This blank page was inserted to preserve pagination.
DIAGNOSIS

by

GEORGE ANTHONY GORRY

B. Engineering, Yale University, 1962

M. S., University of California (Berkeley), 1963

Submitted in Partial Fulfillment
of the Requirements for the Degree
of Doctor of Philosophy in
Computer Science

June 1967

Signature of Author

Alfred P. Sloan School of Management, May 12, 1967

Certified by

Thesis Supervisor

Accepted by

Chairman, Departmental Committee on Graduate Students

"Work reported herein was supported (in part) by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nour-4102(O1). Reproduction in whole or in part is permitted for any purpose of the United States Government."
A SYSTEM FOR COMPUTER-AIDED
DIAGNOSIS

by

GEORGE ANTHONY GORRY

Submitted to the Alfred P. Sloan School of Management on
May 12, 1967 in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science.

ABSTRACT

This thesis describes a model diagnostic problem and a computer
program designed to deal with this problem. The model diagnostic
problem is an abstract problem. A major contention of this thesis,
however, is that this problem subsumes the principal features of a
number of ostensibly different real diagnostic problems including
certain problems of medical diagnosis and the diagnosis of machine
failures. A second major contention of this thesis is that strate-
gies for the solution of the model diagnostic problem can be formu-
lated in terms sufficiently explicit to permit their incorporation
in a computer program.

The model diagnostic problem assumes that the system being di-
gnosed (e.g., a person, or machine) is in one of a finite number of
known states. Tests can be performed at some cost to discover attrib-
utes of the system, for example signs or symptoms in medical diagno-
sis. The current state of the system is to be deduced from the ob-
served attributes and past experience with similar systems. In the
model, this experience is represented principally in terms of proba-
bilities (e.g. the conditional probability of a certain attribute
given the system state).

The statement of the model diagnostic problem requires that the
diagnostician also account for the cost of various misdiagnoses. In
particular for each pair of states i and j, the cost of misdiagnosing
state j as state i, \( c_{ij} \), is given. Thus the diagnostician must bal-
ance the cost of performing additional tests against the expected
reduction in the cost of misdiagnosis. This requirement suggests the
value of sequential diagnosis.
A computer program was developed to solve the model diagnostic problem. It consists of 1) an inference function which is based on a Bayesian analysis of attributes and includes a flexible way of dealing with non-independent attributes, 2) a pattern-sorting function which allows the program to detect irrelevant attributes and patterns of attributes corresponding to two different system states, and 3) a test selection function which employs various heuristics to select good tests for the user of the program to perform on the system under consideration. The diagnostic program is specialized for a particular problem by providing it with the appropriate experience. The program is embedded in an environment (set of programs) which facilitates the study of various diagnostic strategies.

The diagnostic program was implemented on the time-sharing system at Project MAC. It was applied to two medical problems, the diagnosis of congenital heart disease, and the diagnosis of primary bone tumors. The results obtained here suggest 1) that a computer program can be of considerable value as a diagnostic tool, and 2) that it is quite advantageous for such a program to perform sequential diagnosis as it interacts with the user.

Thesis Supervisor: Joseph Weizenbaum
Title: Associate Professor of Electrical Engineering and Political Science
Acknowledgments

I would like to express my gratitude to my thesis advisor, Professor Joseph Weizenbaum, for his advice and encouragement. Also I would like to thank the members of my committee, Professors Donald Carroll and Murray Eden, for their illuminating criticism of my work. Special thanks are due to Dr. G. Octo Barnett of the Massachusetts General Hospital. Dr. Barnett obtained for me the data used in the research. He also provided insights into the nature of some fundamental problems of medical diagnosis. In total his assistance was invaluable.

I would like to acknowledge the assistance of Charles Coleman of the Boston Programming Center of IBM who provided me with valuable facilities for a major part of my research.

Mrs. Jean Stanton typed the drafts and final copy of the thesis, and I owe her particular thanks for her work.

Finally, my thanks to Lucinda who, for a year, had to share her husband with a machine.

G. A. G.
Brookline, Massachusetts
May, 1967
# TABLE OF CONTENTS

Abstract 11

Acknowledgments iv

Chapter 1  Introduction 1
  Diagnostic Problems and Processes 2
  A Brief Outline of a Diagnostic Process 5
  Some Further Comments on the Difficulties of Diagnosis 6
  A Preface to the Material Which Follows 9

Chapter 2  Literature Survey 12
  Diagnostic Programs 12
  Perspectives on Diagnosis 19

Chapter 3  Two Views of Diagnosis 22
  Diagnosis as a Problem in Pattern Recognition 23
  Diagnosis as a Sequential Decision Problem 33
  Heuristic Considerations in Test Selection 42

Chapter 4  A Diagnostic System 52
  An Information Structure for the Diagnostic System 55
  The Diagnostic Program 69
  The Pattern Sorting Function 74
  The Inference Function 88
  The Test Selection Function 96
  The Generator Program 103

Chapter 5  Diagnosis of Primary Bone Tumors 112
  Experiments in Bone Tumor Diagnosis 114

Chapter 6  Diagnosis of Congenital Heart Disease 127
  Experiments in Congenital Heart Disease Diagnosis 128

Chapter 7  Further Experiments with the Diagnostic System 137
  The Effect of a Very Serious State 137
  Studies of a Test Selection Heuristic 144
  The Pattern-Sorting Capability 157

Chapter 8  Discussion of the Research 162
  Some Comments on the Diagnostic Model 177
References 183
Appendix 1  Sample of an Input File 185
Appendix 2  Trace of a Session with the Diagnostic Program 187
Appendix 3  Listings of Diagnostic System 191
Biographical Note 244
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Two Pattern Classes</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>Intersecting Pattern Classes</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>Transformation of Pattern Classes</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Section of a Decision Tree</td>
<td>38</td>
</tr>
<tr>
<td>5</td>
<td>A Simple SLIP List</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>A Sample State List</td>
<td>59</td>
</tr>
<tr>
<td>7</td>
<td>A Sample Attribute List</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>A Sample Test List</td>
<td>62</td>
</tr>
<tr>
<td>9</td>
<td>A Portion of an Information Structure</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>State List With Cluster</td>
<td>68</td>
</tr>
<tr>
<td>11</td>
<td>Flow of Diagnostic Program</td>
<td>73</td>
</tr>
<tr>
<td>12</td>
<td>Pattern Stack</td>
<td>78</td>
</tr>
<tr>
<td>13</td>
<td>Effect of New Attribute on Pattern Stack</td>
<td>81</td>
</tr>
<tr>
<td>14</td>
<td>Schematic of Diagnostic System</td>
<td>109</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Growth of Search with Depth</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>Example for Bayesian Analysis</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>Disease Description for Generator Example</td>
<td>107</td>
</tr>
<tr>
<td>4</td>
<td>Histological Types for Bone Tumor Diagnosis</td>
<td>115</td>
</tr>
<tr>
<td>5</td>
<td>Attributes for Bone Tumor Diagnosis</td>
<td>116</td>
</tr>
<tr>
<td>6</td>
<td>Tests for Bone Tumor Diagnosis</td>
<td>117</td>
</tr>
<tr>
<td>7</td>
<td>Diagnoses Based on all Available Signs for Case Histories</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>Sequential Diagnosis--An Example</td>
<td>121</td>
</tr>
<tr>
<td>9</td>
<td>Sequential Diagnosis of Bone Tumor Cases</td>
<td>122</td>
</tr>
<tr>
<td>10</td>
<td>Sequential Diagnosis of Simulated Case Histories</td>
<td>126</td>
</tr>
<tr>
<td>11</td>
<td>Heart Disease Types</td>
<td>129</td>
</tr>
<tr>
<td>12</td>
<td>Attributes for Congenital Heart Disease</td>
<td>130</td>
</tr>
<tr>
<td>13</td>
<td>Tests for Congenital Heart Disease</td>
<td>131</td>
</tr>
<tr>
<td>14</td>
<td>Diagnoses Based on All Available Attributes</td>
<td>132</td>
</tr>
<tr>
<td>15</td>
<td>Sequential Diagnosis of Actual Heart Disease Cases</td>
<td>134</td>
</tr>
<tr>
<td>16</td>
<td>Loss Function Matrix for Bone Tumor Diagnosis</td>
<td>139</td>
</tr>
<tr>
<td>17</td>
<td>Sequential Diagnosis of Cases for Loss Function of Table 17</td>
<td>140</td>
</tr>
<tr>
<td>18</td>
<td>Loss Function Matrix for Bone Tumor Diagnosis</td>
<td>142</td>
</tr>
<tr>
<td>19</td>
<td>Sequential Diagnoses of Cases for Loss Function of Table 18</td>
<td>143</td>
</tr>
<tr>
<td>Page</td>
<td>Title</td>
<td>Start Page</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>20</td>
<td>Sequential Diagnosis of Heart Disease Cases</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>--Standard Test Selection Function</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Sequential Diagnosis of Heart Disease Cases</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>--Dominated Test Heuristic</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Sequential Diagnosis of Bone Tumor Cases</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td>--Standard Test Selection Function</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Sequential Diagnosis of Bone Tumor Cases</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>--Dominated Test Heuristic</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Artificial Structure</td>
<td>159</td>
</tr>
<tr>
<td>25</td>
<td>Loss Function for Six State Problem</td>
<td>160</td>
</tr>
</tbody>
</table>
This empty page was substituted for a blank page in the original document.
Chapter 1

