature by satisfying one of the fourteen psychotic rules. If the presence of a psychotic feature is confirmed, the `diagnose_psychosis` rule in file "psychotic.p" is used to confirm one of the seven possible mental disorders in the following order: Brief Reactive Psychosis, Major Affective Disorder with Psychotic Features, Schizoaffective Disorder, Paranoid Disorder, Schizophreniform Disorder, Schizophrenic Disorder and Atypical Psychosis. The diagnostic criteria for each of these disorders have been translated into one or more Prolog rules for each disorder and can be found in Appendix D.

If there is no major psychotic feature, the possibility of a mood disorder is confirmed or denied with the sixth `diagnose` rule. The `diagnose_mood` rule in file "mood.p" (D-5) is included so that the system could be expanded, in the future, to diagnose the particular type of mood disorder. At this time the diagnostic phase of the consultation is simply ended.

The final `diagnose` rule is satisfied if all the others fail and no diagnosis can be made. In this case the diagnosis, No Diagnosis, is returned and the category, Not Specified, is appended to the explanation frame.

The translation of the diagnostic criteria (in the second box of the decision tree) into the Prolog rule, `organic`, in file "organic.p" (D-17), will be discussed in the

following paragraphs. The knowledge contained in the second box may be translated into the following pseudo-code production rule:

```
If  history shows organic factor
      or
      physical examination shows organic factor
      or
      laboratory tests show organic factor
then
      organic brain disorder
```

The pseudo-code may be translated into a Prolog rule of the form:

```
(organic_brain_disorder)  if
      (or
        (history_organic yes)
        (physical_organic yes)
        (lab_test_organic yes)
      )
```

In the Prolog system, the three premises the rule uses to prove its conclusion are actually written this way :

```
(fact history_organic yes)
```

The word "fact" is listed in front of all rules which are asserted during a consultation to distinguish them from rules asserted before a consultation is started. This is necessary

to enable the user to run a new consultation without reloading the system itself. When a consultation is finished, and the user chooses to run another consultation, all rules preceded by fact are deleted with the built-in delete function, and the psychiatric diagnostic rules are left intact.

The variables \*in, \*out and the append premise are used to store information in the explanation frame, the details of which are not important at this point and will be described in the section on the explanation subsystem. Each question premise causes the user interface to ask the pertinent question, read the answer, then assert the answer into the data base. The user responds to most questions with yes, no or unsure. The implementation details are described in the section on the user interface.

### **Reasoning With Uncertain or Incomplete Information**

Each premise of the Prolog rule shown above is expanded into an or clause with a part for each possible response or fact. This question, as with most questions, may be answered by either yes, unsure or no. Each response is assigned a numerical value or certainty measure (CM), and a variable is instantiated to the certainty measure for each premise. The variable, \*m1, is instantiated to 4 if (fact history\_organic yes) can be proved, 0 if (fact organic\_history no) can be proved, and 1 if (organic\_history unsure) can be proved. The same process is repeated for the second and third premises.

The variable, \*m2, is set to the sum of \*m1 and the appropriate weight. The variable, \*m3, contains the total certainty measure for all three premises in the organic rule.

The total certainty measure, \*m3, is then compared to a predetermined threshold of 4. If \*m3 is equal to or greater than 4, the diagnosis is organic brain disorder with a confidence factor of 2, indicating total certainty. If \*m3 is less than 4 and greater than or equal to 1, the diagnosis is organic brain disorder with a confidence factor of 1, indicating less than total certainty. If \*m3 is 0, organic brain disorder is not confirmed, the organic rule fails, control returns to the diagnose rule and the next rule is tried.

Each one of the three premises in the organic rule is considered to be as important as the next. If one premise of a rule is more important than another, an importance measure may be attached to the premise and is used to give the certainty measure of that premise more weight when figured into the calculation of the total certainty factor (figure 10). A certainty factor for each premise of a rule (CFsub) is arrived at by multiplying the certainty measure for the fact or response by the importance measure ( $CF_{sub} = CM * IM$ ). In the schizophrenic rule, (figure 10), the premise, ( $:= *cf1 (* *cm1 2)$ ), instantiates \*cf1 (CFsub), the certainty factor for that premise, to the product of \*cm1, the certainty measure for that response, and 2, the importance measure for that premise. The total certainty measure for all the

---

```

((schizophrenia *in *out) if
  (duration psychosis *time *b)
  (>= *time 2)
  (mood_disturbance *c *d secondary)
  (onset psychosis *weeks_ago1 *e)
  (onset mood *weeks_ago2 *e)
  (> *weeks_ago1 *weeks_ago2)
  (psychological_disorder *c *list *cm1)
  (:= *cf1 (* *cm1 2))
  (question 76 *answ *f)
  (or((fact functioning_impaired yes)
      (:= *cf2 (+ *cf1 (* 2 2))) )
      ((fact functioning_impaired unsure)
      (:= *cf2 (+ *cf1 (* 1 2))) ) )
  (onset_prior *h)
  (:= *cf3 (+ *cf2 (* 2 2)))
  (>= *time 26)
  (:= *cf4 (+ *cf3 (* 2 2)))
  (or((>= *cf4 14)
      (append *k ((CF2 2)) *out))
      ((= *cf4 12)
      (append *k ((CF2 1)) *out)) ) )

```

**Figure 10. -- A Rule from the Psyxpert System**

Note : Premises which store information in the explanation frame have been removed for clarity. The entire rule is listed in Appendix D.

---

premises of the rule are added together in the same way as before and are compared to a predetermined threshold to arrive at a diagnosis with a confidence factor. Importance measures, as well as thresholds, may be changed during the testing phase to fine tune the system and produce the expected conclusions. Thorough testing and adjusting of importance measures and thresholds has not been completed for the Psyxpert system.

### 5.3.2 The User Interface

All the possible questions the system may ask are numbered and stored in the file "questfile.p" (D-23). Each question is stored as a fact with the predicate quest and four variable pieces of information: a unique question number; the question string to be printed on the screen; a list of possible answers or menu choices; and a list containing the fact to be asserted.

When the system needs a piece of information that it cannot infer, the user interface invokes the question rule in file "userinter.p" (D-30). There are two different question rules, but they both have the same three variables: an input variable instantiated to the number of the question being asked; an output variable that becomes instantiated to the answer; and an input variable instantiated to a list containing the information known about the patient so far. This variable is referred to as the explanation frame, because the information is used for explanation purposes and

will be discussed later. The first question rule is always tried first but fails when the information is not already known and will be described later. The second question rule is described here.

When the current goal is unified with the question rule, its premises, in turn, become subgoals; and each of its variables, \*qnum, \*answ and \*struct, are instantiated to the variables in the current question goal. If, for example, the current goal is (question 33 \*answ \*struct), then the second question rule is tried, and the variable \*qnum becomes instantiated to 33. The first premise in the question rule, (quest 33 \*line \*list \*assertion), is matched with the corresponding quest fact described above and \*line is instantiated to the string "Is the patients form of thought logical?", \*list is instantiated to the list (yes no unsure), and \*assertion is instantiated to the list (thought\_logical).

The second premise, (putline \*line), is a built-in function that prints the string that \*line is instantiated to on the screen (figure 3). The third premise, (print\_menu \*list 1) matches the print\_menu rule in the same file. The variable \*list is instantiated to the list of menu choices, and 1 is the number to start listing the choices with. The print\_menu rule uses the built-in dot operator and putline functions to break up the list of possible menu choices and print them beside a number (figure 3).

The fourth premise, (:= \*in (read)), uses the built-in assignment operator, :=, and the built-in read function to



wait for input, read the input and instantiate \*in to whatever was read. If the user typed 1, for yes, \*in is instantiated to 1. The fifth premise is an or clause. The first part, (== \*in stop)(stop\_it), calls the stop\_it rule in file "end.p" (D-7), if \*in is instantiated to stop. Otherwise the second part, (== \*in \*in), always succeeds.

The sixth premise, (explain\_answ \*qnum \*in \*new \*struct), matches the first explain\_answ rule in the same file and is used to check for explanation keywords. The variable \*qnum is instantiated to 33; \*in is instantiated to 1; \*new is uninstantiated and will be instantiated to the new answer; and \*struct is used as before. (The contents of \*struct are not important at this point.) The first premise in the first explain\_answ rule, (or(== \*in ?)(== \*in why)(== \*in how)), is an or clause that succeeds if \*in is ?, why or how. The second premise, (explain \*qnum \*in \*struct), invokes the explanation subsystem and will be explained in detail in the following section. In this case the first rule fails, the second explain\_answ rule matches and \*new becomes instantiated to \*in.

The seventh premise, (interp\_answ \*assertion \*new \*answer \*list), matches the interp\_answ rule in the same file and has four variables: \*assertion is the list to be asserted, in this case (thought\_logical); \*new is the new answer, in this case 1; \*answer will be instantiated to the answer that corresponds to the menu selection; and \*list is instantiated to the list of possible answers, in this case (yes no unsure).

There are four `interp_answ` rules. The first matches when `*list` is instantiated to `nil` because the question did not use the menu format and `*answer` is instantiated to `*new`. In this case the first rule fails. The second `interp_answ` rule matches in this case. The variable `*new` is instantiated to `*a1`, `*answer` is instantiated to `*a2` and `*list` is instantiated to `*l`. The first premise, `(numberp *a1)`, is a built-in function that succeeds if `*a1` is a number and fails otherwise. In this case it succeeds. The second premise, `(or((= *a1 0)(:= *a2 nil))(:= *a2 (nth '*l *a1)))`, is an or clause. If `*a1` is instantiated to the illegal answer 0, `*a2` is instantiated to `nil`. Otherwise the built-in assignment operator, `:=`, and the built-in function `nth` are used to instantiate `*a2` to the `nth` item in list `*l`, in this case `*a2` is instantiated to `yes`. When `*a1` is not instantiated to a number, the second `interp_answ` rule fails and the third rule is tried. The third rule is used if `*a1` is an answer generated by the explanation subsystem; and if it fails, the fourth rule instantiates `*a2` to `nil` to indicate an input error.

The eighth premise, `(or((= *answer nil)(putline "Illegal answer !!!) (question *qnum *z *struct) (= *answ *z)) (= *answ *answer))`, is an or clause. If `*answer` is instantiated to `nil`, the error message is printed, the question is asked again and `*answ` is instantiated to `*z`, the new answer. Otherwise, `*answ` is instantiated to `*answer`, in this case `*answ` is instantiated to `yes`.

The ninth premise, `(assert_answ *answ *assertion)`,

matches the `assert_answ` rules in the same file. There are two `assert_answ` rules. The first rule matches when `*assertion` is `nil`, and there is nothing to be asserted. The second rule uses the built-in dot operator to break the list in `*assertion` into `*head` and `*tail`. In this case, `*head` is instantiated to `thought_logical` and `*tail` is instantiated to `nil`. The premise, `(or(fact *head *answ) (assert ((fact *head *any) if (== *any *answ))))`, is an or clause. The first part checks to see if `(fact *head *answ)` can be proven. In this case `(fact thought_logical yes)` cannot be proven. Otherwise a rule is asserted which if satisfied proves `(fact *head *answ)`. In this case the rule `((fact thought_logical yes) if (== yes yes))` is asserted into the data base, and any attempt to prove `(fact thought_logical yes)` will succeed from now on. The predicate `fact` is used to distinguish the rules generated by `assert_answ` from those loaded into the system initially. The tenth premise, `(cut)`, is the built-in cut operator used to prevent backtracking to resatisfy the rule in case of subsequent failure.

If the user interface detects the keyword 'stop', the premise, `(stop_it)`, is matched with the `stop_it` rule in file "end.p" (D-7). The first premise asks question 99 (figure 4). The second premise, `(log_off)`, matches the `log_off` rule in the same file. If a log file has been created, it is disconnected and closed.

The third premise of the `stop_it` rule is an or clause and one of the three cases will succeed. If `*answ` is instantiated to `end_the_consultation`, the built-in stop

function is invoked. The message following stop, enclosed in quotation marks, is printed to the screen; the HC Prolog prompt is returned, and the data base is left intact. If \*answ is instantiated to rerun\_Psyxpert, a message is printed; the facts in the data base are deleted, although the rules are left intact; and another consultation is started. If \*answ is instantiated to exit\_the\_interpreter, the built-in quit function is invoked, a message is printed and the operating system prompt is displayed.

#### 5.3.4 The Explanation Subsystem

When the explanation subsystem is invoked, it interprets the explanation keyword and reacts accordingly. In the case of '?' and 'how' questions, the file that corresponds to that particular question is printed on the screen by the user interface. The explanation subsystem needs more information to answer 'why'. For this reason key facts are stored throughout the consultation in the explanation frame. When responding to a 'why' question, the explanation subsystem has the user interface print the relevant information contained in the explanation frame (figure 11). After the explanation subsystem responds to the explanation keyword, control is returned to the user interface so the consultation may continue, and the original question is repeated.

The explanation subsystem is invoked when the explain\_answ rule in file "userinter.p" (D-30) detects an explanation keyword in the variable \*in. The premise,

---

**The system asks--**

Would you characterize the patient's delusions as  
bizarre?

1)yes

2)no

3)unsure

**The user responds--**

why

**The system responds--**

WHY?

Because the possibility of PARANOID\_DISORDER  
is under consideration.

Trying to determine PSYCHOLOGICAL\_DISTURBANCE

The facts are :

DELUSIONS

DURATION\_PSYCHOSIS\_GREATER\_THAN\_2\_WEEKS

MOOD\_DISTURBANCE

MODERATELY\_ELATED

ONSET\_PSYCHOSIS\_PRIOR\_TO\_ONSET\_MOOD\_DISTURBANCE

-----

**Figure 11. -- Explanation example**

---

(explain \*qnum \*in \*struct), matches the explain rule in file "explain.p" (D-11). The variable \*qnum is the question number; the variable \*answ is the explanation keyword; and \*list is the explanation frame containing the information known about the patient so far.

If \*answ is instantiated to "?", the explain\_? rule is invoked with \*qnum and an explanation of the question is printed. The first premise, (:= \*newnum (\* \*qnum 1000)), instantiates \*newnum to the result of \*qnum multiplied by 1000. The second premise, (or ((quest \*newnum \*line \*list \*assertion) (putline \*line)) (putline "Not implemented for this question at this time."))), is an or clause with two parts. The first part uses \*newnum to find the explanation corresponding to the question and print it to the screen. If this part fails because no explanation exists, the second part prints a message stating this to the screen.

If \*answ is instantiated to "how", the explain\_how rule is invoked with \*qnum, and how an answer to the question may be determined is printed to the screen. The first premise, (how\_file \*qnum \*statement), matches a fact by \*qnum, and \*statement is instantiated to the string containing the pertinent information. The second premise, (putline "HOW?"), prints HOW? to the screen, and the third premise prints the explanation. If a how\_file is not included for the question, the first explain\_how rule fails, and the second rule prints "Not implemented for this question" to the screen.

If \*answ is instantiated to "why", the explain\_why rule is invoked with \*qnum and \*list. The explain\_why rule gives

the reason the question is being asked, and the answer depends upon the type of question. For some questions no explanation is needed as the information is routine, for instance, the patient's name. Some questions are part of the initial information gathering phase and are asked of every patient. Other questions are not asked routinely, because the answer is needed to rule in or out a diagnostic category and is crucial to the diagnosis. For this reason, key facts are stored throughout the consultation by labeled fields in the explanation frame (figure 12).

The explanation frame consists of a series of lists within a list. Each list is headed by a labeled field and followed by a particular fact, for instance, (NAME Jones). As facts or other information about the patient become known, they are made into a list headed by the labeled field and appended to the explanation frame. The labeled field and fact combination allows the storage and retrieval of specific information from the explanation frame.

For example, if the system asks the question "Would you characterize the patient's delusions as bizarre?" and the user responds with 'why', an explanation is printed (figure 11). In this case the explain\_why rule is invoked with \*qnum instantiated to 77 and \*list instantiated to the explanation frame (figure 12). The first premise, (find\_list \*list \*out1 DISORDER \*rest1), is matched with the find\_list rule, \*out1 is instantiated to (DISORDER PARANOID\_DISORDER) and \*rest1 is instantiated to the list beginning with the DETERMINE field.

---

```
(  
  (NAME Jones)  
  (AGE 23)  
  (CATEGORY PSYCHOTIC_DISORDER)  
  (CF 2)  
  (FACT DELUSIONS)  
  (DISORDER PARANOID_DISORDER)  
  (DETERMINE PSYCHOLOGICAL_DISTURBANCE)  
  (FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS)  
  (FACT MOOD_DISTURBANCE)  
  (FACT MODERATELY_ELATED)  
  (FACT ONSET_PSYCHOSIS_PRIOR_TO ONSET_MOOD_  
    DISTURBANCE)  
)
```

**Figure 12. -- Explanation Frame**

---



The fourth premise in the `explain_why` rule prints "WHY?" then uses the `println` rule to print "Because the possibility of PARANOID\_ DISORDER is under consideration." The fifth premise looks for the field `DETERMINE` with the `find_list` rule, and prints "Trying to determine PSYCHOLOGICAL\_DISTURBANCE". The premise `(print_facts *list)` prints the facts in the explanation frame. Control returns to the `explain_answ` rule in the file "userinter.p" (D-30), the original question is repeated and the consultation continues.

The fourth type of explanation is the diagnostic explanation. The diagnostic explanation is given, if the user wants one, at the end of the consultation (figure 13). The premise, `(explain_diagnosis *struct)`, in the end rule in file "end.p" (D-7), matches the `explain_diagnosis` rule in file "explain.p" (D-11). The premise, `(print_diagnosis *struct)`, matches the `print_diagnosis` rule in file "end.p" and prints an explanation of the diagnosis.

#### 5.4 An Example Consultation

An example consultation is presented and discussed to illustrate the features and implementation details of the Psyxpert system. A copy of test case #51, "Miriam and Esther", taken from the DSM-III Casebook [27], the log run produced by Psyxpert and the actual Prolog code listing of all the rules listed alphabetically by file name are listed in Appendix B, C and D respectively. The following example

---

DIAGNOSTIC EXPLANATION

Diagnosis : SCHIZOPHRENIC\_DISORDER

Confidence factor : 2

Based upon the following facts :

DELUSIONS

DURATION\_PSYCHOSIS\_GREATER\_THAN\_2\_WEEKS

MOOD\_DISTURBANCE

MODERATELY\_ELATED

ONSET\_PSYCHOSIS\_PRIOR\_TO\_ONSET\_MOOD\_DISTURBANCE

PSYCHOLOGICAL\_DISTURBANCE

BIZARRE\_DELUSIONS

FUNCTIONING\_IMPAIRED

ONSET\_PRIOR\_TO\_AGE\_45

DURATION\_GREATER\_THAN\_6\_MONTHS

---

Figure 13. -- Diagnostic explanation example

---

will refer to these materials.

When beginning a consultation, the user logs on to the operating system, types 'hc' to invoke the HC Prolog interpreter and waits for the prompt, ':-', to be displayed on the screen. The user types '(load "psyxpert.p")' and the built-in load function causes the interpreter to load the file "psyxpert.p" (not listed) into Prolog's data base. Each file begins with the built-in (assert) function, which causes the Prolog rules to be asserted into the data base when the file is loaded. The user then types the goal (start), which is matched with the start rule in file "psyxpert.p", and the interpreter attempts to prove the goal by satisfying its premises.

The first premise, (load "....."), is a second call to the built-in load function with a list of all the files that make up the Psyxpert system. The second premise, (psyxpert), is matched with the psyxpert rule in file "control.p" (D-4). The psyxpert rule has three main premises that represent the three main stages of the consultation, begin, diagnose and end.

The first premise of the psyxpert rule, (begin \*chart), has one output variable, \*chart, that is uninstantiated at first but will eventually be instantiated to a list of pertinent patient information referred to as the explanation frame. The explanation frame was discussed in the previous section on the explanation subsystem. The begin rule causes the user interface to ask the initial set of questions and assert the answers into the data base, the details of which

were explained in the previous section on the user interface.

The second premise of the psyxpert rule, (diagnose \*chart \*final\_chart), has two variables: \*chart is the input variable, instantiated to the explanation frame generated by the begin rule; and \*final\_chart is an output variable that will be instantiated to the explanation frame generated by the diagnose rule. The diagnose rule causes the system to use the psychiatric knowledge base of Prolog rules to make a diagnostic inference.

The third premise of the psyxpert rule, (end \*final\_chart), has one input variable, \*final\_chart, which contains the explanatin frame generated by the diagnose rule. The end rule causes the user interface to print the diagnostic report and end the consultation.

The first premise of the begin rule in file "begin.p" (D-1) uses the built-in putline function to print a message on the screen. The second premise, (question 97 \*a \*struct), causes the user interface to ask if the user wants to create a log file of the current consultation and returns the answer in \*a.

The third premise, (log\_session \*a), matches one of the two log\_session rules in file "begin.p" (D-1). The first rule matches when \*a is instantiated to no and simply returns. The second rule matches when \*a is instantiated to yes. The first premise of the log\_session rule, (question 98 \*a \*s), asks the user for a file name and returns the answer in \*a. The second premise, (open \*a fd1 w), is a built-in

function, which opens a file named \*a for writing and equates it with the file descriptor fd1. The third premise, (connect stdout fd1), is a built-in function which, connects fd1 to stdout, the standard output. In this example, the user answered no, because a log file was created before the consultation was begun to illustrate the mechanics of starting a consultation.

The fourth premise of the begin rule, (question 95 \*answ \*struct), asks the user if he/she needs help to run the system and returns the answer in \*answ. The fifth premise is an or clause, and if \*answ is instantiated to yes, help\_file1 is printed to the screen, and the user is asked if he/she wants to see more. If the answer is again yes, help\_file2 is printed to the screen, and the user is again asked if he/she wants to see more. If the answer is again yes, help\_file3 is printed to the screen. In this example all three help files have been requested. If the user answers no, the help\_file printing is skipped.

The sixth premise uses the built-in putline function to print the message "Begin the consultation". The seventh premise, (initial\_info \*struct), matches the initial\_info rule in file "begin.p" (D-1). The initial\_info rule asks the patient's name and age and stores the information in the explanation frame, \*struct.

The eighth and last premise of the begin rule, (enter\_amsit \*struct), matches the enter\_amsit rule in file "userinterface.p" (D-30). The first premise of the enter\_amsit rule simply prints a message. The second and

third premises, (call\_quest 22 22 \*struct) and (call\_quest 26 51 \*struct), match the call\_quest rule in the file "userinterface.p". The call\_quest rule has three variables: \*start is the first question number; \*end is the last question number; and \*struct contains the explanation frame. Each question is asked with the question rule, the question number is increased and call\_quest is called recursively until the last question has been asked. This causes the user interface to ask the initial set of questions about the information gathered during the mental status examination.

The interpreter returns to the second premise of the psyxpert rule in file "control.p" (D-4), and the message, "Begin the diagnostic phase.", is printed. The third premise, (diagnose \*chart \*final\_chart), matches the diagnose rule in file "diagnose.p" (D-5). There are seven diagnose rules with two variables each, \*in\_struct and \*out\_struct, which are unified with \*chart and \*final\_chart respectively.

The first premise of the first diagnose rule, (organic \*in\_struct \*a\_struct), matches the organic rule in file "organic.p" (D-17). The first premise of the organic rule, (append \*in ((CATEGORY ORGANIC\_BRAIN\_DISORDER)) \*a), appends the information to the explanation frame. The second premise, (question 3 \*answ1 \*a), causes the user interface to ask question number 3. The fourth premise is an or clause with three parts, one for each possible response to the question. Each part appends the appropriate fact and certainty factor to the explanation frame then checks the data base to try to prove the fact. If the fact can be

proved, the variable, \*m1, is instantiated to a numerical weight, 4 for a positive response, 0 for a negative response and 1 for an uncertain response. In this case the user answers no so the fact, no\_organic\_history, and a certainty factor of 0 are appended to \*a\_struct, the explanation frame.

The fourth premise of the organic rule asks question number 6, and the process described above is repeated. The variable, \*m2, is instantiated to the sum of \*m1 and the numerical weight for the response to question number 6. The sixth premise asks question number 7, and the process is again repeated. In this case the user answers with the explanation keyword, 'why', and the explanation subsystem is invoked, an explanation is given and the question is repeated. The variable, \*m3, is instantiated to the sum of \*m2 and the numerical weight for the response to question number 7.

The variable, \*m3, now contains the total certainty weight. The eighth premise of the organic rule is an or clause with two parts. The first part compares \*m3 to a preset threshold of 4. If \*m3 is greater than or equal to 4, at least one response must have been positive, and the variable, \*cf, is instantiated to 2 for total certainty. In the second part, if \*m3 is equal to or greater than 1 but less than 4, at least one response must have been unsure and the variable, \*cf, is instantiated to 1 for less than total certainty. The ninth premise of the organic rule appends the certainty factor, \*cf, to the explanation frame, \*a\_struct,

and the interpreter returns to the diagnose rule.

The last premise in the diagnose rule, (diagnose\_organic \*a\_struct \*out\_struct), matches the diagnose\_organic rule in the file "organic.p" (D-17) and simply appends DIAGNOSIS NOT\_DIFFERENTIATED to the explanation frame and returns. In this example all responses are no and \*m3 is instantiated to 0, causing the organic rule to fail. The variable \*a\_struct, the explanation frame is again uninstantiated. The interpreter now attempts to satisfy the next diagnose rule.

The first premise in the second diagnose rule, (psychotic \*in\_struct \*a\_struct), matches the psychotic rule in file "psychotic.p" (D-21). There are fourteen psychotic rules that represent all the conditions that define a psychosis. The first and second premises append information to the explanation frame. The third premise, (delusions \*any 2), is matched with the delusions rule in file "rules.p" (D-25). The variable \*any is instantiated to persecutory, and 2 is matched with 2. The first premise of the delusions rule, (fact persecutory\_ delusions yes) is proved, and the delusions rule succeeds. The fourth premise in the psychotic rule appends CF 2 to the explanation frame, and the interpreter returns to the diagnose rule.

The second premise in the second diagnose rule, (find\_list \*a\_struct \*out CF \*rest), and the list containing the CF field of the explanation frame is returned in the variable \*out. The third premise, (!= \*out nil), succeeds if the list is not empty. The fourth premise, (return\_fact 1 \*out \*cf) returns \*cf, the confidence factor for the



psychotic conclusion. The fifth premise, (`== *cf 2`), compares the confidence factor to 2 and in this example succeeds. If the confidence factor had not been 2, the rule would have failed and the next diagnose rule would have been tried.

The sixth premise of the second diagnose rule, (`diagnose_psychosis *a_struct *out_struct`), matches the `diagnose_psychosis` rule in file "psychotic.p" (D-21). There are eight `diagnose_psychosis` rules, one for each of the seven disorders that Psyxpert can diagnose and one for failure. Each rule has two premises; the first attempts to prove the disorder, and the second appends information to the explanation frame.

The first premise of the first `diagnose_psychosis` rule, (`brief_reactive_psychosis *in *a`), matches the `brief_reactive_psychosis` rule in file "reactive.p" (D-24). The first two premises append information to the explanation frame. The third premise, (`duration psychosis *time *b`), matches the `duration` rule in file "rules.p" (D-25).

The first premise of the `duration` rule, (`onset psychosis *weeks_ago *in`), matches the `onset` rule in the same file. The `onset` rule causes the user interface to ask "How many weeks ago did the psychotic symptoms first appear?". In the example, the user's answer, 208, is returned in the variable `*weeks_ago`.

The second premise of the `duration` rule causes the user interface to ask if the psychotic symptoms are present, not present or unsure. The third premise is an `or` clause with

three parts, one for each possible response. In the example the user's response is present, so the variable `*time` is instantiated to `*weeks_ago` and the interpreter returns to the `brief_reactive_psychosis` rule.

The fourth premise, `(< *time 2)`, uses the built-in `less than` function to compare the duration of the psychosis in `*time` to 2. In the example, `*time` is 208, which is not less than 2; so the rule fails, and the interpreter attempts to satisfy the next `diagnose_psychosis` rule.

The first premise of the second `diagnose_psychosis` rule, `(major_affective *in *a)`, matches the `major_affective` rule in the file `"major_affect.p"` (D-15). The first and second premises append information to the explanation frame; the third and fourth premises are as described above, and the fifth premise appends information to the explanation frame. The sixth premise, `(mood_disturbance *c *d secondary)`, matches the second `mood_disturbance` rule in file `"mood.p"` (D-16).

The first premise appends information to the explanation frame; the second premise, `(mood_disorder) *list *cm)`, matches the `mood_disorder` rule in the same file. The first premise is an `or` clause with four parts that succeeds if the patient's mood is described as severely depressed, moderately depressed, moderately elated or severely elated. In the example the patient's mood is described as mildly elated; so the `mood_disorder` rule fails, the `mood_disturbance` rule fails, and the `major_affective` rule fails. The second `major_affective` rule is attempted and fails for the same

reason.

The interpreter attempts to satisfy the third `diagnose_psychosis` rule. The first premise, `(schizo_affective *in *a)`, matches the `schizo_affective` rule in the file `"sch_affective.p"` (D-29). The first seven premises are as described above. In this example, the rule fails and an attempt is made to satisfy the other `schizo_affective` rules, which fail in the same way as the previous rule, because there is no mood disorder.

The interpreter attempts to satisfy the fourth `diagnose_psychosis` rule. The first premise, `(paranoid *in *a)`, matches the `paranoid` rule in file `"paranoid.p"` (D-19). The first five premises are as described above. The sixth premise, `(!(mood_disturbance *c *d secondary))`, succeeds if the `mood_disturbance` rule described above fails, as in the example.

The seventh premise, `(psychological_disorder *c *list *cm1)`, is matched with the `psychological_disorder` rule in file `"schizo.p"` (D-27). The first premise appends information to the explanation frame. The second premise, `(psychological_disturbance *a *out *cm)`, is matched with the `psychological_disturbance` rule in the same file. There are nine `psychological_disturbance` rules, each of which characterizes a disturbance characteristic of schizophrenic disorder.

The first premise of the first `psychological_disturbance` rule, `(delusions *any 2)`, is satisfied as described above. The second premise causes the user interface to ask if the

patient's delusions can be characterized as bizarre. The third premise, (fact bizarre\_delusions yes), attempts to prove the fact, and in the example, the premise succeeds. The variable \*out is instantiated to the list (FACT BIZARRE\_DELUSIONS 2).

The eighth premise of the paranoid rule, (== \*list nil), succeeds if the last psychological\_disturbance rule succeeded and instantiated \*out and subsequently \*list to nil. In this example, the eighth premise fails, causing the paranoid rule to fail. An attempt is made to satisfy the other paranoid rules, but they fail in the same way.