DIAGNOSTIC PROBLEMS AND PROCESSES

There are many problem areas in which attention is focused on some system. In these areas, the principal problem is to ascertain the current state of the given system. In general terms such a problem is a diagnostic problem. The problem-solver or diagnostician is equipped for his task with information distilled from past experience with such systems, and he attempts to couple this general knowledge with specific observations or tests of the given system in such a way that he can deduce the identity of the current state. The extent of the general knowledge, its organization, and the particular manner in which it is brought to bear on the diagnostic problem, the diagnostic process, may vary considerably among different problem areas, but the general nature of the problem persists.

Thus the medical diagnostician deals with the problem of discovering the "state" of the patient. Through training and experience, the physician has learned the sign and symptom patterns associated with possible diseases from which the patient can suffer. One problem is the effective utilization of this experience which is framed in terms of the abstraction of the disease and the reality of the individual patient. An additional complication arises from the
fact that different diseases may result in similar signs and symptoms. The physician exploits his general knowledge or experience in the selection of a sequence of tests to apply to the patient. The results of these tests provide him with information from which he constructs a more complete picture of the health of the patient. These tests may include simple questions as in the history-taking or complicated medical procedures such as in an exploratory operation. Since tests may exact a high cost (in terms of risk to the patient, patient discomfort, the time of skilled persons, money, etc.), it is the additional task of the diagnostician to properly balance this cost against the potential usefulness of the test results. For these and other reasons, medical diagnosis is often a complex and difficult intellectual problem.

A second example of a diagnostic problem is that of debugging computer programs. A program containing one or more errors can be thought of as a system for which it is desired to determine the state. The state in this case is characterized by the particular combination of errors. The programmer brings his past experience with a variety of programs to bear on this diagnostic problem. By controlling the inputs to the program, applying traces, or altering instruction sequences, or employing a post mortem, he can perform a range of tests on the program. The results of these tests may suggest new tests as well as providing the programmer with new insight into the problem currently confronting him. Like medical diagnosis, program debugging
is often a difficult task, requiring considerable judgment both in the selection of tests and the interpretation of results.

The research reported here is concerned with a particular diagnostic problem and a diagnostic process for solving that problem. It has several aims. The first is to formulate the model of the diagnostic problem in such a way that the definition subsumes the principal features of problems in a number of ostensibly different problem areas. For example, the definition might apply both to medical diagnosis and to program debugging, although it might not be the particular definition employed by diagnosticians in the respective areas. That such a model can be formulated is the major contention of this thesis. The second aim is to develop and investigate strategies for the solution of this model diagnostic problem. Because they are to be stated in terms of an abstract problem, such strategies will be independent of any real diagnostic problem. These diagnostic procedures then are to be embodied in a computer program. This step serves two purposes. First, the program provides an explicit statement of the diagnostic strategies, and thus facilitates the testing of these strategies on particular problems. Second, if the strategies in the program prove effective in practical applications, the program could be of considerable value in computer-aided diagnosis. In the event that this approach were successful, the resulting program may be useful in a number of distinct diagnostic problems, since the methods it employed would be problem-independent. The second
major contention of this thesis is that given a model for the diagnostic problem, effective strategies for the solution of the problem can be formulated in terms appropriate for their implementation in a program.

Such a program for diagnosis could be embedded in an environment (other programs) which would permit two different uses of the program. First, the program could be applied to actual diagnostic problems so that its effectiveness could be determined. Second, the environment could permit the study of a variety of artificial problems, each designed to test a particular aspect of program performance. The first type of application might be termed "open diagnosis"; and the second, "closed diagnosis." Closed diagnosis may facilitate the development of improved diagnostic strategies.

In order for a diagnostic problem to exist, one must have at least some knowledge of the nature of the system being considered. Further the various states of the system must manifest themselves through certain observable attributes. It should also be possible to apply tests to the system at some cost to obtain more attributes. Finally, the general knowledge of the system must include some comprehension of the relationships among signs, states, and tests. The prerequisites are satisfied by the two examples of diagnostic problems presented above. In fact, in simplest terms, this is the basis

\[1\text{The term attribute is used in this thesis to denote any observable manifestation of system state which is employed in the deductive phase of diagnosis. For example, it includes both signs and symptoms in medicine.}\]
for the diagnostic problem studied in this work.

A Brief Outline of a Diagnostic Process

The basic outline of a diagnostic process is as follows. Because the observation of certain initial attributes suggest a diagnostic problem in some system, the diagnostician wishes to ascertain the current state of the system. He selects a test (based on some criterion) and applies it to the system. The application of the test yields to update his current view of the problem. He then applies another test and obtains more attributes. This process continues until the diagnostician makes a decision about the current state. Now this is a most sketchy outline of the diagnostic process. There can be a great deal of sophisticated information processing during each iteration of the process. The point is that test selection and inference are the two principal features of diagnosis as performed in a number of distinct areas. The outline above seems equally applicable to medical diagnosis, qualitative chemical analysis, and the problem of diagnosing a malfunctioning automobile. At this level, then, the diagnostic processes in these and other areas exhibit considerable similarity. Inference and test selection appear to be the keys to diagnostic strategies of some generality. If it could be demonstrated that these features of the process necessarily differ fundamentally from area to area, then there would be little hope for the formulation of general diagnostic strategies. In fact, as will be shown in this work, there is reason to believe quite the contrary.
It appears that, for a number of areas, problem-independent diagnostic strategies can be developed. Note that the strategies employed by experts in different fields may be quite dissimilar, there is no requirement that the strategies developed here resemble theirs. The criterion by which strategies will be judged is how effective they are in particular applications, not how closely they approximate those currently used by human experts.

The diagnostic process then merits careful study for several reasons. First, as indicated above, variations of this problem arise in many different contexts and so the problem is of general interest. Second, the nature of the diagnostic problem is such that it often requires a great deal of intellectual effort to solve it, and any means of improving the problem-solving process will be of considerable value. Finally, the general form of the problem suggests the value of a man-machine partnership in the problem-solving process. Before such a partnership can be established, however, the diagnostic process must be carefully explored in order to determine respective parts to be played by man and machine.