The interpreter attempts to satisfy the fifth diagnose\_psychosis rule. The first premise, (schizophreniform \*in \*a), matches the schizophreniform rule in file "phreniform.p" (D-20). The first seven premises of the schizophreniform rule are as described above. The eighth premise, (!= \*list nil), succeeds if \*list is not equal to nil as in this example. The ninth and tenth premises append information to the explanation frame.

The eleventh premise, (:= \*cf1 ( \* \*cm1 2), instantiates \*cf1, to the product of \*cm1 and 2. The variable \*cm1 is the certainty measure returned by the psychological\_disturbance rule; 2 is the importance measure assigned to the premise, and \*cf1 is the confidence factor for this premise. The details were described in the section on reasoning with uncertainty. In this example \*cf1 is instantiated to 4.

The twelfth premise causes the user interface to ask if there has been a deterioration from a previous level of

functioning in work, social relations and self-care. The thirteenth premise is an or clause with two parts. The first part succeeds if the response to the question is yes; and the variable \*cf2 is instantiated to the sum of \*cf1 and the product of the certainty measure for the yes response times the importance measure for this premise (2 times 2). The second part succeeds if the response to the question is unsure; and the variable \*cf2 is instantiated to the sum of \*cf1 and the product of the certainty measure for the unsure response times the importance measure for this premise (1 times 2). If the response to the question is no, the or clause fails, causing the entire schizophreniform rule to fail. In the example, the response to the question is yes, so \*cf2 is instantiated to 8.

The fourteenth premise appends information to the explanation frame. The fifteenth premise, (onset\_prior \*h), matches the onset\_prior rule in file "schizo.p" (D-27). The first premise of the onset\_prior rule, (fact age \*age), returns the patient's age in the variable \*age. The second premise, (<= \*age 45), uses the built-in less than or equal to function to compare the patient's age to 45. In the example, the \*age is instantiated to 30, which is less than or equal to 45, and the rule succeeds.

The sixteenth premise appends informatin to the explanation frame. The seventeenth premise, (:= \*cf3 (+ \*cf2 (\* 2 2))), instantiates \*cf3 to the sum of \*cf2 and the product of 2, the certainty measure for the age response, and 2, the importance measure for the premise. In the example,

\*cf3 is instantiated to 12.

The eighteenth premise appends information to the explanation frame. The nineteenth premise, ( $< *time\ 26$ ), uses the built-in less than function to compare the duration of the psychotic symptoms in \*time to 26 weeks or six months. The premise succeeds if the duration is less than six months. In this example, \*time is instantiated to 208 weeks so the premise fails, causing the schizophreniform rule to fail. An attempt is made to satisfy the other schizophreniform rules but they fail for the same reason.

The interpreter attempts to satisfy the sixth diagnose\_psychosis rule. The first premise, (schizophrenia  $*in\ *a$ ), matches the schizophrenia rule in file "schizo.p" (D-27). The first eighteen premises are as described above. The nineteenth premise, ( $\geq *time\ 26$ ), uses the built-in greater than or equal to function to compare \*time to 26. In the example \*time is instantiated to 208, so the premise succeeds. The twentieth premise, appends information to the explanation frame.

The twenty-first premise, ( $:= *cf4\ (+\ *cf3\ (*\ 2\ 2))$ ), instantiates \*cf4 to the sum of \*cf3 and the product of 2, the certainty measure for the duration response, and 2, the importance measure for the premise. In this example, \*cf4 is instantiated to 16.

The twenty-second premise is an or clause with two parts. The first part compares \*cf4 to a preset threshold of 14. If \*cf4 is greater than or equal to 14, the list (CF2 2) is appended to the explanation frame. In the second part, if

\*cf4 is equal to 12, the list (CF2 1) is appended to the explanation frame. If the or clause fails, the schizophrenia rule fails and the next schizophrenia rule is tried. In the example, the first part of the or clause is satisfied and the rule succeeds.

The interpreter returns to the diagnose\_psychosis rule, and the third premise appends information to the explanation frame. The interpreter returns to the diagnose rule and then to the psyxpert rule in file "control.p" (D-4). The fourth premise of the psyxpert rule prints "The diagnostic phase is now complete." to the screen.

The fifth and final premise in the psyxpert rule, (end \*final\_chart), matches the end rule in file "end.p" (D-7). The variable \*final\_chart is an input variable containing the final explanation frame. The first premise, (print\_heading), matches the print\_heading rule in the same file and uses the built-in putline function to print the heading for the diagnostic report to the screen. The second premise, (print\_patient\_data \*struct), matches the print\_patient\_data rule in the same file and uses the find\_list, return\_fact and printline rules to print the patient's name and age. The third premise, (print\_diagnosis \*struct), matches the print\_diagnosis rule in the same file and also uses the find\_list, return\_fact and printline rules to print the diagnosis and the confidence factor stored in the explanation frame.

The fourth premise, (print\_recommendations \*struct), matches the print\_recommendations rule in the same file. The

print\_recommendations rule uses the find\_list and return\_fact rules to instantiate the variables, \*disorder and \*cf, to the disorder and confidence factor stored in the explanation frame. The premise, (recommendations \*disorder \*cf), matches one of the eight recommendations rules which use the built-in putline function to print the appropriate recommendations to the screen.

The fifth premise of the end rule causes the user interface to ask if the user would like an explanation of the diagnosis. The sixth premise is an or clause, and if the response to the question is positive, invokes the explain\_diagnosis rule in file "explain.p" (D-11) and described in the section on the explanation subsystem. If the response is negative, the explain\_diagnosis rule is simply skipped.

The seventh and last premise of the end rule, (stop\_it), matches the stop\_it rule in the same file and is described in the section on the user interface. In the example, the user chose to end the consultation, so the message is printed and the HC Prolog prompt, ':-', is displayed. The psyxpert rule in file "control.p" (D-4) has been satisfied and the consultation is now over.



## 6. Knowledge Acquisition

Since knowledge is the foundation of an expert system, the sources of knowledge and the process of acquiring it are fundamental to any attempt to build such a system. The knowledge used in Psyxpert came from three sources: the DSM-III diagnostic criteria [26], the AMSIT mental status report format [25], and Dr. Daniel Pearson, Ph.D., M.D., a psychiatrist at the University of Texas Health Science Center at San Antonio. The first knowledge source, the DSM-III diagnostic criteria, provided the basis for the rules that make diagnostic decisions. The second knowledge source, the AMSIT mental status report format, provided questions that gather the facts used by the production rules in the process of diagnostic inference.

Dr. Pearson, the third knowledge source, provided three main services. First, he provided assistance in the process of finding a common ground between the facts gathered during the mental status examination and the production rules written from the DSM-III diagnostic criteria. Secondly, he provided suggestions for the user interface and the explanation subsystem. Thirdly, he provided and ran test cases.

The process of acquiring knowledge and assimilating it into the system proceeded incrementally. First, the problem was divided in a natural way so that a subset of the knowledge could be used to build a prototype system. The prototype was built and tested quickly to check the system

design.

Psyxpert was limited to mental disorders with a prominent psychotic feature as described in Chapter 3. The Psyxpert prototype was further limited to knowledge about schizophrenic disorder. The diagnostic criteria for schizophrenic disorder from the decision tree for differential diagnosis of psychotic disorders in Appendix A of the DSM-III Casebook (figure 1) were encoded in production rules written in Prolog. The form of the production rules and their relationship to the diagnostic criteria in the DSM-III were described in the section on knowledge representation.

At this point, the production rules needed facts to make their inferences, so contrived facts were asserted into Prolog's data base before the consultation was started. The production rule format with asserted facts was tested and able to diagnose schizophrenic disorder. The diagnostic criteria proved amenable to the production rule representation, so a method of gathering and asserting the facts during the consultation was needed.

The AMSIT mental status report format was used as a source for the questions asked about the patient in the initial stage of the consultation. The user interface asks the questions and asserts the answers, with part of the questions, as facts in the data base. This process was described in detail in Chapter 5.

The process of creating a common ground between the diagnostic criteria and the AMSIT questions was begun. Dr.

Pearson translated parameters used in the diagnostic criteria into basic facts to be asked in the initial questioning stage. He suggested rules that could use the facts to infer new facts needed by the production rules representing the diagnostic criteria.

Dr. Pearson suggested that a five to seven point scale be used when asking the user if the patient had a certain condition. A scale this broad would create complexity beyond the scope of this project, so it was limited, in most cases, to a range of three points. The possibility of expanding the scale to five or seven points is discussed in Chapter 7.

The next step was to come up with a set of pertinent questions that would cover all the possible situations beforehand. Dr. Pearson helped to enumerate some possibilities and combine or eliminate unimportant ones to keep down the number of questions.

Then the user interface was created, and Dr. Pearson provided input on the ways that questions should be asked and explanations should be given. He tested the feasibility of combining the AMSIT questions and the DSM-III diagnostic criteria into production rules by running various test cases. He approved of, or suggested changes to the questions, explanations, and results.

The final version of Psyxpert was written using this method of knowledge representation. Knowledge about other mental disorders with psychotic features was added incrementally. Each time knowledge was added, the system was tested with new cases until all the knowledge was included.

Dr. Pearson then performed the final test phase on which the results of this research are based. The test cases and results are described and discussed in Chapter 7.

Incorporating three diverse sources of knowledge was successful. The DSM-III diagnostic criteria provided an explicit, already verbalized template for the system's production rules. Furthermore, the DSM-III diagnostic criteria represent a body of knowledge arrived at through the general consensus of many psychiatric experts.

The AMSIT mental status report format provided a framework for gathering the pertinent facts to be used for inference. There are other formats for the mental status report, however, and they could be consulted for future projects.

A human expert is always necessary to supplement the knowledge gleaned from printed materials. The expert on this project, Dr. Pearson, proved invaluable as a source of information on finding a common ground between the two printed resources. Printed material is limited, because it is static and cannot provide the interactive exchange of information and ideas that a human expert can. Furthermore, Dr. Pearson was invaluable in the test phase and provided the basis for the user interface and suggestions for the explanation subsystem.

I found Dr. Pearson, who by the way is my first cousin, to be easy to work with. We met informally, at my home, on several occasions. Taking the expert away from his busy office kept interruptions to a minimum. The first time we

met, he described psychiatric diagnosis in general. At subsequent meetings we discussed high level details of the project itself. He related several actual patient interviews, always protecting the patient's identity. I found that an expert with previous computer experience, such as Dr. Pearson, already knows the capabilities and limitations of a computer based approach to diagnostic problem solving, and can bring realistic expectations to the software project.

## 7. Results and Discussion

### 7.1 Results

The Psyxpert system is currently in a test/revise stage and has reached the level of a small research prototype. Waterman [3] defines a research prototype as a system that performs successfully on problems within its scope but may fail when given problems near the domain boundary. The failure may be due to incomplete testing and revision.

Psyxpert has been tested by a psychiatrist using thirteen cases from the DSM-III Case Book [27]. In all cases the patient exhibited psychiatric symptoms upon examination or reported a history of psychotic symptoms. At least one case was chosen to test each of the seven mental disorders Psyxpert diagnoses. Dr. Pearson read each case and ran a consultation based on each one. After running the system, Dr. Pearson recommended some changes in the wording of numerous questions. These changes were made and Dr. Pearson approved the results.

Brief Reactive Psychosis, Paranoid Disorder and Schizophreniform Disorder are relatively straightforward disorders and in most cases the diagnosis is also straightforward. One case of Brief Reactive Psychosis, two cases of Paranoid Disorder and one case of Schizophreniform Disorder were chosen from the DSM-III casebook. Psyxpert's diagnosis on all four of these cases was in agreement with the case book's diagnosis.

Psyxpert was tested using three cases of Schizophrenic Disorder from the DSM-III Casebook. In two of the cases, Psyxpert was successful in concluding Schizophrenic Disorder. In the third case Psyxpert missed Schizophrenic Disorder and concluded Schizophreniform Disorder. Schizophreniform Disorder has the same diagnostic criteria as Schizophrenic Disorder with one exception. In Schizophreniform Disorder the duration of the psychotic symptoms must be less than six months. The psychotic symptoms in this case had been occurring for only three months. If this had been the patient's first episode, Psyxpert's diagnosis would have been correct, but the patient had a past history of similar psychotic episodes, making the proper diagnosis Schizophrenic Disorder. This points out a major flaw in the system, which is that it does not have a special consultation mode for cases with a prior diagnosis. The implications of this will be described in the following section.

When the psychotic symptoms are present with a disordered affective component, an accurate diagnosis is often difficult to make. Psyxpert was tested using three cases of Major Affective Disorder with psychotic features and one case of Schizoaffective Disorder. In two of the three cases of Major Affective Disorder with psychotic features, a correct diagnosis was made. In the third case, Psyxpert concluded Schizoaffective Disorder on the basis that the psychotic symptoms might not only be present with the affective component. Psyxpert should have diagnosed Major Affective Disorder with psychotic features, and it could be

easily changed to do this. In the one case of Schizoaffective Disorder, Psyxpert made the correct diagnosis.

Atypical Psychosis is a category used when there are psychotic symptoms with no affective component, yet the criteria for any other psychotic disorder cannot be met. Two cases of Atypical Psychosis were tested. Psyxpert's diagnosis was in agreement with the DSM-III Casebook's diagnosis on both cases.

Psyxpert was also tested by Dr. Pearson with data from three actual cases. Psyxpert's diagnosis of Schizophrenic Disorder on two of the three cases agreed with Dr. Pearson's diagnosis. However, on the third actual case, Psyxpert diagnosed Schizophrenic Disorder and Dr. Pearson diagnosed Borderline Personality Disorder. Borderline Personality Disorder is a difficult diagnosis to make because the patient exhibits signs of many different types of disorders, and the patient may have brief psychotic episodes without being considered psychotic. A case of this type is on the boundary of the system's domain, and Psyxpert should have indicated that it could not diagnose this type of disorder. Rules are currently being revised to correct the problem.

These test cases pointed out some problems with the system. Dr. Pearson sees patients who have been admitted to the psychiatric ward of the Medical Center Hospital. He performs physical examinations, orders laboratory tests when necessary, and performs mental status examinations. Dr. Pearson found that it was difficult to answer all of



Psyxpert's questions on the basis of one interview with the patient. The case book cases were easier to run because more information was provided.

## 7.2 Limitations and Restrictions

Psyxpert is to be used as a consultation aid by a psychiatrist and cannot be used in place of a psychiatrist. The psychiatrist must perform the history, physical examination and the mental status examination. The psychiatrist may then use Psyxpert to ensure consistency between his or her diagnosis and the DSM-III diagnostic criteria.

The domain of this research is Psychiatry. Psychiatry is a broad domain with many classes of mental disorders. The Psyxpert system is restricted to the diagnosis of mental disorders in which psychotic features are present. This decision was made to limit the size of the project and did not affect any of the design decisions.

Psyxpert was not designed to diagnose multiple disorders or monitor patients over time. These capabilities, in this context, would complicate the problem beyond the scope of a master's thesis. Psyxpert would need to be redesigned to incorporate these capabilities.

The implementation of the '?' and 'how' explanations of the explanation subsystem is currently limited due to time constraints. A subset of questions has been implemented to demonstrate the capability, and this could easily be extended

to cover every question. The extension would require only the addition of new knowledge with no change to the existing knowledge or structure.

When asking most questions, Psyxpert uses a three point scale of possible answers. A broader scale of five to seven possible answers could be incorporated into the system. The broader scale would complicate the decision making process by adding more possibilities to consider but would not require a change in the system design.