Some Further Comments on the Difficulties of Diagnosis

Diagnostic problems on the whole are difficult ones, particularly for non-experts. Moreover, a great many diagnostic problems constitute considerable challenges to the skill of even the most expert diagnostician. Several factors contribute to the complexity of the diagnostic problem. First, an expert diagnostician must be aware of
a large number of relationships among system states and attributes. As evidence of this, consider the considerable training required to develop the skills of a medical diagnostician. Observation of many different attributes may be required to identify a particular state, and a given attribute may suggest many possible states. These facts coupled with the often large number of states and attributes require the diagnostician to master considerable amounts of information.

Often the relationships mentioned above are known only in probabilistic terms. In such a case, the task of the diagnostician is complicated by the need for some form of probability analysis, a task which generally proves quite difficult for human beings. The accurate assessment of probabilities for a large number of possible states given observed attributes requires extensive training and experience.

Another factor complicating the task of the diagnostician is the difficulty of establishing and maintaining an appropriate structure for all the information relevant to the diagnostic area. Much of the usefulness of that information in the diagnostic process accrues from its organization. A major portion of the expert's skill is derived from his ability to associate particular attributes or attribute patterns with possible system states and subsequent testing strategies. Again extensive experience and training are required to organize the relevant information into a useful associative structure. Unfortunately such a structure is not easily maintained. Associations
which are seldom used may be effectively lost to the diagnostician. As a result, his field of competence tends to become narrow. This tendency is accelerated when the diagnostician must devote considerable effort to the mastering of a continual stream of newly-relevant information.

A computer program to provide general diagnostic assistance to its user would help circumvent some of these difficulties. One of the significant advantages to be gained from the use of a computer is the sheer bulk of information which it can maintain. A diagnostic program would be able to deal with extremely large information structures. Since the program would be independent of the content of the information structure which it employed, that content could be continually updated without affecting the operation of the program (although better information should result in better program performance).

The amount of logical and probabilistic inference with which the program could cope would exceed that comprehensible to a human being. This capability would permit the more extensive exploration of possible testing strategies. Because the program could consider more possible diagnoses than a human being, it would provide a strong safeguard that a particular state is not overlooked in the diagnosis. Finally, a diagnostic program which was "table-driven" would be of all the more value because of its potential applicability to a variety of problems.
Note that diagnostic strategies suited for a computer are not necessarily suited for a human diagnostician. While human diagnosticians possess many special skills and hence serve as good sources of information about diagnosis, the purpose of this research does not restrict the set of possible strategies to those employed by humans. The goal is to develop strategies which enable the peculiar capabilities of the computer to be exploited. Additional insight into the nature of the human diagnostic process and the discovery of ways to improve it would be a valuable, but derivative result of this research.

A Preface to the Material Which Follows

This thesis describes a computer program for diagnosis and presents the results of some experiments performed with this program. The design of the program was strongly influenced by the model diagnostic problem chosen for this research. Although later chapters contain detailed discussions of this problem, a brief summary of its principal characteristics is presented here to provide some perspective on the problem.

The statement of the diagnostic problem considered here assumes that the system is in one of a finite number of states. The object of the diagnosis is to identify the current state of the system. Experience with similar systems is assumed to be available. This experience is in the form of probabilities for the various states and
probabilities of attributes given state. Test costs are constant and known. Furthermore the application of a test does not change the state of the system. Tests are also assumed to be accurate. Finally, it is assumed that the decision loss for each possible misdiagnosis is given in the same units as test costs. This work, then, is concerned with the development of strategies to solve diagnostic problems which can be stated in keeping with these assumptions.

Chapter 2 examines some of the research reported in the literature which has direct relevance to this work.

Chapter 3 presents two views of a diagnostic problem. In the first view, diagnosis is considered as a problem in pattern recognition. The implications and limitations of this view are examined. Then the problem of diagnosis is formulated as a sequential decision problem. This formulation underscores the computational problems associated with the determination of optimal testing strategies. Finally, a discussion of heuristic considerations in test selection is presented.

A system for the study of computer-aided diagnosis is described in detail in Chapter 4. This system includes both a diagnostic program and a variety of programs which provide an environment within which different diagnostic strategies can be studied.

The next three chapters are devoted to experiments performed with the diagnostic system. Chapter 5 discusses the use of the sys-
tem in the diagnosis of primary bone tumors; and Chapter 6, an application of the system to the diagnosis of congenital heart disease.

A number of other experiments with the system are discussed in Chapter 7. Chapter 8 presents a discussion of the results of the research and delineates some areas for further investigation.
Chapter 2

LITERATURE SURVEY

A. Diagnostic Programs

In recent years, there has been an increasing amount of work done on various aspects of diagnosis. Some of this work has been aimed at the development of computer programs to perform particular diagnostic tasks. Other work has been more oriented toward the study of human diagnosticians and the strategies they employ. A brief survey of this work is presented in this chapter. Examples of computer programs for diagnosis are discussed. Of particular interest are the diagnostic strategies and models employed by such programs. Finally, some broad views of diagnosis and its attendant difficulties are considered.

By far the greatest concentration of research in computer-aided diagnosis has been focused in the area of medical diagnosis. A number of programs have been written which are capable of performing diagnosis in particular medical areas. These programs, as a rule employ a Bayesian analysis of attributes based on a disease-attribute probability matrix for the given set of diseases considered. That is the programs compute the probability of disease $D$ given the set of attributes $A$ as follows

$$P(D|A) = \frac{P(D) P(A|D)}{\sum_D P(D) P(A|D)}$$
where \( P(D) \) is the \textit{a priori} probability of \( D \).

\( P(A/D) \) is the conditional probability of \( A \) \textit{given} \( D \).

The use of a disease-attribute model and Bayesian inference was advocated by a number of researchers as early as 1959 (R1, R2, R3, R4,). While other means of inferring diseases from their attributes were suggested at this time (R5, R6), the Bayesian approach has proved the most widely used. In certain areas the use of analog computers has been explored, but this work will not be reviewed here.

In recent years, computer programs incorporating the Bayesian model have been developed for problems of heart disease (R7, R8), Thyroid disease (R9), epigastric pain (R10), Cushing's syndrome (R11) and others. Some of these programs have enjoyed striking success in attaining levels of performance comparable to that of the expert human diagnosticians. For example, a Bayesian analysis of 268 cases of patients with one of three thyroid problems yielded the accepted diagnosis in 96% of the cases. (R9). In a similar analysis of acquired valvular heart disease patients, a computer program correctly identified 96% of the problems. (R7). In both cases this level of performance compares favorably with that attained by experienced diagnosticians.

In order to provide a more detailed view of the use of Bayesian analysis in computer-aided diagnosis, two studies will be reviewed here. The first is the diagnosis of congenital heart disease; and the second, the diagnosis of thyroid function.

In a series of papers (R12, R13, R14), Warner, Toronto, and
Veasy have reported on the development and use of a computer program for the diagnosis of congenital heart disease. This program employs fifty-seven possible attributes to classify patients into thirty-five different disease classes. The basic strategy employed by the program is the use of Bayes' rule to obtain the posterior conditional probabilities for the different diseases given a particular set of attributes. The necessary a priori disease probabilities and conditional probabilities of attributes given disease were derived from statistical studies of a large number of known congenital heart disease patients. In certain instances, the statistical information so obtained was deemed inadequate and the probabilities involved were estimated from 1) the available literature and 2) consideration of the pathologic physiology of the disease. The program takes into account the significance of attributes which are absent as well as those which are present. Thus, the absence of cyanosis is significant in the diagnosis. The program is also designed to account for certain mutually exclusive sets of attributes. For instance, if one of a set of mutually exclusive attributes is present, it would be incorrect to consider the absence of the other attributes in the set as additional information in the diagnosis.

The program is used in the following way. For each patient examined, the examining physician determines the presence of absence of the required attributes. When the examination has been completed, the information obtained is punched on cards and fed to the computer
in the field. Furthermore, the accuracy of the computer
diagnosis is still improving with refinements in the data
matrix. (R-12)

Overall and Williams (R-9) developed a computer program for the
diagnosis of thyroid function. The object was to classify patients
into one of four classes: 1) no thyroid disease, 2) hypothyroidism,
3) enthyroidism or 4) hyperthyroidism. By analyzing 879 cases, the
authors obtained a disease-probability matrix which included 21 in-
dices of thyroid function. Although over 800 cases were involved in
the analysis, not all of the 21 measures were available for each
case. Relative frequencies of each attribute were based on the num-
ber of cases in which the necessary data were available. Independence
of attributes was assumed, although the authors note that this assump-
tion is suspect.

In an extensive series of tests, the program performed extremely
well. According to the authors

... computer diagnoses agreed with the clinical diagnoses
in over 96% of the cases in which anything like complete
data were available. (R-9)

Both of these examples of computer-aided diagnoses lend credence to
the belief that Bayesian attribute-disease models of diagnosis may
prove extremely useful in a whole range of medical applications.

As noted earlier, not all applications of mathematical methods
to medical diagnosis have been founded on Bayesian inference. An
interesting example of a different view of the problem involves
considering a point in an n-dimensional space (where n is the
number of attributes). From past experience with diseases, one can
consider each disease as representable by a class of points in the space. The diagnosis of the current disease is derived from a consideration of the "closeness" of the corresponding point to the classes for each of the known diseases respectively.\(^1\) In a recent paper (R-7), Lerner discusses the use of such an approach in the recognition of handwritten letters and the detection of oil-bearing strata in petroleum geology. In the latter problem (another type of diagnostic problem), he reports that a program based on this method far surpassed the performance of the most experienced experts. He then advocates the application of this method to problems of medical diagnosis and asserts that the possibilities of this approach "considerably exceed those of doctors-diagnosticians."

While this method differs markedly from that employed in the two medical applications above, it shares with them a very important limitation. In Chapter 1 it was suggested that the diagnostician performs two major tasks in his problem-solving. The first task is the interpretation of attributes manifested by the system being diagnosed. An equally important task is the selection of an appropriate testing strategy. All of the programs above map a set of attributes into a diagnosis in one stage. There is no test selection function performed in any of these programs. As a result, all the data which are to be employed by the program must be collected before the program is invoked. There is no opportunity for selective testing based

---

\(^1\)This approach will be examined in more detail in Section A of Chapter 3.
on an analysis of an incomplete set of attributes. Thus, it may happen that the cost of determining a number of attributes (for example, by taking an X-ray) is incurred unnecessarily. While this may not be a major problem in the particular areas discussed above, it is easy to think of situations in which this approach would be highly undesirable. Consider, for example, the computer-aided diagnosis of diseases from a group which exhibit clusters of relatively disjoint attribute patterns. The approach outlined above required the determination of a full set of attributes to be made available to the program. Since only a small subset of the set of all attributes is necessary for a diagnosis, many attributes are unnecessary in any particular application. If the cost of obtaining these unnecessary attributes is high, then the diagnostic procedure will be less than satisfactory. This is because the quality of diagnosis should reflect its cost as well as its accuracy. As Lusted has observed (R-17),

A great many medical diagnostic tests have been developed to supplement the patient information obtained from history and physical examination. These tests vary greatly in the amount of discomfort to the patient, complexity, and cost. It is obvious that diagnostic tests should be kept to a minimum.

It seems that a more satisfactory solution is to permit the diagnostic program to operate sequentially, choosing tests for the user to run based on a continually updated view of the problem. The program could engage in a dialogue with the user as it performs both the inference and test selection functions of diagnosis. The
testing strategy evolved by the program should reflect the information derived from the attributes observed to date, past experience with similar systems, the cost of tests, and the relative seriousness of various disease states. Part of the research reported in this thesis is aimed at developing a program which satisfies these requirements.

Less has been done with computer-aided diagnosis in other areas. One problem which has received attention, however, is the diagnosis of faults in a computer. Although the problems here are not well understood at present, recent research (R-18) shows considerable promise. Significant results pertaining to the selection of an optimal set of diagnostic tests have been obtained (R-19), but they are restricted to the case of a single fault.

B. Perspectives on Diagnosis

One of the chief motivations for this research is belief that a computer is potentially a very useful tool to be employed in diagnostic problems. The need for such a tool becomes apparent when the difficulty of particular diagnostic problems is considered.

A considerable portion of the effort expended in implementing computer programs is devoted to program debugging. As programming applications become increasingly sophisticated, the complexity of the associated problems of debugging increases at an equally rapid rate. The tremendous effort required to debug a large operating system is a testament to the magnitude of the diagnostic problem in-
involved. This is so even though many of the programmers involved in such an effort are experts.

The non-expert who ventures into the world of programming also faces many diagnostic problems. Often the magnitude of these problems relative to his limited programming skill and experience is such as to prevent him from effectively using the computer in his particular research. In both these cases, there is a need for an improved diagnostic facility. Research into the potential usefulness of diagnostic computer programs seems especially appropriate in this context.

Much the same situation exists in medicine, although here there exists more explicit evidence of difficulty of problems in medical diagnosis and the need for new aids in the problem-solving process. Physicians receive extensive training in their profession, and they devote considerable efforts to the development of their diagnostic acumen. For all their training, however, the difficulties of the diagnostic problems confronting them have resulted in a surprisingly low level of performance. In a recent research report of the United States Public Health Service entitled "Completeness and Reliability of Diagnosis in Therapeutic Practice," the author concludes from an extensive study.

On the basis of available evidence, I estimate if we regard all diagnosable diseases at a given time that are considered of significance for current health as 1, the number of therapeutically determined diseases constitute numerically 0.4. Of this 0.4 nearly half are conditions diagnosed incorrectly. This suggests that correctly
Chapter 3

TWO VIEWS OF DIAGNOSIS

This chapter concerns the theoretical framework for the study of computer-aided diagnosis. Here the nature of the diagnostic problem is examined and the model for the problem is developed. Two views of diagnosis are considered. The first view is that of diagnosis as a pattern recognition problem. This consideration brings into focus those features of the diagnosis which distinguish it from the "classical" pattern recognition problem. The second view involves analyzing diagnosis as a problem in sequential decision-making. The problems arising from this formulation are explained and various means of circumventing these problems are discussed. The view of diagnosis as sequential decision-making is the one taken for this research and so this discussion leads directly to the specification of a computer program for performing general diagnosis.

In the following chapter, a discussion of a program to perform general diagnosis is presented within the framework of the program actually implemented as part of this research. Each of the major logical functions of the program is discussed in turn with the emphasis on the way in which these functions match the requirements of a diagnostic process. In a very real sense, the program can be taken as a statement of an overall diagnostic strategy for computer-
Aided diagnosis.

A. DIAGNOSIS AS A PROBLEM IN PATTERN RECOGNITION

Consideration of the diagnostic problem as a pattern recognition focuses attention on some of the more significant aspects of the problem. Also it is quite natural to conceive of diagnosis as a pattern recognition problem. The observable attributes associated with the system of interest in a diagnostic problem do constitute a pattern which is the direct evidence upon which a classification decision is based. Thus a medical diagnostician confronted with an ailing patient employs his observations of the patient's symptoms and signs in conjunction with his experience and training to deduce the nature of the patient's problem. While there are many features which are shared by the diagnostic problem and a wide variety of particular pattern recognition problems, there are additional constraints on the former which add to its complexity. The purpose here is to explore both the similarities and differences between the diagnostic problem and the "classical" pattern recognition problem.