Psyxpert's questions concerning time are rather awkward. The psychiatrist is asked about the onset of a symptom in terms of weeks, but if the onset is more than a year or two ago, responding in weeks is awkward, and a response in terms of years would be more appropriate. The same argument applies when the onset is less than one week ago, in which case the response should be made in terms of days. This problem could be remedied with a major rewrite of all diagnostic criteria and questions concerning time. A change in the system design would not be necessary.

Psyxpert does not take into account previous diagnoses, so issues concerning what it takes to confirm or overturn a previous diagnosis are not dealt with. This is the system's major shortcoming. Another consultation mode would be needed to deal with patients who are not being diagnosed for the first time. This would require a major change in the system design.

### 7.3 Possible Enhancements

All the limitations mentioned above could be dealt with. Psyxpert's domain could be extended beyond mental disorders with psychotic features to include mental disorders from other DSM-III categories. Psyxpert is designed in such a way that the knowledge base could easily be extended with rules to diagnose subtypes of mental disorders, as well as with rules to diagnose mental disorders in other diagnostic categories. These new rules should be added and tested incrementally.

A knowledge acquisition interface could be added to allow psychiatrists to enter their own knowledge into the system's knowledge base, as well as their own recommendations for treatment or further evaluation. The interface might take the form of an intelligent editing program, and the expert could communicate with it in a restricted subset of English. The expert could enter a new rule, and the interface would analyze the rule and return its understanding of it. The expert would then be allowed to accept the rule or make changes to it. The interface could perform tasks, such as recording revisions and additions, and checking typographic and semantic errors. The interface has the ability to detect inconsistencies between new revisions and existing knowledge in the knowledge base.

The interface could be designed, developed and tested as a separate module and added to the system later. As long as the input is restricted to a subset of the English language

to avoid the problems of English language understanding, the bookkeeping and typographic error checking functions would not be difficult to implement. Semantic error checking is a much more sophisticated ability and is difficult to implement.

Psyxpert's output facilities could be enhanced by printing explanations, diagnostic reports and confidence factors in natural language. Instead of just printing atoms, the explanation frames could be mapped into natural language and then printed as an explanation or diagnostic report. This would be moderately difficult to implement but would not require a redesign of the system just the addition of a translation module. The numerical confidence factor given in the diagnostic report, could easily be translated into a natural language description.

A broader scale of possible answers could be added with relatively few design changes, but several rules would have to be added to the knowledge base. The three point scale of confidence factors also could be broadened.

An alternate consultation mode could be incorporated to allow the user to run the system differently when the patient's history showed a previous diagnosis. This mode would allow the user to enter the previous diagnosis, and the system would ask an extra set of questions in addition to the initial questions about the mental status examination. The system then would either confirm the previous diagnosis or overturn it and give a new diagnosis.

If the system were to be rebuilt from the beginning, the

design could be changed to incorporate alternate consultation modes. Otherwise, rules to handle the problem when it arose could be added to the knowledge base. In either case, knowledge about when to confirm or overrule a previous diagnosis would need to be added to the system.

Psyxpert could be enhanced by giving it the ability to store information gathered and generated during the consultation for each patient. When a consultation was begun, the system could look for any previous information it had on the patient and utilize this information in the diagnostic process. The storage and retrieval of the information could easily be implemented with no changes to the design of the system. The use of any previous information would require an alternate consultation mode, such as the one described previously, and a redesign of the system.

## 8. Conclusions

The conclusions drawn from the results of the Psyxpert research project are examined from the following perspectives: Psyxpert as a psychiatric system, as a system implemented in Prolog, as an academic learning experience and as a clinical tool. This research has demonstrated that psychiatric knowledge may be represented in the form of production rules. The psychiatric knowledge used to classify patients according to the DSM-III classification system is very high level. No knowledge about the causes of the various disorders is needed, therefore, production rules are appropriate for representing knowledge in this domain. If a deeper, causal knowledge is needed, another method of knowledge representation should be used.

The Psyxpert research has shown that Prolog is a good implementation language for expert systems in this domain because the knowledge can be adequately represented by production rules. If the domain requires causal knowledge, a method other than production rules should be employed, and in these cases, Prolog would not be a good choice for an implementation language. Prolog's built-in inference engine and implicit pattern matching capabilities permit domain knowledge to be written and added to the system very early on. This allows systems to be prototyped very quickly, an important stage in expert systems development.

Psyxpert has been a success in terms of an academic learning experience. I have learned about the field of

artificial intelligence in general, and expert systems in particular. I have explored artificial intelligence techniques for knowledge representation, reasoning with uncertainty, and explanation. I have a feel for where expert systems research has been, where it is now, and where it is going in the future.

I now have experience in the development of a large software project. I took the system from the proposal stage through prototyping, testing, revising and retesting. I now consider myself an experienced expert system developer, as well as, an experienced Prolog programmer. In fact, I credit my current position as a research analyst in an artificial intelligence group, to my work on Psyxpert.

Finally, I have learned about psychiatric diagnosis of psychotic disorders, and medical diagnosis in general. I have gained practical experience working with a physician. In the future, I plan to design and develop an intelligent tutoring system for medical students in a sub-specialty of medicine.

The research has shown that the Psyxpert system is not a clinically effective tool. A study of the clinical situations where a computer might be utilized needs to be made, and then the results applied to the design stage of such a system. The conditions under which psychiatrists might take advantage of the knowledge available to them from a computerized consultant need to be characterized. In order for expert systems to be used by physicians, the rationale behind the high level rules must be made accessible through

the explanation subsystem. Justification for the use of a particular rule is needed, in addition to simply showing the rule the system used.

**Acknowledgements**-- I want to thank Al Biles, M.S., and Walter Wolf, PhD. - my thesis committee members - for their support, time and good advise; Pamela Fink, PhD. - my expert systems expert - for her expertise, guidance and time; Daniel Pearson, MD. - my psychiatric expert - for his help, interest and enthusiasm; E.P. Hanavan, PhD. and UTSA for the use of the VAX-11/780; George Dodds - medical student and my husband - for his creative insights into computers and medicine; and finally Shannon and Elizabeth Dodds - my daughters - for giving me quiet time to develop and write the Psyxpert system.



## References

1. A. Barr and E.A. Feigenbaum (eds.), The Handbook of Artificial Intelligence, Vol. 1. Kaufmann, Los Altos, Calif., (1981).
2. F. Hayes-Roth, D. Waterman and D. Lenat (eds.), Building Expert Systems, Vol. 1.. Addison-Wesley, New York, (1983).
3. D. Waterman, A Guide to Expert Systems, Addison-Wesley, New York, (1986).
4. I. Goldstein and S. Papert, Artificial Intelligence, Language, and the Study of Knowledge, Cognitive Science, Vol. 1, No. 1, (1977).
5. E.A. Feigenbaum, The art of artificial intelligence: Themes and case studies of knowledge engineering, Proc. Intl. Joint Conf. Artific. Intell., 1014-1029, (1977).
6. T. Winograd, Frame representations and the declarative/procedural controversy, in Representation and Understanding, D. Bobrow and A. Collins (eds.), pp. 185-210. Academic Press, New York, (1975).
7. R. Davis and J.J. King, An overview of production systems, in Machine Intelligence 8, E.W. Elcock and D. Michie (eds.), pp. 300-332, Wiley & Sons, New York, (1977).
8. M. Minsky, A Framework for Representing Knowledge, The Psychology of Computer Vision, Winston, P. (ed.), pp. 211-277, McGraw-Hill, (1975).
9. M.I. Bernstein, Knowledge-based Systems: A Tutorial, Systems Development Corporation, Santa Monica, Calif., (1977).
10. W.A. Woods, What's in a Link: Foundations for Semantic Networks, in Bobrow, D. and Collins, A., (eds.), Representation and Understanding, Academic Press, New York, (1975).
11. B.G. Buchanan and E.H. Shortliffe (eds.), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, New York, (1984).
12. L. Zadeh, Fuzzy Sets, Information and Control, Vol. 8, 338-353, (1965).
13. R.O. Duda, J.G. Gaschnig and P.E. Hart, Model Design in the Prospector Consultant System for Mineral Exploration, pp. 153-167, in Michie, D., (ed.), Expert Systems in the Micro-electronic Age, Edinburgh University Press, Edinburgh, (1979).
14. E.H. Shortliffe, Computer-Based Medical Consultations: MYCIN. American Elsevier, New York, (1976).
15. C.A. Kulikowski, Artificial intelligence methods and systems for medical consultation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 464-476, (1980).
16. W. van Melle, A domain-independent production-rule system for consultation programs, Proc. Intl. Joint Conf. Artific. Intell., 942-947, (1979).

17. J.C. Kunz, R.J. Fallat et. al., A Physiological Rule-Based System For Interpreting Pulmonary Function Test Results, Rept. HPP-78-19, Heuristic Programming Project, Computer Science Department, Stanford University, Stanford, Calif., (1978).
18. J.S. Aikins, Prototypes and production rules: an approach to knowledge representation for hypothesis formation, Proc. Intl. Joint Conf. Artific. Intell., 1-3, (1979).
19. J.S. Aikins, Prototypes and production rules: a knowledge representation for computer consultations, Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, Calif., (1980).
20. L.M. Fagan, et. al., Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit, Proc. Intl. Joint Conf. Artific. Intell., 260-262, (1979).
21. W. Clancey and R. Letsinger, NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching, Proc. Intl. Joint Conf. Artific. Intell., 829-836, (1981).
22. American Psychiatric Association, Psychiatric Glossary, American Psychiatric Press, Washington, D.C., (1984).
23. D.F. Klein, et. al., Diagnosis and Drug Treatment of Psychiatric Disorders: Adults and Children, 2nd Edition, Williams and Wilkins, Baltimore, Md., (1980).
24. H.H. Goldman (ed.), Review of General Psychiatry, LANGE Medical Publications, Los Altos, Calif., (1984).
25. David S. Fuller, AMSIT: A Description of the Patient's Current Mental Status.
26. American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders, Third Edition, APA, Washington, D.C., (1980).
27. R.L. Spitzer et. al., DSM-III Case Book, American Psychiatric Association, Washington, D.C., (1981).
28. W.F. Clocksin and C.S. Mellish, Programming in Prolog, Springer-Verlag, Berlin, (1981).
29. E. Charniak and D. McDermott, Introduction to Artificial Intelligence, Addison-Wesley, (1985).
30. J. Roach and G. Fowler, The HC Manual: Virginia Tech Prolog, Virginia Polytechnic Institute and State University, Blacksburg, Va., (1983).

## Bibliography

### General Artificial Intelligence

Barr, A. and Feigenbaum, E.A. (eds.), The Handbook of Artificial Intelligence, vol. 1. Kaufmann, Los Altos, CA (1981).

Charniak, E. and McDermott, D., Introduction to Artificial Intelligence, Addison-Wesley, (1985).

Goldstein, I. and Papert, S., Artificial Intelligence, Language, and the Study of Knowledge, Cognitive Science, Vol. 1, No. 1, (1977).

### Prolog

Clocksin, W.F. and Mellish, C.S., Programming in Prolog, Springer-Verlag, Berlin (1981).

McDermott, D., "The Prolog Phenomena", SIGART Newsletter, No. 72, pp. 16-20, (1980).

Roach, J. and Fowler, G., The HC Manual: Virginia Tech Prolog, Virginia Polytechnic Institute and State University, Blacksburg, Va. (1983).

### General Expert Systems

Bernstein, M.I., Knowledge-based Systems: A Tutorial, Systems Development Corporation, Santa Monica, Calif. (1977).

Buchanan, B.G., New Research on Expert Systems, in Machine Intelligence 10, Hayes, J.E., Michie, D. and Poa, Y.H., (Eds.), pp. 269-299, Edinburgh University Press, Edinburgh, (1982).

Buchanan, B.G. and Duda, R.O., Principles of Rule-based Expert Systems, in Advances in Computers, Vol. 22, M. Yovits (Ed.), Academic Press, New York, (1982).

Davis, R., Expert Systems: Where Are We? and Where Do We Go From Here?, The AI Magazine, 3-22, (1982).

Feigenbaum, E.A., The art of artificial intelligence: Themes and case studies of knowledge engineering, Proc. Intl. Joint Conf. Artific. Intell., 1014-1029, (1977).

Hart, P., Directions for AI in the Eighties, SIGART Newsletter, 79, (1982).

Hayes-Roth, F., Waterman, D. and Lenat, D. (eds.), Building Expert Systems, vol. 1. Addison-Wesley, New York (1983).

McDermott, J., Extracting Knowledge From Expert Systems, Proc. Intl. Joint Conf. Artific. Intell., 100-107, (1983).

Michie, D., High-Road Low-Road Programs, AI Magazine 3, 21-22, (1982).

Michie, D., Expert Systems, The Computer Journal, Vol. 23, 369-376, (1980).

Nau, D., Expert Computer Systems, Computer, pp. 63-68, Feb., (1983).

Sauers, R. and Walsh, R., On the Requirements of Future Expert Systems, Proc. Intl. Joint Conf. Artific. Intell., 110-115, (1983).

Stefik, M., et. al., The Organization of Expert Systems, A Tutorial, Artificial Intelligence, 18, pp. 135-173, (1982).

Waterman, D., A Guide to Expert Systems, Addison-Wesley, New York (1986).

Waterman, D. and Hayes-Roth, F. (Eds.), Pattern-directed Inference Systems, Academic Press, New York, (1978).

### Expert Systems in Medicine

Aikins, J.S., Prototypical Knowledge for Expert Systems, Artificial Intelligence, 20, 163-210, (1983).

Aikins, J.S., The Use of Models in a Rule-based Consultation System, Proc. Intl. Joint Conf. Artific. Intell., 788-789, (1981).

Aikins, J.S., Prototypes and production rules: a knowledge representation for computer consultations, Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, Calif. (1980).

Aikins, J.S., Representation of Control Knowledge in Expert Systems, AAAI 1, 121-123, (1980).

Aikins, J.S., Prototypes and production rules: an approach to knowledge representation for hypothesis formation, Proc. Intl. Joint Conf. Artific. Intell., 1-3, (1979).

Bravos, A., et. al., Application of the CSRL Language to the Design of Diagnostic Expert Systems: The MOODIS experience,

a Preliminary Report, SPIE Vol. 635, Applications of Artificial Intelligence III, 76-81, (1986).

Buchanan, B.G. and Shortliffe, E.H. (eds.), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, New York (1984).

Chandrasekaran, B. and Mittal, S., Deep Versus compiled Knowledge approaches to diagnostic problem solving, Int. Jour. Man-Machine Studies, 425-436, (1983).

Chandrasekaran, B., et. al., MDX and Related Medical Decision-Making Systems, Proc. Intl. Joint Conf. Artific. Intell., 1055-1058, (1981).

Chandrasekaran, B., et. al., An Approach to Medical Diagnosis Based upon Conceptual Procedures, Proc. Intl. Joint Conf. Artific. Intell., 134-142, (1979).

Clancey, W. and Letsinger, R., NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching, Proc. Intl. Joint Conf. Artific. Intell., 829-836, (1981).

Fagan, L.M., et. al., Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit, Proc. Intl. Joint Conf. Artific. Intell., 260-262, (1979).

Kulikowski, C.A., Artificial intelligence methods and systems for medical consultation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 464-476, (1980).

Kunz, J.C., Fallat, R.J., et. al., A Physiological Rule-Based System For Interpreting Pulmonary Function Test Results, Rept. HPP-78-19, Heuristic Programming Project, Computer Science Department, Stanford University, Stanford, Calif. (1978).

Miller, P., Medical Plan-analysis: The Attending System, Proc. Intl. Joint Conf. Artific. Intell., 239-241, (1983).