The classical pattern recognition problem is fundamentally one of recognizing class membership and establishing decision criteria for measuring membership in each class. Given a set of pattern classes the problem is to assign a new pattern to one of the classes. For example in the recognition of handwriting, knowledge of the general properties of individual letters is utilized in the determina-
tion of the identity of that segment of handwriting which is currently of interest. The individual pattern classes may be known in a variety of ways ranging from a set of representative patterns to a functional characterization of the probabilistic process by which patterns of the class are generated. In general, a pattern is comprised of a set of features; each feature being represented by some numerical value. In the handwriting recognition problem, an unknown letter could be represented by numerical values for such features as the height, number of loops and the number of intersections the letter makes with certain reference lines. Such a representation leads quite naturally to the representation of a pattern as an \( n \)-dimensional vector where \( n \) is the number of features which are taken to be relevant to the classification problem.

Hence, each pattern class can be conceived of as a set of points in an \( n \)-dimensional space. Similarly, any pattern which is to be classified can be represented as a point in the space (provided, of course, the same set of features obtains). The problem of classifying a new pattern sample involves determining the "closeness" of the sample to each of the respective classes. For instance, we may decide a certain letter is an "e" because it more closely resembles representatives of the class of known "e's" than representatives of other classes of letters. In the \( n \)-dimensional space, this corresponds to measuring the distance (in some abstract sense) between the point denoting the new pattern and those representative of
the various classes. The problem of establishing criteria upon which the "resemblance" of a particular letter to the class of letters known to be "e's" is but one instance of the general problem of deciding exactly how the "closeness" of a sample to various classes is to be established. For a given application, the determination of an appropriate metric is a fundamental problem of pattern recognition.

Consider the schematic of a pattern recognition problem presented in Figure 1. Here two pattern classes are of interest, classes A and B. In this case, there are two features in the patterns and an orthogonal coordinate system corresponding to these features is shown. Notice that in this simple example all members of class A are "closer" to all other members of class A than to any member of class B and vice versa. Unfortunately, this condition does not hold in general. The more common case is to have "close" or intersecting pattern classes. Members of a class can be closer to members of another class than to certain other members of the same class. For example, some handwritten "e's" look very much like "i's" and vice versa. A schematic of intersecting pattern classes is presented in Figure 2. The problem of recognizing the pattern x in these figures involves establishing a metric which can be employed to decide whether x is "closer" to the class A or the class B (or in some cases deciding that x is a member of neither A nor B). The actual decision regarding the identity of x can be based on the cost of misclassification as well as the chosen metric.
Figure 1
Two Pattern Classes
Figure 2

Intersecting Pattern Classes
When the pattern classes are inherently close or intersecting in the space, recognition is more difficult. In some cases matters can be improved by devising class separating transformations. Such a transformation has the property that the classes resulting from the application of the transformation to the original classes are more separated from one another in the transform space. Figure 3 represents the effect of a class-separating transformation on classes A and B. The particular transformation will depend on both the characteristics of the classes to be transformed and the constraints placed upon the transformation. Suffice it to say here that transformations of this type can be derived by solving constrained optimization problems. Given such a transformation, the pattern to be recognized is first transformed and then its "distance" from each of the transformed classes is measured. It is this distance in the transform space which is incorporated in the classification decision rule.

The problem of diagnosis has much in common with the pattern recognition problem discussed above. The pattern classes in the pattern recognition problem correspond to the system states in the diagnostic problem, and there is a similar analogy between particular patterns and sets of attributes. The object of diagnosis is to classify a set of attributes as being a manifestation of a particular system state. Again, the notions of an n-dimensional space and vector representations of attribute sets is suggested. There is an important difference between diagnosis and the pattern recognition method out-
lined above. In the latter, it was assumed that a pattern to be recognized is given as a point in the sample space. This implies a complete specification of the corresponding vector. In the usual diagnostic problem, the pattern of attributes is incompletely specified. There exist means for obtaining the values of unspecified components of this vector (tests which can be run, etc.), but in general there is a cost associated with the use of these means. These costs make it advantageous to analyze the diagnostic problem sequentially and to make decisions based on an incompletely specified attribute vector.¹ Doctors, for example, make diagnostic decisions without performing all possible tests on the patient.

Thus, in the diagnostic problem, one is concerned throughout with subspaces of the sample space. The dimensionality of the subspace which contains the pattern vector is reduced by obtaining previously unspecified values for certain pattern features. In general, each value so obtained reduces the dimensionality of the subspace in which the point corresponding to the fully specified attribute set must lie. Because of the costs associated with the tests for particular attributes, a good diagnostic scheme must include some means for assessing the expected value of a test in determining the class to

¹Note that this distinction between pattern recognition techniques and diagnostic techniques is not a necessary one. Certain pattern recognition schemes have employed sequential methods while most medical diagnosis programs have avoided sequential analysis entirely. The distinction, however, does have appreciable generality.
which the attribute vector belongs. While the sequential nature of the diagnostic process complicates its realization, it also offers a potential advantage of the pattern recognition scheme described above. Although an attribute vector may be incompletely specified, the subspace corresponding to it may include only one class. In such a case it may be possible to make the classification decision at that point without investigating the remaining attributes. This reduction in the amount of the processing required for a classification decision is especially significant when many of the system states are represented by disjoint subspaces in the n-dimensional sample space. This reduction can be obtained only if the diagnostic scheme incorporates some stopping rule for the attribute sampling (or testing) process.

So while the pattern recognition problem and the diagnostic problem have a number of features in common, there are significant differences between the strategies indicated for their solution. The former problem concerns the classification of a fully-specified vector into one of a number of known classes. The latter problem is equally one of classification, but the initial specification of the vector is generally incomplete. Part of the problem is to ascertain which tests to run (at some cost) to obtain a more complete specification of the vector. Decisions based on an incompletely specified vector are the rule rather than the exception. Note, however, that there may well be inherently close or intersecting
classes in the diagnostic problem as in the pattern recognition problem.

One aspect of the pattern recognition problem which was not discussed above was that of choosing the coordinate system for the sample space. This has a direct and significant analogy in diagnosis. In the discussion of pattern recognition, it was assumed that the pattern features were given. The efficiency and the accuracy of the recognition scheme often can be improved by the selection of a new coordinate system (set of features). The problem of establishing the coordinate system is often termed the pattern detection problem.

Thus, for example, in Figure 1 the dotted coordinates are in a sense more efficient, for they permit the characterization of classes A and B solely in terms of one coordinate. Again general mathematical techniques are known for establishing "good" coordinate systems for a number of problem types.

Clearly, a similar situation obtains in diagnostic problems. Generally speaking, the attributes considered in diagnostic problems are chosen without any particular regard for the efficient separation of pattern classes. It is apparent, however, that there is potential value in conducting such an analysis for a given problem area. In certain areas, especially in a medical diagnosis, there has been an increasing awareness of the importance of the proper choice of pattern features; a number of articles on the "taxonomy of disease"
have appeared in the literature.\footnote{In recent years, there has been much medical work directed at developing specific tests for diseases. Thus a particular attribute (test result) may indicate exactly one disease.} While this problem is an extremely interesting one, it is beyond the scope of this thesis. Here the pattern features of attributes for any particular area are taken as given.

This discussion provided only a brief overview of pattern recognition and its relation to diagnosis. The particular type of pattern recognition which constitutes diagnosis will be explored in considerable detail in other sections of this work.

B. DIAGNOSIS AS A SEQUENTIAL DECISION PROBLEM

In this section, the problem of diagnosis is formulated in terms of statistical decision theory. This formulation is in very general terms, but it suggests a number of the factors which complicate particular diagnoses. In many areas of diagnosis, attention is focused on a system. In medicine the system is a human being; in program debugging, a computer program. The object of the diagnostic problem is to determine the state of the system (e.g. the disease in the person or the error in the computer program). This state is one of a finite but perhaps quite large number of possible states. Information about the state of the system can be obtained by performing a variety of tests on the system. Information obtained from testing
coupled with experience with other diagnostic problems is employed by the diagnostician in his attempt to deduce the state of the system. In this work, the goal of diagnosis is taken to be the determination of the state of the system of interest. It is assumed that knowledge of the system state will greatly facilitate further (non-diagnostic) action. For example, the identification of the state of a patient as "tuberculosis" may lead directly to a course of treatment. The system under consideration here is a finite state machine. The diagnostician knows about all the states of the machine in the sense that he has available probability distributions which characterize the response of the machine to certain tests given the machine state. In particular, this information relates attributes, the results of the tests, to particular system states.\(^1\) At the outset of the problem, the machine is in a particular, but unknown state and the task of the diagnostician is to employ the available tests to obtain information about the identity of that state. Tests are assumed to be free from error and it is further assumed that they do not alter the state of the system.

Associated with each test is a cost of applying it to the system (called the testing loss) and thus it is advantageous to make a decision about the state of the system based on a limited number of tests. On the other hand there is a decision loss associated with an

\(^1\)An attribute is binary-valued. That is, each attribute is either present or absent. A test is used to determine the presence or absence of some number (perhaps greater than one) of attributes.
incorrect decision. The loss resulting from each particular decision about the unknown state as a function of the actual state is given by a loss function for the problem. For example, the loss resulting from the decision that a tumor is benign when it is in fact malignant is very costly and a diagnostic procedure for tumors should take cognizance of this fact. In general, the possibility of loss for an incorrect decision indicates the value of extensive testing prior to any decision. The problem is to balance the testing loss and the decision loss in a sequential decision function for the problem. This function would specify a diagnostic procedure such that the total expected loss of the final decision is minimized. The following is a formal statement of this problem.

1. The states of the Machine $M$ are $M_j$, $j=1,n$, and the current state is denoted by $M_0$. It is assumed that $M_0$ does not change during the course of the diagnosis.

2. $\Pi = (\Pi_1, \ldots, \Pi_n)$ is a vector of a priori probabilities for $M_0$. That is $\Pi_1 = P(M_0 = M_1|\xi)$ and $\xi$ denotes experience.

3. $T = \{t_1, \ldots, t_r\}$ is the set of available tests.

4. $(t_i)_q$ is a vector of length $q$ with each $t_i \in T$. It represents a series of tests with test $t_i$ being run at the $i^{th}$ stage.
5. \( S = \{ S_1, \ldots, S_p \} \) is the finite set of possible attributes for \( M \) and the set \( T \).

6. \( (S_i)_q \) is a vector of length \( q \) with each \( S_i \in S \). It denotes a sequential set of attributes.

7. \( d_t \) is a terminal decision and \( d_t \in D_t \) where \( D_t \) is the finite set of all possible terminal decisions.

8. \( C((t_i)_q, (S_i)_q) \) is the testing loss for a sequence of tests \( (t_i)_q \) resulting in the attribute sequence \( (S_i)_q \) followed by terminal decision at stage \( q+1 \).

9. \( P((S_i)_q/M_j) \) is the conditional mass function for \( (S_i)_q \) given \( M_j \).

10. \( P((t_i)_q, d_t/(S_i)_q) \) = conditional mass function for the testing sequence \( (t_i)_q \) followed by terminal decision \( d_t \) given the attribute sequence \( (S_i)_q \).

11. \( L(\Pi, d_t) \) is the decision loss function.

12. \( \vartheta(d_t/(t_i)_q, (S_i)_q) \) is the sequential decision function to be determined.

Let \( \overline{L}_1(\Pi, \vartheta) = \) the average decision loss
\( \overline{L}_2(\Pi, \vartheta) = \) the average testing loss.

then the problem is to determine \( \vartheta \) such that

\[ \overline{L}_1(\Pi, \vartheta) + \overline{L}_2(\Pi, \vartheta) \]

is a minimum.

\[ \overline{L}_1(\vartheta, \pi) = \sum_{q=0}^\pi \sum_{j=1}^{\pi} \sum_{d_t} \sum_{q=0}^\pi \sum_{d_t} d_t \vartheta(d_t/(S_i)_q, (t_i)_q) \cdot P((S_i)_q/M_j) \]
\[ L_2(1, \theta) = \sum_{q=0}^{q} \sum_{t_1} \sum_{t_2} \prod_j \sum_{s_q} P((t_1)_q, (t_2)_q) \cdot C((S_2)_q, (t_1)_q) P((S_1)_q/M_j) \]

where \( T_q \) is the set of all \((t_1)_q\)
and \( S_q \) is the set of all \((S_1)_q\)

The great difficulty with this problem is not conceptual but computational. For finite sets of attributes and decisions, the optimal solution can be obtained in principle by laying out a decision tree.

Such a tree includes by two types of nodes—decision nodes and "nature's nodes." Nodes of the former type are characterized by 1) a current view of the diagnostic problem as embodied in the probability distribution over the states of the system. (This distribution accounts for both the attributes observed to date and the \textit{a priori} likelihood of system states in a manner to be made explicit later in this thesis.), and 2) a \textit{branch} emanating from the node for each alternative available to the decision-maker at the node. In the context of diagnosis, then, there is at each decision node one branch for each possible test which can be run and one branch corresponding to a terminal decision. Once an alternative branch away from a decision node has been chosen by the decision a particular one of nature's nodes is encountered.

Such a node represents the possible outcomes of the decision corresponding to the branch which leads to the node. Each of these "outcome branches" leads to a new decision node. A portion of such a decision tree is shown in Figure 4. The node A is a decision node
Figure 4
Section of a Decision Tree
which is characterized by the prior probability distribution and history embodied in the path to the node. There is a branch from this node for every relevant test (given the history and \( \mathbf{H} \)) as well as a branch corresponding to a terminal decision. If a particular test is chosen, say test \( T_1 \) in the diagram, a new node (here node B) is obtained. This node is one of the "nature's nodes" mentioned above. There is a branch from this node for each possible test outcome given \( T_1 \) and given the state of the diagnosis at B, the conditional probability for each attribute branch can be computed.

If it is assumed that the total number of potentially useful test sequences is finite then the entire tree for the diagnosis can be specified. By folding back this tree in terms of expected loss, one can obtain an optimal decision for every decision node on the tree. This problem is amenable to techniques such as dynamic programming. There is little conceptual difficulty in solving the problem.

The difficulty is the exponential growth of the number of decision nodes with the number of signs and tests. Since diagnostic problems involving large numbers of possible attributes are common, it is expected that the problems of searching large decision trees contribute a large part of the complexity of specific diagnostic problems. One of the major concerns of this research is with the development of effective heuristics for this tree searching problem. While such heuristics produce sub-optimal solutions, it is possible
that the reduction in the size of the search space may more than offset this disadvantage.

As an indication of the potential size of such a problem, consider the diagnosis of a ten-state, twenty-attribute system. Such a case might arise when one was attempting to employ twenty attributes to classify a person into one of ten disease groups. Assuming that there is a test for the presence or absence of each attribute and that each test is run but once, the number of decision nodes in the decision tree for the problem can be expressed as

\[ n^N_k = \frac{2^k n!}{(n-k)!} \]

Where \( n^N_k \) = the number of decision nodes  
\( k \) = the depth of the tree  
\( n \) = the number of tests.