Patil, R., Szolovits and Schwartz, Causal Understanding of Patient Illness in Medical Diagnosis, Proc. Intl. Joint Conf. Artific. Intell., 893-899, (1981).

Pople, H.E., The Formation of Composite Hypotheses in Diagnostic Problem Solving: An exercise in Synthetic Reasoning, Proc. Intl. Joint Conf. Artific. Intell., 1030-1037, (1977).

Pople, H.E., Myers and Miller, Dialog: A Model of Diagnostic

Logic for Internal Medicine, Proc. Intl. Joint Conf. Artific. Intell., 848-855, (1975).

Roach, J. and Lee, S., An Expert System for Information on Pharmacology and Drug Interactions, Computers in Biology and Medicine, vol. 15, No. 1, 11-23, (1985).

Rubin, A., The Role of Hypotheses in Medical Diagnosis, Proc. Intl. Joint Conf. Artific. Intell., 856-862, (1975).

Shortliffe, E.H., Reasoning Methods in Medical Consultation Systems: Artificial Intelligence Approaches, Computer Programs in Biomedicine, 18, 5-14, (1984).

Shortliffe, E.H., Consultation Systems for Physicians: The Role of Artificial Intelligence. Readings in AI, (1981).

Shortliffe, E.H., Computer-Based Medical Consultations: MYCIN. American Elsevier, New York (1976).

Shortliffe, E.H., et. al., Oncocin: An Expert System for Oncology Protocol Management, Proc. Intl. Joint Conf. Artific. Intell., 876-880, (1981).

Swartout, W.R., Explaining and Justifying Expert Consulting Programs, Proc. Intl. Joint Conf. Artific. Intell., 815-818, (1981).

Swartout, W.R., A digitalis therapy advisor with explanation, Proc. Intl. Joint Conf. Artific. Intell., 819-825, (1977).

Trigoboff, and Kulikowski, C.A., Propagation of Inferences in a Semantic Net, Proc. Intl. Joint Conf. Artific. Intell., 274-280, (1977).

van Melle, W., A domain-independent production-rule system for consultation programs, Proc. Intl. Joint Conf. Artific. Intell., 942-947, (1979).

Weiss, S.M. and Kulikowski, C.A., Expert: A System for Developing Consultation Models, Proc. Intl. Joint Conf. Artific. Intell., 942-947, (1979).

Weiss, S.M., Kulikowski, C.A. and Safir, A Model-based Consultation System for Long-term Management of Glaucoma, Proc. Intl. Joint Conf. Artific. Intell., 826-832, (1977).

Witten, M. and Maloney, D., SYN-PEDS: Syntactical Pediatric Evaluation and Diagnostic System, Computer Applications in Medical Care, 241-247, (1983)

### Expert Systems in Other Fields

Bonnet, A. and Dahan, C., Oil-well Data Interpretation Using Expert System and Pattern Recognition Techniques, Proc. Intl. Joint Conf. Artific. Intell., 185-189, (1983).

Buchanan, B.G. and Feigenbaum, E.A., Dendral and Meta Dendral: Their Applications Dimension, Artificial Intelligence 11, 5-24, (1978).

Davis, R., et. al., Diagnosis Based on Description of Structure and Function, AAAI-82, 137-142, (1982).

Duda, R.O., Gaschnig, J.G. and Hart, P.E., Model Design in the Prospector Consultant System for Mineral Exploration, pp. 153-167, in Michie, D., (ed.), Expert Systems in the Micro-electronic Age, Edinburgh University Press, Edinburgh, (1979).

Fink, P., Control and Integration of Diverse Knowledge in a Diagnostic Expert System, Proc. Intl. Joint Conf. Artific. Intell., (1985).

Fink, P., Lusth, J. and Duran, J., A General Expert System Design for Diagnostic Problem Solving, IEEE Workshop on Principles of Knowledge-based Systems, 45-52, (1984).

Georgeff, M. and Bonollo, U., Procedural Expert Systems, Proc. Intl. Joint Conf. Artific. Intell., 151-157, (1983).

Lenat, D., et. al., Knoesphere: Building Expert Systems With Encyclopedic knowledge, Proc. Intl. Joint Conf. Artific. Intell., 167-169, (1983).

Lusth, J.C., Diagnosis Using a Functional Model, Intl. Conf. Systems, Man and Cybernetics, (1985).

Reinstein, H.C. and Aikins, J., Application Design: Issues in Expert System Architecture, Proc. Intl. Joint Conf. Artific. Intell., 888-894, (1981).

### Knowledge Representation

Davis, R., Buchanan, B.G. and Shortliffe, E.H., Production Rules as a Representation for a Knowledge-based Consultation System, Artificial Intelligence, 8, 15-45, (1977).

Davis, R. and King, J.J., An overview of production systems, in Machine Intelligence 8, E.W. Elcock and D. Michie (Eds.), pp. 300-332, Wiley & Sons, New York (1977).

Georgeff, M., A Framework for Control in Production Systems, Proc. Intl. Joint Conf. Artific. Intell., 328-331, (1979).

Kaminuma, T., Conceptual Lattice: A Unified Model For Medical Inference Processes, Proc. Intl. Joint Conf. Artific. Intell., 451-454, (1979).

Kehler, T., et. al., Industrial Strength Knowledge Bases: Issues and Experiences, Proc. Intl. Joint Conf. Artific. Intell., 108-109, (1983).

Kuipers, B.A., A Frame for Frames, Representation and Understanding, Bobrow, D.G. and Collins, A.M., (eds.), Academic Press, New York, (1975).

Minsky, M., A Framework for Representing Knowledge, The Psychology of Computer Vision, Winston, P. (ed.), pp. 211-277, McGraw-Hill, (1975).

Mizoguchi, R. and Kakusho, O., Hierarchical Production System, Proc. Intl. Joint Conf. Artific. Intell., 586-588, (1979).

Winograd, T., Frame representations and the declarative/procedural controversy, in Representation and Understanding, D. Bobrow and A. Collins (eds.), pp. 185-210. Academic Press, New York (1975).

Woods, W.A., What's in a Link: Foundations for Semantic Networks, in Bobrow, D. and Collins, A., (eds.), Representation and Understanding, Academic Press, New York, (1975).

### Reasoning With Uncertainty

Prade, H., A Synthetic View of Approximate Reasoning Techniques, Proc. Intl. Joint Conf. Artific. Intell., 130-136, (1983).

Quinlan, J.R., Consistency and Plausible Reasoning, Proc. Intl. Joint Conf. Artific. Intell., 137-144, (1983).

Zadeh, L., Fuzzy Sets, Information and Control, Vol. 8, 338-353, (1965).

### Psychiatry

American Psychiatric Association, Psychiatric Glossary, American Psychiatric Press, Washington, D.C. (1984).

American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders, Third Edition, APA, Washington, D.C. (1980).



Fuller, David S. AMSIT: A Description of the Patient's Current Mental Status.

Goldman, H.H., (ed.), Review of General Psychiatry, LANGE Medical Publications, Los Altos, Calif. (1984).

Klein, D.F., et. al., Diagnosis and Drug Treatment of Psychiatric Disorders: Adults and Children, 2nd Edition, Williams and Wilkins Baltimore, Md. (1980).

Spitzer, R.L., et. al., DSM-III Case Book, American Psychiatric Association, Washington, D.C. (1981).

## Appendix A

### Standard Prolog vs. HC Prolog

Appendix A illustrates the syntactical differences between the standard version of Prolog as described by Clocksin and Mellish [28] and the nonstandard HC Prolog developed at Virginia Polytechnic Institute and State University [30].

Standard Prolog:

```
likes(john,X) :- likes(X,wine),
                likes(X,food).
```

HC Prolog:

```
((likes john *X) if (likes *X wine)
                        (likes *X food))
```

Variables in standard Prolog must begin with a capital letter; in HC Prolog they are preceded by a '\*'. Standard Prolog ends a rule or fact with a '.'; HC Prolog uses parentheses to begin and end a rule or fact. Standard Prolog signifies the conjunction of goals by a ','; HC Prolog simply lists the goal inside the main parentheses. Standard Prolog uses ':-' to mean if; HC Prolog simply uses 'if'.

Standard Prolog:

```
printeach([X|Y]) :- write(X), printeach(Y).
```

the goal

```
?- printeach([roses,are,red]).
```

produces

```
roses
```

```
are
```

```
red
```

HC Prolog:

```
((printeach (*X . *Y)) if (print *X) (printeach *Y))
```

the goal

```
:- (printeach (roses are red))
```

produces

```
roses
```

```
are
```

```
red
```

Standard Prolog uses '?-' for the prompt; HC Prolog uses ':-'. Standard Prolog uses [roses,are,red] to indicate a list and '|' for the dot operator, which breaks a list into the head (first element) and the tail (a list of the rest of the elements); HC Prolog uses (roses are red) to indicate a list and the '.' for the dot operator. Standard Prolog uses '[]' to stand for the empty list; HC Prolog uses nil. Some of the built-in operators and functions have different names, such as the write operator in Standard Prolog and the print operator in HC Prolog.

Standard Prolog:

```
read(X)
```

HC Prolog:

```
(:= *X (read))
```

Standard Prolog:

```
consult("filename").
```

HC Prolog:

```
(load "filename")
```

The Standard Prolog `consult` function and the HC Prolog `load` function both assert the rules in `filename` to the end of the database.

Standard Prolog:

```
append([], X, X).
```

```
append([A|B], C, [A|D]) :- append(B, C, D).
```

HC Prolog:

```
((append nil *x *x))
```

```
((append (*a . *b) c (*a . *d)) if (append *b *c *d))
```

---

Function	Standard Prolog	HC Prolog
<hr/>		
or	p1(A);p2(B)	(or(p1 *a)(p2 *b))
<hr/>		
negation	not	!
<hr/>		
cut	!	(cut)
<hr/>		
anonymous variable	_	*
<hr/>		

The HC Prolog version used in the implementation of the Psyxpert system is written in Pascal, and a LISP system is also resident. Therefore, procedures written in LISP or Pascal may be called from the Prolog interpreter and vice versa.

## Appendix B

### Test Case #85

"Miriam and Esther"

## Miriam and Esther

**85** Miriam was hospitalized after her mother called the police because she feared Miriam might hurt both of them. Miriam claimed she was 56 years of age and lived with her 76-year-old "assumed" or "estranged" mother, Esther, and her 12-year-old daughter, Alice. She described Esther as a family friend who had given her and her daughter a room some years ago, but who had increasingly angered her by acting as a mother and a grandmother, invading her privacy, attacking her in her sleep, and jealously turning Alice against her. The night of admission, domestic squabbling had threatened to become violent, she said, which caused Esther to send her to the hospital "for hygiene." The patient expected to leave as soon as the ward social worker could relocate her and Alice in a "condominium or other suitable environment in which to rear my own child, who is coming of age as a young lady." She admitted to a recent sense of confusion, but denied sleep and appetite changes, mood disturbance, and hallucinations. However, she did describe a "whooshing" sound in her "cranium" intermittently over the past several years, which she felt resulted from fluid in her ear; at other times she had felt "very aware" of her own thoughts, but she denied hearing voices.

Miriam gave a vague but complex past history, as follows: She was born 56 years ago in Italy, on November 15, 1924. Her "biologic parents" (as she put it), Louise and William, were wealthy from oil. They took her to their country house in Mt. Vernon, New York, where she spent her childhood. Esther, a family friend, visited often. Miriam recalled people driving Packards and Rolls Royces. She stated she later lived in Europe and North Africa, and was present in Hiroshima when the atom bomb was dropped. This event left her with a steel plate in her head and an "atom brain." She lived with Louise and William from 1957 to 1968. She said she had three husbands and seven children. Her youngest, Alice, fathered by her last husband, was born in 1968, four years after his death. When asked how this could be, she explained that a "tubal infection" had delayed the baby's conception in a "technical way."

According to Miriam, after the birth of Alice she moved in with Esther and enrolled at Hunter College in a special program for middle-aged students, where she excelled in Romance languages. She became an alcoholic, consuming up to a pint of whiskey daily. Once when she didn't drink, she became shaky and broke into a sweat. Following the death in 1973 of her "biologic mother," Louise, Miriam became depressed and lost weight. A "nervous breakdown" landed her in a state mental hospital for three months, where she stopped drinking and improved with medication. For several years thereafter a local "mental hygiene" clinic gave her medication for "stability," in-

cluding Prolixin, which made her hair fall out. Since then she had worked steadily, first for the Board of Education, and then as a home health aide. For the last year she had remained home to care for her child.

Miriam's mother, Esther, related quite a different history, corroborated by family members and clinic staff. Miriam is actually 30 years old and was born October 8, 1950. Esther is 56 and is, in fact, her biologic mother. When Miriam was seven, her father walked out on the family. The next year she and her older sister were sent to Mt. Vernon to live with Esther's middle-class Aunt Louise and Uncle William, probably for financial reasons. Esther visited on weekends.

Miriam was a good student, but had few friends and kept to herself. In 1968, at 17 years of age, she became pregnant by a cousin from Trinidad, whom she never saw again. She finished high school but, shamed, returned to her mother's home to have the baby. Esther cared for them both and took responsibility for rearing her granddaughter, Alice. Miriam attended night classes in business skills for two years at Hunter College, but did poorly. She then worked for a year as a home health aide, but quit because she thought people were against her. She began to hear voices that commented about her actions, and was finally admitted to a state hospital in 1973, where she improved with medication. The voices ceased several months after discharge. She lived at home and worked occasionally as a secretary, but failed a stenography and typing course. In 1977 her mother paid for her to have her own apartment. Miriam mismanaged her money, and was evicted after a year. The stress apparently caused her to become psychotic again (details are not known). She moved back to her mother's, and improved greatly on antipsychotics. She worked inconsistently for a year, again as a home health aide, but then stopped her medication and quit work. She began to call her mother "Esther," rather than "Mother," and began to say she was not her real mother. Friction developed because of Esther's disappointment in Miriam and Miriam's jealousy of the continued mothering role taken by Esther toward Miriam's child. The child clearly preferred Esther.

Miriam spent more time alone in her room, friendless, venturing out only for shopping trips in which she would spend her disability check on expensive clothes. Relatives say she was often belligerent when talking about her mother. She became unkempt and unable to help with the household chores. She began yelling at imaginary people to leave her alone and not touch her. On several occasions, by the time police had been summoned, Miriam had calmed down. However, the night of admission she was out of control, threatening to throw herself and her mother out the window, and was forcibly handcuffed and brought to the hospital.

According to her mother and family members, there is no history of alcohol (or drug) abuse.

Miriam's mental status in numerous interviews was characterized by calm, socially appropriate behavior. She was obese and homely but tastefully dressed. Speech and movement were of normal tempo and quantity. Her affect was constricted, although at times she seemed pedantic and slightly haughty. Contained anger and sarcasm were apparent during a joint interview with her mother. Thought processes were slightly loose, vague, and circumstantial. Most striking was her odd language, ranging from idiosyncratic usage — "my assumed mother," "my estranged mother" — to neologisms. ("Medicine makes me incognizant. I am not correlative enough." "My mother does not accreditize me." "The hospital will have my records if they are consor-tive; they must have a litigation department." "My cousin was a devasive schizoid.").

When confronted with inconsistencies in her account of her life, Miriam only smiled or giggled. While hospitalized she admitted neither to currently hearing the "whooshing" sound nor to any hallucinations. Nurses reported that when unknowingly observed, she acted as if she were aware of nonexistent beings.

### Discussion of Miriam and Esther

This woman clearly has an illness with prominent psychotic features, the most notable being a delusion that the woman who claims to be her mother is actually only a family friend. (This delusion seems to be a variant of the Capgras syndrome, in which the individual believes that one or more people in his or her environment are actually imposters who either look exactly or almost exactly like the people whose roles they have assumed.) Other bizarre delusions include the belief that the conception of her youngest child was delayed several years beyond the death of the biologic father and that she was in Hiroshima when the atom bomb was dropped, leaving her with a steel plate in her head.

Although she denies hallucinations, the sound in her head probably is an auditory hallucination; her mother claims that she has heard voices in the past.

The absence of a known organic factor that could account for the symptoms, the deterioration in functioning over several years, and the bizarre delusions and hallucinations clearly establish the diagnosis as Schizophrenia. The prominence of the persecutory delusions indicates the Paranoid subtype. The course of her illness is charac-

terized as chronic (ill for more than two years) with a current acute exacerbation (reemergence of prominent psychotic symptoms).

An unusual feature of this case is what is sometimes referred to as pseudologia fantastica, the presentation of fantastic and elaborate details about oneself that are completely false but that the patient appears to believe. In Miriam's case this is illustrated by her account of her past history, being born in Italy, growing up with wealthy parents, later living in Europe and North Africa, and her claim of having had a serious alcohol problem (which her family denied).

### DSM-III Diagnosis:

**Axis I: 295.34 Schizophrenia, Paranoid Type, Chronic with Acute Exacerbation (p. 191)**



## Appendix C

### Log Run

```
((connect stdout fd1))
:-(load "psyxpert.p")
((load "psyxpert.p"))
:-(start)
```

```
t
t
t
t
t
t
t
t
t
t
t
t
t
t
t
t
```

Welcome to the Psyxpert System.

Do you want to create a log of the current session?

- 1) yes
- 2) no

2

Would you like help to run the Psyxpert system?

- 1) yes
- 2) no

1

The following information is intended to help the user run the Psyxpert system. This system is designed to be used by a qualified psychiatrist as a consultation aide in the diagnosis of psychiatric disorders in accordance with the DSM-III classification scheme. The system is not to be used in place of a psychiatrist.

The system is to be used after the psychiatrist has taken a history, performed a physical examination, and given a mental status examination. The psychiatrist should answer the questions based upon these findings. The answers to most questions must be selected from a menu by entering the number corresponding to the answer that in the psychiatrist's judgement best fits the facts. The patient's name must be entered enclosed in parenthesis.

If you would like to see more type "yes".  
yes

During the consultation, the explanation facility provides the following types of explanations:

- 1) If the psychiatrist does not understand a question, entering "?" will provide a detailed explanation of the question and the question will be asked again.
- 2) If the psychiatrist wants to know why a question was

- asked, typing "why" will give the goal the system is trying to satisfy and the question will be asked again.
- 3) If the psychiatrist wants to know how the answer to a question may be determined, typing "how" will give ways to determine the answer and the question will be asked again.

If you would like to see more type "yes".  
yes

After a diagnosis is given, the explanation facility provides the following explanation:

- 1) The psychiatrist is asked if he/she would like an explanation of the diagnosis, if yes is chosen an explanation of the diagnosis with the facts is given.

The psychiatrist is then given the options of:

- 1) ending the consultation,
- 2) rerunning the consultation or
- 3) exiting from the interpreter and returning to the operating system command level.

At any time during the consultation the psychiatrist may abort the consultation by typing "stop" in response to a question.

Begin the consultation.

Please enter the patient's name.

Miriam

Please enter the patient's age.

30

Please answer the following questions based upon your mental status examination findings.

Rank the patient's current mood along the following 7-point depression-elation continuum :

- 1) severely depressed
- 2) moderately depressed
- 3) mildly depressed
- 4) euthymic
- 5) mildly elated
- 6) moderately elated
- 7) severely elated

?  
The patient's mood is the dominant emotion for the duration of the interview.

Rank the patient's current mood along the following 7-point depression-elation continuum :

- 1) severely depressed
- 2) moderately depressed
- 3) mildly depressed
- 4) euthymic
- 5) mildly elated
- 6) moderately elated
- 7) severely elated

5  
Rank the patient's range of affect :

- 1) markedly\_decreased
- 2) moderately\_decreased
- 3) mildly\_decreased
- 4) normal
- 5) mildly\_increased
- 6) moderately\_increased
- 7) markedly\_increased

?

Affect refers to an immediately expressed and observed emotion. Normally affect changes repeatedly through an interview, always in synchrony with what the person is currently saying. Range refers to the extent to which both emotional highs and lows appear in the interview.

Rank the patient's range of affect :

- 1) markedly\_decreased
- 2) moderately\_decreased
- 3) mildly\_decreased
- 4) normal
- 5) mildly\_increased
- 6) moderately\_increased
- 7) markedly\_increased

2

Rank the patient's intensity of affect :

- 1) markedly\_decreased
- 2) moderately\_decreased
- 3) mildly\_decreased
- 4) normal
- 5) mildly\_increased
- 6) moderately\_increased
- 7) markedly\_increased

?

Intensity of affect is the amplitude of emotional expression.

Rank the patient's intensity of affect :

- 1) markedly\_decreased
- 2) moderately\_decreased
- 3) mildly\_decreased
- 4) normal
- 5) mildly\_increased
- 6) moderately\_increased
- 7) markedly\_increased

2

Is the patient's affect labile?

- 1) yes
- 2) no
- 3) unsure

?

Lability is evidenced by a rapid, extreme and brief swing of emotion followed by a return to the previous level.

Is the patient's affect labile?

- 1) yes
- 2) no
- 3) unsure

2

Is the patient's affect always appropriate?

- 1) yes
- 2) no
- 3) unsure

?

Appropriateness refers to whether the emotion at a particular moment is the one expected for the patient's currently-expressed thought.

Is the patient's affect always appropriate?

- 1) yes
- 2) no
- 3) unsure

1

Rate the patient's sensorium :

- 1) clear
- 2) impaired
- 3) unsure

?

The sensorium describes the condition of a subject relative to the subject's mental clarity and is evaluated in terms of orientation, memory and calculating ability.

Rate the patient's sensorium :

- 1) clear
- 2) impaired
- 3) unsure

how

HOW?

Orientation may be evaluated by asking the subject the month date, year and place. Memory may be tested by asking the subject to recall recent events. Calculating ability may be tested by asking the subject to perform serial seven subtractions (the subject must have had the skill previously).

Rate the patient's sensorium :

- 1) clear
- 2) impaired
- 3) unsure

1

Rate the patient's intelligence :

- 1) below\_average
- 2) average
- 3) above\_average

2

Is the patient's form of thought coherent?

- 1) yes
- 2) no
- 3) unsure

1

Is the patient's form of thought logical?

- 1) yes
- 2) no
- 3) unsure

2

Is the patient's form of thought goal-directed?

- 1) yes
- 2) no
- 3) unsure

2

Describe the patient's pattern of associations :

- 1) flighty
- 2) loose
- 3) blocked
- 4) normal
- 5) other

2

Does the patient show evidence of auditory experiences in the

absence of an external stimulus?

- 1) yes
- 2) no
- 3) unsure

1

Does the patient show evidence of visual experiences in the absence of an external stimulus?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient show evidence of other sensory experiences in the absence of an external stimulus?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient have ideas of reference?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient have illusions?

- 1) yes
- 2) no
- 3) unsure

2

Is the patient experiencing depersonalization?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient have a distortion of body image?

- 1) yes
- 2) no
- 3) unsure

1

Does the patient show evidence of delusions of persecution?

- 1) yes
- 2) no
- 3) unsure

1

Does the patient show evidence of delusions of grandeur?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient show evidence of somatic delusions?

- 1) yes
- 2) no
- 3) unsure

2

Does the patient show evidence of delusions of jealousy?

- 1) yes
- 2) no
- 3) unsure

2

Enter any other central themes of the patient's content of thought or enter none. (Enclose multiple answers in parenthesis).  
none

Is the patient's content of thought impoverished?

- 1) yes
- 2) no
- 3) unsure

1

Describe the patient's judgement :

- 1) good
- 2) fair
- 3) poor

?

Judgement involves the ability to make appropriate decisions concerning what to do in occupational, economic and interpersonal situations.

Describe the patient's judgement :

- 1) good
- 2) fair
- 3) poor

how

HOW?

The best tests of judgement are customized to the patient's life experiences. A housewife, for instance, might be asked how she would spend \$40 to buy groceries for her family to last a week if there were no food in the house.

Describe the patient's judgement :

- 1) good
- 2) fair
- 3) poor

3

Rate the patient's abstracting ability :

- 1) abstract
- 2) concrete
- 3) unsure

?

Abstracting ability is the ability to think in generalizations.

Rate the patient's abstracting ability :

- 1) abstract
- 2) concrete
- 3) unsure

how

HOW?

Abstracting ability may be tested by asking the patient to identify the way in which objects are similar, e.g., an apple and an orange or a table and a chair.

Rate the patient's abstracting ability :

- 1) abstract
- 2) concrete
- 3) unsure

3

Does the patient have any insight into his/her problem?

- 1) yes
- 2) no
- 3) unsure

3

Begin the diagnostic phase.

Is there evidence from the patient's history of a specific organic factor that is judged to be etiologically related to the disturbance?

- 1) yes

- 2) no
- 3) unsure

2

Is there evidence from the physical examination of a specific organic factor that is judged to be etiologically related to the disturbance?

- 1) yes
- 2) no
- 3) unsure

2

Is there evidence from laboratory tests of a specific organic factor that is judged to be etiologically related to the disturbance?

- 1) yes
- 2) no
- 3) unsure

why  
WHY?

Because the possibility of ORGANIC\_BRAIN\_DISORDER is under consideration.

The facts are :

NO\_ORGANIC\_HISTORY  
NO\_ORGANIC\_PHYSICAL

-----  
Is there evidence from laboratory tests of a specific organic factor that is judged to be etiologically related to the disturbance?

- 1) yes
- 2) no
- 3) unsure

2

How many weeks ago did the psychotic symptoms first appear?

208

Are the psychotic symptoms :

- 1) present
- 2) not\_present
- 3) unsure

why  
WHY?

Because the possibility of BRIEF\_REACTIVE\_PSYCHOSIS is under consideration.

Trying to determine IF\_DURATION\_OF\_PSYCHOSIS\_LESS\_THAN\_2\_WEEKS

The facts are :

DELUSIONS

-----  
Are the psychotic symptoms :

- 1) present
- 2) not\_present
- 3) unsure

1

Would you characterize the patient's delusions as bizarre?

- 1) yes
- 2) no
- 3) unsure

why  
WHY?

Because the possibility of PARANOID\_DISORDER is under consideration.

Trying to determine PSYCHOLOGICAL\_DISTURBANCE

The facts are :



## DELUSIONS

DURATION\_PSYCHOSIS\_GREATER\_THAN\_2\_WEEKS

-----  
Would you characterize the patient's delusions as bizarre?

- 1) yes
- 2) no
- 3) unsure

1

Has there been a deterioration from a previous level of functioning in work, social relations and self-care?

- 1) yes
- 2) no
- 3) unsure

why

WHY?

Because the possibility of SCHIZOPHRENIFORM\_DISORDER is under consideration.

The facts are :

DELUSIONS

DURATION\_PSYCHOSIS\_GREATER\_THAN\_2\_WEEKS

PSYCHOLOGICAL\_DISTURBANCE

BIZARRE\_DELUSIONS

-----  
Has there been a deterioration from a previous level of functioning in work, social relations and self-care?

- 1) yes
- 2) no
- 3) unsure

1

The diagnostic phase is now complete.

-----  
DIAGNOSTIC REPORT:

Patient : Miriam

Age : 30

Diagnosis : SCHIZOPHRENIC\_DISORDER

Confidence factor : 2

Recommendation :

Treatment of the Schizophrenic disorder.

-----  
Would you like an explanation of the diagnosis?

- 1) yes
- 2) no

1

-----  
DIAGNOSTIC EXPLANATION

Diagnosis : SCHIZOPHRENIC\_DISORDER

Confidence factor : 2

Based upon the following facts :

DELUSIONS

DURATION\_PSYCHOSIS\_GREATER\_THAN\_2\_WEEKS

PSYCHOLOGICAL\_DISTURBANCE

BIZARRE\_DELUSIONS

FUNCTIONING\_IMPAIRED

ONSET\_PRIOR\_TO\_AGE\_45

DURATION\_GREATER\_THAN\_6\_MONTHS

```
-----  
Do you want to :  
1) end_the_consultation  
2) rerun_Psyxpert  
3) exit_the_interpreter  
1  
End the consultation.  
:-(disconnect stdout)
```

## **Appendix D**

### **Prolog Code Listed Alphabetically by Filename**

[illegible]

)

```
((log_session no))
```

```
((log_session yes) if
  (question 98 *a *s)
  (open *a fd1 w)
  (connect stdout fd1)
)
```

```
((initial_info *struct) if
  (question 1 *answ *struct)
  (cut)
  (question 2 *answ1 *struct)
  (cut)
  (append ((NAME *answ)) ((AGE *answ1)) *struct)
)
```

```
((help_file1) if
  (putline
  "
  The following information is intended to help the user run
  the Psyxpert system. This system is designed to be used by a
  qualified psychiatrist as a consultation aide in the diagnosis
  of psychiatric disorders in accordance with the DSM-III
  classification scheme. The system is not to be used in
  place of a psychiatrist.
```

```
The system is to be used after the psychiatrist has taken a
history, performed a physical examination, and given a mental
status examination. The psychiatrist should answer the
questions based upon these findings. The answers to most
questions must be selected from a menu by entering the number
corresponding to the answer that in the psychiatrist's
judgement best fits the facts. The patient's name must be
entered enclosed in parenthesis.
```

```
"
)
```

```
((help_file2) if
  (putline
```

```
During the consultation, the explanation facility provides
the following types of explanations:
```

- 1) If the psychiatrist does not understand a question, entering "\"?" will provide a detailed explanation of the question and the question will be asked again.
- 2) If the psychiatrist wants to know why a question was asked, typing "\"why\" will give the goal the system is trying to satisfy and the question will be asked again.
- 3) If the psychiatrist wants to know how the answer to a question may be determined, typing "\"how\" will give ways to determine the answer and the question

```

        will be asked again.
    ")
}

```

```

((help_file3) if
{putline
"

```

After a diagnosis is given, the explanation facility provides the following explanation:

- 1) The psychiatrist is asked if he/she would like an explanation of the diagnosis, if yes is chosen an explanation of the diagnosis with the facts is given.

The psychiatrist is then given the options of:

- 1) ending the consultation,
- 2) rerunning the consultation or
- 3) exiting from the interpreter and returning to the operating system command level.