For this example, \( n \) is 10, and the number of decision nodes in a tree of depth \( k \) is given by

\[ 10^N_k = \frac{2^k 10!}{(10-k)!} \]

Table 1 gives values of \( 10^N_k \) for selected values of \( k \). Notice the extremely rapid increase of \( 10^N_k \) with \( k \). Also, at any given decision node at depth \( k \) it is necessary to compare \((n-k+1)\) decisions (one for each of the \( n-k \) remaining tests and one for the possible terminal decision). Although in many cases such an attribute set is highly
redundant, it is often possible that a depth of 5 may be required for an optimal decision. In such a case there are still almost a million decision nodes. Even in the simple case of a specific test for each state, there are $n!$ different decision nodes, where $n$ is the number of states. Again the growth of the decision tree with $n$ is enormous.

Table 1
Growth of Search with Depth

<table>
<thead>
<tr>
<th>$k$</th>
<th>$10^N_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>5,760</td>
</tr>
<tr>
<td>5</td>
<td>967,680</td>
</tr>
</tbody>
</table>

While there are certain factors in particular diagnostic areas which allow the decision tree to be considerably reduced in size, the determination of an optimal testing strategy remains computationally infeasible for the most part. The value of good heuristics is apparent from considerations such as the above.
C. HEURISTIC CONSIDERATIONS IN TEST SELECTION

As previously noted, the problem of obtaining an optimal testing strategy for a particular diagnostic area generally will be computationally infeasible. Many diagnostic areas are characterized by overlapping attribute patterns for different states and highly redundant attribute patterns, however, and there is strong motivation for developing "good" diagnostic strategies. Unnecessary or redundant tests may exact a high cost which could be avoided by a more efficient testing strategy. In certain areas of medicine, tests are quite costly and may cause the patient considerable discomfort. If such tests contribute little additional information to the diagnosis, it is especially important that these tests not be employed. A second difficulty is that a poor sequence of tests may generate results which, being unnecessary for a diagnosis, simply tend to obscure the truly relevant attributes. One approach to this problem was mentioned earlier. This approach consists essentially of sharpening the taxonomy of the problem states. While success here can substantially reduce the redundancy in attribute patterns, it will not necessarily make the determination of an optimal testing strategy computationally feasible. While the possibilities of this approach are extremely interesting, they will not be considered here. For the purposes of this work, it is assumed that in any diagnostic area, the attributes for states are given. No attempt is made to improve on the efficiency of the given attributes with regard to the characterization of the states.
A second approach to the problem of test selection is to develop heuristics for the selection process. Such heuristics would employ only limited segments of the decision tree in evaluating the potential efficacy of relevant tests. The general nature of the diagnostic problem is such as to offer two distinct means of controlling the growth of the number of decision nodes considered. The size of the decision tree (the number of decision nodes) depends on the number of tests considered at any decision node, and the depth of the analysis of that tree. By restricting either of these quantities, the diagnostician can limit the growth of the tree. In this discussion, heuristics which limit the number of branches from a decision node will be called breadth-limiting; and those which limit look-ahead, depth-limiting. In what follows, the set of relevant tests for a particular decision node will be taken to mean all those tests which can result in a sign which is manifested by at least one state with a non-zero probability in the prior for the node. The set of relevant tests is a subset of the set of all tests.

Breadth-limiting heuristics are easily formulated. Perhaps the simplest is to limit the number of branches from a decision node to some fixed number. If this number is less than the number of possible test branches for a given node, then a decision rule for selecting (or rejecting) branches must be established. In terms of the diagnostic problem, this means selecting a subset of the
relevant tests for consideration given a prior distribution for
the unknown state.

Heuristics which limit the number of branches from a decision
node to a certain fixed number have several shortcomings. Principal
among these is the problem of the selection decision rule. If
certain tests are to be selected over other tests, then some measure
of test effectiveness should be employed. That is, one test is
chosen over another because by some standards the former is more
promising. The difficulty with this is that almost any reasonable
measure of expected test effectiveness requires information obtained
from a look-ahead in the decision tree. To assess the potential
value of a particular test, one needs to consider the likelihood
of various test results and the value of these results in improving
the current view of the diagnostic problem. If this look-ahead is
performed, the purpose of the heuristic is defeated. A breadth-
limiting heuristic is intended to select a subset of relevant tests
without employing a look-ahead procedure. Then this subset is
subjected to further analysis.

Since a breadth-limiting heuristic probably should not employ
a look-ahead to obtain information, the only information upon which
it should make its decisions is that contained in the current prior
distribution and the test cost data.\(^1\) Thus one possible breadth-

---
\(^1\)This may be overly restrictive, since one can imagine breadth-
limiting heuristics which employ a priori probabilities. Such heuristics
are not in general very sophisticated, and are not considered here
limiting heuristic is "At any decision node consider at most 5 tests in order of increasing cost." This heuristic obviously ignores all the information embodied in the current prior distribution, and so while it limits the breadth of the decision tree, it does not appear to be a particularly good heuristic.

An alternative breadth-limiting heuristic employs the current prior distribution to generate the subset of relevant tests which are to be considered. For each state there are a number of relevant tests. These tests may produce an attribute which is significant in the diagnosis of the state. Consider, for example, a problem in medical diagnosis in which one of the diseases which currently is being considered as the explanation of the patient's problem is tuberculosis. Since a chest X-ray is a useful test in the diagnosis of this disease, it would be considered a relevant test. On the other hand, the absence of any attributes associated with an injured ankle would exclude an X-ray of the ankle from the set of relevant tests at this stage in the diagnosis. The union of the sets of tests relevant to currently possible states is the set of all relevant tests. By limiting the number of states considered, one can limit the number of branches at the decision node. A heuristic of this type is "Create the total set of relevant tests from the sets of relevant tests for the three most probable states (based on the current prior)." In the above example, if tuberculosis were currently the most probable disease, the diagnostician might choose to
consider only those tests which are relevant to tuberculosis and ignore all others. Note that such a heuristic is only potentially breadth-limiting. There is no guarantee that any test branches are excluded in this way since the same set of tests may be relevant to all states currently being considered. Also the actual number of branches from a given decision node is not specified and generally will vary from node to node.

Such an heuristic has intuitive appeal, however, because it prunes branches corresponding to tests for attributes specific to improbable states. If an attribute for an improbable state is also manifested by a state which is currently quite likely, however, then the appropriate test will be included in the set of those considered. The weakness of this heuristic lies in its sensitivity to the current probability distribution on the states of the system. This distribution can undergo radical change upon the observation of one new attribute. Thus, states which were previously unlikely can become very probable as a result of one new observation. This phenomenon cannot be accounted for by breadth-limiting heuristics based on the current prior distribution. In fact, no breadth-limiting heuristic which does not employ look-ahead can completely account for this possibility. A breadth-limiting heuristic of this type is applied at each decision level, however, and in some sense it can "recover" from a drastic change in the probability distribution. This capability is derived from the consideration of the probability distribution at the current decision node. Thus, when a state which was formerly improbable
at one decision node becomes probable, it will automatically be incorporated in the test selection scheme at the next level. Unfortunately, this state may not become very likely until a large number of tests have been run. If it is the actual state, its probability can remain low simply because the "wrong" tests are being run. Thus a doctor may fail to obtain a chest X-ray of a patient because it seems unlikely that the patient has tuberculosis, when this disease would become very probable if only the X-ray were taken. This, of course, is a general problem encountered with all test selection heuristics.

The evaluation of the heuristic involves a comparison of the benefits of its tree-pruning power with the losses incurred from the sub-optimal testing strategies it produces. In general, a heuristic based on the current probability of various system states appears to be the most promising form of a breadth-limiting heuristic, but its actual value can be determined only in the context of a particular diagnostic problem area. For example, in one area a breadth-limiting heuristic which restricts the search to tests relevant to the n most probable states may prove useful. In another area, tests relevant to all states with current probability greater than some threshold may be considered. Finally, in certain areas breadth-limiting heuristics may be of no value regardless of the particular specification. One of the areas explored in this research is that of evaluating several breadth-limiting heuristics in particular diagnostic problem areas. In such an evaluation, the capability of closed diagnosis may be particularly valuable.
As noted in the beginning of this section, there are two general types of heuristics which reduce the number of decision nodes considered in test selection: breadth-limiting and depth-limiting. As the name of the latter implies, such heuristics limit the extent of the look-ahead in the decision process for test selection. As with breath-limiting heuristics, there are several variations of the depth-limiting heuristic to be considered.