At any time during the consultation the psychiatrist may abort the consultation by typing "\"stop\" in response to a question.

```

    ")
}
)

```

```
;      CONTROL. P
;
;::::::::::::::::::::::::::::::::::::::::::::::::::;
;      PSYXPERT
;
;      Mary Overby
;      April 2, 1986
;
;      This module controls the sequence of the execution of the
;      system.
;
;      Rules:
;
;          begin
;              diagnose
;          end
;
;::::::::::::::::::::::::::::::::::::::::::::::::::;

(assert
((psyxpert) if
(begin *chart)
(putline
"Begin the diagnostic phase.
")
(diagnose *chart *final_chart)
(putline
"The diagnostic phase is now complete.
")
(end *final_chart)
)
)
```







```

(printline "Patient : " *name)
(printline "Age : " *age)
)

((print_patient_data *struct))

((print_diagnosis *struct) if
(find_list *struct *out1 DISORDER *rest1)
(= *out1 nil)
(return_fact 1 *out1 *disorder)
(find_list *struct *out2 CF2 *rest2)
(= *out2 nil)
(return_fact 1 *out2 *cf)
(printline "Diagnosis : " *disorder)
(printline "Confidence factor : " *cf)
)

((print_diagnosis *struct) if
(find_list *struct *out1 CATEGORY *rest1)
(= *out1 nil)
(return_fact 1 *out1 NOT_SPECIFIED)
(putline
"Diagnosis : NO DIAGNOSIS ")
)

((print_diagnosis *struct) if
(find_list *struct *out1 CATEGORY *rest1)
(= *out1 nil)
(return_fact 1 *out1 *category)
(find_list *struct *out2 CF *rest2)
(= *out2 nil)
(return_fact 1 *out2 *cf)
(printline "Diagnosis : " *category)
(putline " Not differentiated")
(printline "Confidence factor : " *cf)
)

((print_diagnosis *struct))

((print_recommendations *struct) if
(find_list *struct *out1 DISORDER *rest1)
(= *out1 nil)
(return_fact 1 *out1 *disorder)
(find_list *struct *out2 CF2 *rest2)
(= *out2 nil)
(return_fact 1 *out2 *cf)
(recommendations *disorder *cf)
(putline
"-----")
)

((print_recommendations *struct) if
(find_list *struct *out1 CATEGORY *rest1)
(= *out1 nil)
(return_fact 1 *out1 NOT_SPECIFIED)
(recommendations NOT_SPECIFIED nil)
(putline
"-----")
)

```

```

((print_recommendations *struct) if
  (find_list *struct *out1 CATEGORY *rest1)
  (!= *out1 nil)
  (return_fact 1 *out1 *category)
  (find_list *struct *out2 CF *rest2)
  (!= *out2 nil)
  (return_fact 1 *out2 *cf)
  (recommendations *category *cf)
  (putline
    "-----")
)

((print_recommendations *struct))

((recommendations ORGANIC_BRAIN_DISORDER 2) if
  (putline
    "Recommendation :
    The Psyxpert system cannot differentiate Organic Brain
    Disorders at this time. Further evaluation may be needed
    to determine the type of Organic Brain Disorder.
    ")
  )

((recommendations ORGANIC_BRAIN_DISORDER 1) if
  (putline
    "Recommendation :
    Further evaluation is needed to determine if the patient
    does in fact have an Organic Brain Disorder.
    ")
  )

((recommendations PSYCHOTIC_DISORDER 2) if
  (putline
    "Recommendation :
    The Psyxpert system cannot differentiate this type of
    psychotic disorder.
    ")
  )

((recommendations PSYCHOTIC_DISORDER 1) if
  (putline
    "Recommendation :
    Further evaluation is needed to determine if the patient
    does in fact have a psychotic disorder.
    ")
  )

((recommendations MOOD_DISTURBANCE *any) if
  (putline
    "Recommendation :
    The Psyxpert system cannot differentiate Mood Disturbances.
    ")
  )

((recommendations SCHIZOPHRENIC_DISORDER *any) if
  (putline
    "Recommendation :
    Treatment of the Schizophrenic disorder.
    ")
  )

```

```

((recommendations NOT_SPECIFIED nil) if
(putline
"Recommendation :
The Psyxpert system is not designed to diagnose cases
of this type.
")
)

((recommendations *any *all) if
(putline
"Recommendation :
Not implemented for this disorder.")
)

((stop_it) if
(question 99 *answ *struct)
(log_off)
(or((= *answ end_the_consultation)
(stop "End the consultation."))
(= *answ rerun_Psyxpert)
(putline "Rerun the consultation.")
(delete ((fact . *) . *))
(psyxpert))
(= *answ exit_the_interpreter)
(quit "Returning to the operating system."))
)

((log_off) if
(fact log no)
)

((log_off) if
(fact log yes)
(disconnect stdout)
(close fd1)
)

)

```

[illegible]

```

        (putline "WHY?")
        (putline "Just curious! ")
    )
    ((explain_why 2 *list) if
        (putline "WHY?")
        (putline
            "The information may be needed later to make a diagnosis.")
    )
    ((explain_why 90 *list) if
        (putline "WHY?")
        (putline
            "No explanation needed.")
    )
    ((explain_why 95 *list) if
        (putline "WHY?")
        (putline "No explanation needed.")
    )
    ((explain_why 96 *list) if
        (putline "WHY?")
        (putline "No explanation needed.")
    )
    ((explain_why 99 *list) if
        (putline "WHY?")
        (putline "No explanation needed.")
    )
    ((explain_why *qnum *list) if
        (find_list *list *out1 DISORDER *rest1)
        (≠ *out1 nil)
        (return_fact 1 *out1 *disorder)
        (putline "WHY?")
        (println "Because the possibility of " *disorder)
        (putline
            "is under consideration.")
        (or((find_list *list *out3 DETERMINE *rest3)
            (≠ *out3 nil)
            (return_fact 1 *out3 *f)
            (println "Trying to determine " *f))
            (≠ *qnum *qnum)
        )
        (putline
            "The facts are :")
        (print_facts *list)
        (putline
            "-----")
    )
    ((explain_why *qnum *list) if
        (find_list *list *out CATEGORY *rest)
        (≠ *out nil)
        (return_fact 1 *out *category)
        (putline "WHY?")
        (println "Because the possibility of " *category)
        (putline
            "is under consideration.")
        (or((find_list *list *out3 DETERMINE *rest3)
            (≠ *out3 nil)
            (return_fact 1 *out3 *f)

```

```

        (println "Trying to determine " *f))
    )
    (== *qnum *qnum)
    (putline
"The facts are :")
    (print_facts *list)
    (putline
"-----")
    )
    ((explain_why *qnum *list) if
      (putline "WHY?")
      (putline
"The information is needed for part of the mental
status examination.")
    )

    ((explain_how *qnum *list) if
      (how_file *qnum *statement)
      (putline "HOW?")
      (putline *statement)
    )

    ((explain_how *qnum *list) if
      (putline "HOW?")
      (putline
"Not implemented for this question.")
    )

    ((explain_diagnosis *struct) if
      (putline
"-----
DIAGNOSTIC EXPLANATION")
      (print_diagnosis *struct)
      (putline
"Based upon the following facts :
")
      (print_facts *struct)
      (putline
"-----")
    )

    )

    ((quest 1000
      "Please enter the patient's full name enclosed in parentheses."
      nil nil))

    ((quest 2000
      "Please enter the patient's age in years."
      nil nil))

    ((quest 22000
      "The patient's mood is the dominant emotion for the duration
of the interview."
      nil nil))

```

```
((quest 26000
"Affect refers to an immediately expressed and observed emotion.
Normally affect changes repeatedly through an interview, always
in synchrony with what the person is currently saying.
Range refers to the extent to which both emotional highs and lows
appear in the interview."
nil nil))
```

```
((quest 27000
"Intensity of affect is the amplitude of emotional expression."
nil nil))
```

```
((quest 28000
"Lability is evidenced by a rapid, extreme and brief swing of
emotion followed by a return to the previous level."
nil nil))
```

```
((quest 29000
"Appropriateness refers to whether the emotion at a particular
moment is the one expected for the patient's currently-expressed
thought."
nil nil))
```

```
((quest 30000
"The sensorium describes the condition of a subject relative to
the subject's mental clarity and is evaluated in terms of
orientation, memory and calculating ability."
nil nil))
```

```
((quest 49000
"Judgement involves the ability to make appropriate decisions
concerning what to do in occupational, economic and interpersonal
situations."
nil nil))
```

```
((quest 50000
"Abstracting ability is the ability to think in generalizations."
nil nil))
```

```
((how_file 30
"Orientation may be evaluated by asking the subject the month
date, year and place. Memory may be tested by asking the
subject to recall recent events. Calculating ability may be
tested by asking the subject to perform serial seven subtractions
(the subject must have had the skill previously)."))
```

```
((how_file 49
"The Best tests of judgement are customized to the patient's
life experiences. A housewife, for instance, might be asked
how she would spend $40 to buy groceries for her family to
last a week if there were no food in the house."))
```

```
((how_file 50
"Abstracting ability may be tested by asking the patient to
identify the way in which objects are similar, e.g., an apple
and an orange or a table and a chair."))
```

```
)
```



MAJOR\_AFFECT.P

PSYXPERT

Mary Overby  
April 24, 1986

This module contains the rules which determine if the patient has a major\_affective disorder.

Rules:

major\_affective

```
(assert
((major_affective *in *out) if
  (append *in ((DISORDER MAJOR AFFECTIVE DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  )
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
  (mood_disturbance *c *d secondary)
  (append *d ((DETERMINE IF_ONSET_MOOD_DISTURBANCE_PRIOR_TO_ONSET_PSYCH
    (onset psychosis *weeks_ago1 *e)
    (onset mood *weeks_ago2 *e)
    (>= *weeks_ago2 *weeks_ago1)
  )
  (append *d ((FACT ONSET_MOOD_DISTURBANCE_PRIOR_TO_ONSET_PSYCHOSIS 2))
    (question 79 *answ *f)
    (fact psychotic only with affective yes)
  )
  (append *f ((FACT PSYCHOSIS_PROMINENT_ONLY_WITH_MOOD_DISTURBANCE 2))
    (append *g ((CF2 2)) *out)
  )
)

((major_affective *in *out) if
  (append *in ((DISORDER MAJOR AFFECTIVE DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  )
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
  (mood_disturbance *c *d secondary)
  (append *d ((DETERMINE IF_DURATION_MOOD_DISTURBANCE_GREATER_THAN_DURA
    (duration mood *time2 *e)
    (> *time2 *time)
  )
  (append *d ((FACT DURATION_MOOD_DISTURBANCE_GREATER_THAN_DURATION_PSY
    (question 79 *answ *f)
    (fact psychotic only with affective yes)
  )
  (append *f ((FACT PSYCHOSIS_PROMINENT_ONLY_WITH_MOOD_DISTURBANCE 2))
    (append *g ((CF2 2)) *out)
  )
)
```

```
MOOD.P
;
;
; PSYXPERT
;
; Mary Overby
; April 2, 1986
;
; This module contains the rules which determine if the patient
; has a mood disorder.
;
; Rules:
;
;         mood_disturbance
;         mood_disorder
;         diagnose_mood
;
;
;
(assert
((mood_disturbance *in *out primary) if
  (append *in ((CATEGORY MOOD_DISTURBANCE)) *a)
  (mood_disorder *list *cm)
  (append *a *list *b)
  (>= *cm 3)
  (append *b ((CF 2)) *out)
)
((mood_disturbance *in *out secondary) if
  (append *in ((FACT MOOD_DISTURBANCE 2)) *a)
  (mood_disorder *list *cm)
  (append *a *list *out)
  (>= *cm 3)
)
((mood_disorder *list *cm) if
  (or((fact mood severely_depressed)
    (: = *cm 5)
    (== *list ((FACT SEVERELY_DEPRESSED 5)) ))
    ((fact mood moderately_depressed)
    (: = *cm 3)
    (== *list ((FACT MODERATELY_DEPRESSED 3)) ))
    ((fact mood moderately_elated)
    (: = *cm 3)
    (== *list ((FACT MODERATELY_ELATED 3)) ))
    ((fact mood severely_elated)
    (: = *cm 5)
    (== *list ((FACT SEVERELY_ELATED 5)) ))
  )
)
((diagnose_mood *in *out) if
  (append *in ((DIAGNOSIS NOT_DIFFERENTIATED)) *out)
)
```

```
; ORGANIC.P  
; ; ; ; ;  
;  
PSYXPERT  
  
Mary Overby  
April 2, 1986  
  
This module contains the rules which determine if the patient  
has an organic brain disorder.  
  
Rules:  
  
    organic  
        diagnose_organic  
; ; ; ; ;
```

```
(assert  
( (organic #in #out) if  
      (append #in ((CATEGORY ORGAINC_BRAIN_DISORDER)) *a)  
      (question 3 #answ1 *a)  
      (or ((append #a ((FACT ORGAINC_HISTORY 2)) *b)  
            (fact history_organic yes)  
            (: = *m1 4))  
          ((append #a ((FACT NO_ORGAINC_HISTORY 0)) *b)  
            (fact history_organic no)  
            (: = *m1 0))  
          ((append #a ((FACT UNSURE_ORGAINC_HISTORY 0)) *b)  
            (fact history_organic unsure)  
            (: = *m1 1)))  
      )  
      (question 6 #answ2 *b)  
      (or ((append #b ((FACT ORGAINC_PHYSICAL 2)) *c)  
            (fact physical_organic yes)  
            (: = *m2 (+ *m1 4)))  
          ((append #b ((FACT NO_ORGAINC_PHYSICAL 0)) *c)  
            (fact physical_organic no)  
            (: = *m2 *m1))  
          ((append #b ((FACT UNSURE_ORGAINC_PHYSICAL 0)) *c)  
            (fact physical_organic unsure)  
            (: = *m2 (+ *m1 1))))  
      )  
      (question 7 #answ3 *c)  
      (or ((append #c ((FACT ORGAINC_LAB 2)) *d)  
            (fact lab_test_organic yes)  
            (: = *m3 (+ *m2 4)))  
          ((append #c ((FACT NO_ORGAINC_LAB 0)) *d)  
            (fact lab_test_organic no)  
            (: = *m3 *m2))  
          ((append #c ((FACT UNSURE_ORGAINC_LAB 0)) *d)  
            (fact lab_test_organic unsure)  
            (: = *m3 (+ *m2 1))))  
      )  
      (or((>= *m3 4)
```

```

        (:= *cf 2))
      ((>= *m3 1)
       (:= *cf 1))
    )
    (append *d ((CF *cf)) *out)
  )
  ((diagnose_organic *in *out) if
   (append *in ((DIAGNOSIS NOT_DIFFERENTIATED)) *out)
  )
)

```

PARANOID.P

PSYXPERT

Mary Overby  
April 24, 1986

This module contains the rules which determine if the patient has a paranoid disorder.

Rules:

paranoid

```
(assert
((paranoid *in *out) if
  (append *in ((DISORDER PARANOID DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (!(mood_disturbance *c *d secondary))
    (psychological_disorder *c *list *cm1)
    (== *list nil)
    (append *c ((FACT NO_PSYCHOLOGICAL_DISTURBANCE 2)) *e)
    (fact jealous_delusions yes)
    (append *e ((FACT JEALOUS_DELUSIONS 2)) *f)
    (append *f ((CF2 2)) *out)
)

((paranoid *in *out) if
  (append *in ((DISORDER PARANOID DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (!(mood_disturbance *c *d secondary))
    (psychological_disorder *c *list *cm1)
    (== *list nil)
    (append *c ((FACT NO_PSYCHOLOGICAL_DISTURBANCE 2)) *e)
    (fact persecutory_delusions yes)
    (append *e ((FACT PERSECUTORY_DELUSIONS 2)) *f)
    (append *f ((CF2 2)) *out)
)

((paranoid *in *out) if
  (append *in ((DISORDER PARANOID DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (mood_disturbance *c *d secondary)
    (append *d ((DETERMINE IF_ONSET_PSYCHOSIS_PRIOR_TO_ONSET_MOOD_DISTURBANCE
```

PHRENIFORM. P

PSYXPERT

Mary Overby  
April 24, 1986

This module contains the rules which determine if the patient has a schizophreniform disorder.

Rules:

schizophreniform

```
(assert
((schizophreniform *in *out) if
  (append *in ((DISORDER SCHIZOPHRENIFORM DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
    (duration psychosis *time *b)
    (>= *time 2)
  (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (! (mood_disturbance *c *d secondary))
    (psychological_disorder *c *list *cm1)
    (!= *list nil)
    (append *c ((FACT PSYCHOLOGICAL_DISTURBANCE 2)) *e)
    (append *e *list *f)
    (:= *cf1 (* *cm1 2))
    (question 76 *answ *f)
    (or ((fact functioning_impaired yes)
      (append *f ((FACT FUNCTIONING_IMPAIRED 2)) *g)
      (:= *cf2 (+ *cf1 (* 2 2)))
      ((fact functioning_impaired unsure)
        (append *f ((FACT FUNCTIONING_IMPAIRED_UNSURE 2)) *g)
        (:= *cf2 (+ *cf1 (* 1 2)))
      )
    (append *g ((DETERMINE IF_ONSET_PRIOR_TO_AGE_45 2)) *h)
    (onset_prior *h)
    (append *g ((FACT ONSET_PRIOR_TO_AGE_45 2)) *i)
    (:= *cf3 (+ *cf2 (* 2 2)))
    (append *i ((DETERMINE IF_DURATION_LESS_THAN_6_MONTHS 2)) *j)
    (< *time 26)
    (append *i ((FACT DURATION_LESS_THAN_6_MONTHS 2)) *k)
    (:= *cf4 (+ *cf3 (* 2 2)))
    (or ((>= *cf4 14)
      (append *k ((CF2 2)) *out))
    (:= *cf4 12)
      (append *k ((CF2 1)) *out))
    )
  )
)

((schizophreniform *in *out) if
  (append *in ((DISORDER SCHIZOPHRENIFORM DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS)
```

```
; PSYCHOTIC.P
;
;
;
; PSYXPERT
;
; Mary Overby
; April 3, 1986
;
; This module contains the rules which determine if the patient
; has a psychotic disorder.
;
; Rules:
;
;     psychotic
;     diagnose_psychosis
;
;
;
(assert
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC_DISORDER)) *a)
  (append *a ((FACT DELUSIONS 2)) *b)
  (delusions *any 2)
    ; NOTE: importance measure may be added here
    ;       (: = *cf (* *cm *im))
  (append *b ((CF 2)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC_DISORDER)) *a)
  (append *a ((FACT HALLUCINATIONS_2)) *b)
  (hallucinations *any 2)
  (append *b ((CF 2)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC_DISORDER)) *a)
  (append *a ((FACT INCOHERENT 2)) *b)
  (fact thought_coherent no)
    ; NOTE: importance measure may be added here
    ;       (: = *cf (* *cm *im))
  (append *b ((CF 2)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC_DISORDER)) *a)
  (append *a ((FACT ILLOGICAL 2)) *b)
  (fact thought_logical no)
    ; NOTE: importance measure may be added here
    ;       (: = *cf (* *cm *im))
  (append *b ((CF 2)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC_DISORDER)) *a)
  (append *a ((FACT LOOSE_ASSOCIATIONS 2)) *b)
```

```

)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC DISORDER)) *a)
  (append *a ((FACT INCOHERENT_UNSURE 2)) *b)
  (fact thought_coherent_unsure)
    ; NOTE: importance measure may be added here
    ; (= *cf (* *cm *im))
  (append *b ((CF 1)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC DISORDER)) *a)
  (append *a ((FACT ILLOGICAL_UNSURE 2)) *b)
  (fact thought_logical_unsure)
    ; NOTE: importance measure may be added here
    ; (= *cf (* *cm *im))
  (append *b ((CF 1)) *out)
)
((psychotic *in *out) if
  (append *in ((CATEGORY PSYCHOTIC DISORDER)) *a)
  (append *a ((FACT POVERTY_CONTENT_THOUGHT_UNSURE 2)) *b)
  (fact impoverished_thought_unsure)
    ; NOTE: importance measure may be added here
    ; (= *cf (* *cm *im))
  (append *b ((CF 1)) *out)
)
((diagnose_psychosis *in *out) if
  (brief_reactive_psychosis *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)
((diagnose_psychosis *in *out) if
  (major_affective *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)
((diagnose_psychosis *in *out) if
  (schizo_affective *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)
((diagnose_psychosis *in *out) if
  (paranoid *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)
((diagnose_psychosis *in *out) if
  (schizophreniform *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)
((diagnose_psychosis *in *out) if
  (schizophrenia *in *a)
  (append *a ((DIAGNOSIS COMPLETE)) *out)
)

```



```

;      QUESTFILE.P
;
;      PSYXPERT
;
;      Mary Overby
;      April 2, 1986
;
;      This module contains the questions that are asked of the user.
;
;
;
(assert

((quest 1
"Please enter the patient's name."
nil (name) ))

((quest 2 "Please enter the patient's age." nil (age) ))

((quest 3
"Is there evidence from the patient's history of a specific
organic factor that is judged to be etiologically related to
the disturbance?"
(yes no unsure) (history_organic) ))

((quest 4
"Does the patient have a history of a mood disorder?"
(yes no unsure) (mood_history) ))

((quest 5
"Does the patient have a history of a psychotic disorder?"
(yes no unsure) (psychotic_history) ))

((quest 6
"Is there evidence from the physical examination of a specific
organic factor that is judged to be etiologically related to
the disturbance?"
(yes no unsure) (physical_organic) ))

((quest 7
"Is there evidence from laboratory tests of a specific organic
factor that is judged to be etiologically related to the
disturbance?"
(yes no unsure) (lab_test_organic) ))

```

```
REACTIVE.P
```

```
PSYXPERT
```

```
Mary Overby  
April 24, 1986
```

```
This module contains the rules which determine if the patient  
has a brief reactive psychotic disorder.
```

```
Rules:
```

```
brief_reactive_psychosis
```

```
(assert  
( (brief_reactive_psychosis *in *out) if  
  (append *in ((DISORDER BRIEF REACTIVE PSYCHOSIS)) *a)  
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_LESS_THAN_2_WEEKS)) *  
    (duration psychosis *time *b)  
    (< *time 2)  
  (append *a ((FACT DURATION_PSYCHOSIS_LESS_THAN_2_WEEKS 2)) *c)  
    (question 78 *answ *c)  
    (fact psychosis_follow_event yes)  
    (append *c ((FACT PSYCHOSIS_FOLLOW_UPSETTING_ENVIRONMENTAL_EVENT 2))  
    (append *d ((CF2 2)) *out)  
)  
)
```

```

;      RULES.P
;
;      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;      PSYXPERT
;
;      Mary Overby
;      April 3, 1986
;
;      This module contains the rules which determine if the patient
;      has various conditions.
;
;      Rules:
;
;          hallucinations
;          delusions
;
;      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
(assert
((hallucinations auditory 2) if
    (fact auditory_hallucinations yes))
((hallucinations visual 2) if
    (fact visual_hallucinations yes))
((hallucinations sensory 2) if
    (fact sensory_hallucinations yes))

((hallucinations auditory 1) if
    (fact auditory_hallucinations unsure))
((hallucinations visual 1) if
    (fact visual_hallucinations unsure))
((hallucinations sensory 1) if
    (fact sensory_hallucinations unsure))

((delusions persecutory 2) if
    (fact persecutory_delusions yes))
((delusions grandiose 2) if
    (fact grandiose_delusions yes))
((delusions somatic 2) if
    (fact somatic_delusions yes))
((delusions jealous 2) if
    (fact jealous_delusions yes))

((delusions persecutory 1) if
    (fact persecutory_delusions unsure))
((delusions grandiose 1) if
    (fact grandiose_delusions unsure))
((delusions somatic 1) if
    (fact somatic_delusions unsure))
((delusions jealous 1) if
    (fact jealous_delusions unsure))

((onset psychosis *weeks_ago *in) if
    (question 70 *weeks_ago *in))

((onset mood *weeks_ago *in) if

```

```

      (question 71 *weeks_ago *in))
((duration psychosis *time *in) if
  (onset psychosis *weeks_ago *in)
  (question 72 *answ *in)
  (or((fact psychotic_symptoms present)
      (:= *time *weeks_ago)
      (:= *cm 2))
    ((fact psychotic_symptoms unsure)
      (:= *time *weeks_ago)
      (:= *cm 1))
    ((fact psychotic_symptoms not_present)
      (question 74 *answ1 *in)
      (:= *time (- *weeks_ago *answ1))
      (:= *cm 0))
  )
)
((duration mood *time *in) if
  (onset mood *weeks_ago *in)
  (question 73 *answ *in)
  (or((fact mood_symptoms present)
      (:= *time *weeks_ago)
      (:= *cm 2))
    ((fact mood_symptoms unsure)
      (:= *time *weeks_ago)
      (:= *cm 1))
    ((fact mood_symptoms not_present)
      (question 75 *answ1 *in)
      (:= *time (- *weeks_ago *answ1))
      (:= *cm 0))
  )
)
)
)

```

```

        (append *i ((FACT FUNCTIONING_IMPAIRED_UNSURE 2)) *j)
        (:= *cf2 (+ *cf1 (* 1 2))) )
    )
    (append *j ((DETERMINE IF_ONSET_PRIOR_TO_AGE_45)) *k)
    (onset_prior *k)
    (append *j ((FACT ONSET_PRIOR_TO_AGE_45 2)) *l)
    (:= *cf3 (+ *cf2 (* 2 2)))
    (append *l ((DETERMINE IF_DURATION_GREATER_THAN_6_MONTHS)) *m)
    (>= *time 26)
    (append *l ((FACT DURATION_GREATER_THAN_6_MONTHS 2)) *n)
    (:= *cf4 (+ *cf3 (* 2 2)))
    (or (>= *cf4 14)
        (append *n ((CF2 2)) *out))
    ((= *cf4 12)
        (append *n ((CF2 1)) *out))
    )
)

((psychological_disorder *in *out *cm) if
  (append *in ((DETERMINE PSYCHOLOGICAL_DISTURBANCE 2)) *a)
  (psychological_disturbance *a *out *cm)
  (cut)
)

((psychological_disturbance *in *out 2) if
  (delusions *any 2)
  (question 77 *answ *in)
  (fact bizarre_delusions yes)
  (== *out ((FACT BIZARRE_DELUSIONS 2)) )
)

((psychological_disturbance *in *out 2) if
  (fact auditory_hallucinations yes)
  (== *out ((FACT AUDITORY_HALLUCINATIONS 2)) )
)

((psychological_disturbance *in *out 2) if
  (fact associations loose)
  (== *out ((FACT LOOSE_ASSOCIATIONS 2)) )
)

((psychological_disturbance *in *out 2) if
  (fact thought_coherent no)
  (== *out ((FACT THOUGHT_INCOHERENT 2)) )
)

((psychological_disturbance *in *out 2) if
  (fact range_of_affect markedly_decreased)
  (fact intensity_of_affect markedly_decreased)
  (== *out ((FACT FLAT_AFFECT 2)) )
)

; ((psychological_disturbance *in *out 1) if
;   (delusions *any 2)
;   (question 77 *answ *in)
;   (fact bizarre_delusions unsure)
;   (== *out ((FACT MAYBE_BIZARRE_DELUSIONS 2)) )
; )

((psychological_disturbance *in *out 1) if
  (delusions *any 1)
  (question 77 *answ *in)
  (fact bizarre_delusions unsure)
  (== *out ((FACT MAYBE_BIZARRE_DELUSIONS 2)) )
)

```

```

; SCH_AFFECT.P

```

```

; PSYXPERT

```

```

; Mary Overby
; April 24, 1986

```

```

; This module contains the rules which determine if the patient
; has a schizo_affective disorder.

```

```

; Rules:

```

```

; schizo_affective

```

```

; (assert

```

```

((schizo_affective *in *out) if
  (append *in ((DISORDER SCHIZO AFFECTIVE DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS
    (duration psychosis *time *b)
    (>= *time 2)
    (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (mood_disturbance *c *d secondary)
    (append *d ((DETERMINE IF_ONSET_MOOD_DISTURBANCE_PRIOR_TO_ONSET_PSYC
      (onset psychosis *weeks_ago1 *e)
      (onset mood *weeks_ago2 *e)
      (>= *weeks_ago2 *weeks_ago1)
      (append *d ((FACT ONSET_MOOD_DISTURBANCE_PRIOR_TO_ONSET_PSYCHOSIS 2)
        (question 79 *answ *f)
        (or((fact psychotic_only_with_affective no)
          (:= *cf 2)
          (append *f ((FACT PSYCHOSIS_NOT_PROMINENT_ONLY_WITH_MOOD_DISTURB
            ((fact psychotic_only_with_affective unsure)
            (:= *cf 1)
            (append *f ((FACT PSYCHOSIS_MAY_NOT_BE_PROMINENT_ONLY_WITH_MOOD_
              )
              (append *g ((CF2 *cf)) *out)
            )
          )
        )
      )
    )
  )

```

```

((schizo_affective *in *out) if
  (append *in ((DISORDER SCHIZO AFFECTIVE DISORDER)) *a)
  (append *a ((DETERMINE IF_DURATION_OF_PSYCHOSIS_GREATER_THAN_2_WEEKS
    (duration psychosis *time *b)
    (>= *time 2)
    (append *a ((FACT DURATION_PSYCHOSIS_GREATER_THAN_2_WEEKS 2)) *c)
    (mood_disturbance *c *d secondary)
    (append *d ((DETERMINE IF_DURATION_MOOD_DISTURBANCE_GREATER_THAN_DUR
      (duration mood *time2 *e)
      (> *time2 *time)
      (append *d ((FACT DURATION_MOOD_DISTURBANCE_GREATER_THAN_DURATION_PS
        (question 79 *answ *f)
        (or((fact psychotic_only_with_affective no)
          (:= *cf 2)
          (append *f ((FACT PSYCHOSIS_NOT_PROMINENT_ONLY_WITH_MOOD_DISTURB
            )
          )
        )
      )
    )
  )

```

[illegible]

```

        )
        (assert_answ *answ *assertion)
        (cut)
    )

    ((print_menu nil *))
    ((print_menu (*head . *tail) *num) if
      (write *num)
      (write " ")
      (putline *head)
      (:= *newnum (+ *num 1))
      (print_menu *tail *newnum)
    )

    ((assert_answ *answ nil))
    ((assert_answ *answ (*head . *tail)) if
      (or (fact *head *answ)
        (assert
          ((fact *head *any) if (== *any *answ))
        )
      )
    )

    ((explain_answ *qnum *in *out *struct) if
      (or (== *in ?)
        (== *in why)
        (== *in how))
      (explain *qnum *in *struct)
      (question *qnum *z *struct)
      (== *out *z)
    )
    ((explain_answ *qnum *in *in *struct))

    ((interp_answ *assertion *new *new nil))
    ((interp_answ *assertion *a1 *a2 *1) if
      (numberp *a1)
      (or ((== *a1 0)
        (== *a2 nil))
        (:= *a2 (nth '*1 *a1))
      )
    )

    ((interp_answ (*head . *tail) *a1 *a2 *1) if
      (fact *head *a1)
      (== *a2 *a1)
    )

    ((interp_answ *assertion *a1 *a2 *1) if
      (== *a2 nil)
    )

    ((enter_amsit *struct) if
      (putline

```



"Please answer the following questions based upon your mental status examination findings.

```
" )
      (call_quest 22 22 *struct)
      (call_quest 26 51 *struct)
      (cut)
)

((call_quest *start *end *struct) if
 (> *start *end)
)

((call_quest *qnum *end *struct) if
 (question *qnum *answ *struct)
 (:= *num (+ *qnum 1))
 (call_quest *num *end *struct)
)

)
```