Perhaps the most obvious form of the depth-limiting heuristic is one which sets a fixed depth of search for all branches of the tree. Thus given a particular decision node, the search would proceed down all branches from that node to a depth \( k \), where \( k \) is a fixed number. The information derived from this search would then be employed in a decision rule to determine the test to be run next. The parameter \( k \) is a relative depth, that is at a decision node at level \( p \), the search is conducted to a depth of \( p+k \) before making the decision for level \( p \). An alternative depth-limiting heuristic might employ a variable depth look-ahead. Such a heuristic might attempt to explore more "promising" branches to a greater depth than less promising ones. The difficulty here is to decide which branches are promising. It is, in fact, the general problem of heuristic test selection all over again.

There are several problems to be resolved in the development of any depth-limiting heuristic. First consider the effect on the decision process of limiting the depth of search. If the depth is
limited to $k$, then the "terminal" nodes will be characterized by probability distributions for the unknown state. (See Figure 4.) Since, in general, there will be a number of states with non-zero probability at any given terminal node, there must be some way of assessing the value of being at the node. One of the major problems in the development of depth-limiting heuristics then is the definition of measures of the desirability of nodes which do not represent a certain diagnosis.

One way of establishing the value of a node is suggested by the presence of a loss function. The value of the node can be obtained by assuming a decision about the unknown state is to be made there. Then the prior distribution for the node and the loss function can be employed to find the expected decision loss for the node. From this loss the value of the node is derived. While this measure seems to be a natural one, it is not without its weakness. The problem with the measure is that it is based on an assumption which is generally untrue. In most cases, one will not make terminal decisions at the nodes which are "terminal" for one state in the look-ahead. For example, if the search depth is limited to 2, the value measure assumes that a terminal decision will be made two tests from this point. Since the actual terminal

---

1 An additional assumption should be noted here. This is the assumption that given the prior distribution, the minimum expected loss decision is made.
decision may not be made until many tests have been run, this measure distorts the value of tests considered for the current level. The problem is that the values of the loss function at the decision nodes of a given level may bear little relation to the values of the best testing strategies which include these nodes. The potential effectiveness of this "loss function" measure is difficult to assess. The expectation is that it depends upon the particular problem area in which the measure is employed.

A second problem with this heuristic is its potential sensitivity to the actual loss function employed. If the heuristic is very sensitive to the loss function then uncertainties as to the true nature of this function may result in testing strategies which are decidedly sub-optimal. The problems of accurately assessing the loss function for a particular application will be discussed later in this thesis.

The above discussion of breadth-limiting and depth-limiting heuristics purposely considered the two independently in order to make clear the considerations involved. The motivation for such heuristics in test selection is the desirability of reducing the number of decision nodes considered. Since the number of decision nodes is dependent on both the breadth and depth of the search, the heuristics employed in an actual problem will interact. Generally speaking, the depth of the search can be increased only at the expense of the breadth, because there is a constraint on the total number of nodes to be considered. The particular balance of
these two heuristics may significantly affect the effectiveness of the test selection process. An additional complication is introduced by the possibility of changing this balance during the course of the diagnosis when many states are possible. It may be desirable to limit the depth and allow full breadth. This is particularly true if the prior distribution is quite diffuse. As the diagnosis progresses and certain states are eliminated from further consideration the breadth of the tree may be reduced and the depth of search may be increased correspondingly. The relation between the depth and the breadth of the search is an important matter for investigation in the development of heuristic test selection schemes.

More of the practical considerations involved in developing heuristics will be discussed in a later section describing the heuristics employed by the diagnostic program and their relative effectiveness.
Chapter 4

A DIAGNOSTIC SYSTEM

The considerations outlined in the previous chapter led to the design and implementation of a diagnostic system. This system is composed of three major parts. The first is a set of programs which perform the actual diagnostic function. The second is a set of programs which facilitate the study of a variety of diagnostic problems and strategies. The third part of the system is the information structure which contains all the relevant information which these programs employ in performing diagnosis for a given problem area. While the content and, to some extent, the nature of the information structure vary with the particular application, it is convenient to consider this structure as a third general part of the diagnostic system. These three aspects of the diagnostic system will be discussed in detail in this chapter.

The diagnostic system is currently operating on the Project MAC time-sharing system at the Massachusetts Institute of Technology. The diagnostic system is designed to exploit the interactive capabilities of the time-sharing system. The programs of the diagnostic system are written in MAD and FAP. They make very extensive use of the SLIP-MAD system developed by Professor Joseph Weizenbaum of M.I.T. The SLIP-MAD system (hereafter referred to as SLIP) is a set of list processing functions embedded in the host language MAD. Because
discussions of SLIP are available elsewhere (R-20), only a brief outline of the system is given here.

The basic data structure employed in the SLIP system is a **SLIP list**. A SLIP list is a list composed of **cells** where a cell is a pair of adjacent words of storage. The first word of the pair is divided into an identifier field, a **link-left** field and a **link-right** field. Each cell in a SLIP list contains a **forward** (right) link and a **backward** (left) link. SLIP lists are **symmetric** in the sense that lists have no particular orientation, the top and bottom of a list are equally accessible. The identifier is used to indicate the type of element stored in the second word of the cell. This element is referred to as the **datum**. An example of a simple SLIP list is given in Figure 5. Notice that any cell may contain an actual datum rather than a symbolic designation for the datum.

Every SLIP list contains a special cell known as the **header** of the list. This cell contains the address of the first cell on the list in its **right-link** field and the address of the last cell on the list in its **left-link** field. Any storage location which contains the address of a list header in both its address and decrement fields is said to contain the **name** of that list. A SLIP list structure can be defined as a SLIP list whose data terms may themselves be names of SLIP lists.

There may be associated with any SLIP list a **description list** or **DLIST**. If a SLIP list possesses a DLIST, the address of the header
Address of Cell  

22402  

22404  

22406  

22410  

22412  

Contents of Cell  

22412  22404  

22402  22406  

22404  22410  

22406  22412  

22410  22402  

Figure 5

A Simple Slip List
of the DLIST is contained in the left-link of the datum of the header cell. The DLIST, which is itself a SLIP list, is used to store data pairs. A variety of SLIP functions are available for creating and accessing these pairs.

The SLIP library is a set of functions for manipulating SLIP lists. Typical functions permit the reading or searching of lists, additions to or deletions from lists, and the creation or erasure of lists. SLIP maintains an available space list, and the system includes an automatic garbage collection facility.

Because the SLIP library consists of compiled subroutines which can be invoked from MAD or FAP programs, SLIP programs run at object speed. The fact that SLIP is embedded in an algebraic language, MAD, means the full arithmetic and logical capability of the latter is available to the programmer in a list-processing application. These two features make SLIP a convenient language to use in the implementation and debugging of a large list-processing application such as the diagnostic system developed in this research. For this particular application, the need for both the flexibility of list-processing and the algebraic power of MAD is well served by the SLIP-MAD system.

A. THE INFORMATION STRUCTURE FOR THE DIAGNOSTIC SYSTEM

The manner in which the information relevant to a particular diagnostic problem area is organized has a considerable effect on the capabilities of the diagnostic program. The information contained
in this structure for a particular application constitutes the "ex-
perience" which the diagnostic program brings to bear on problems.
This experience includes relationships between observable attributes and
states of the system to be diagnosed. For example, in an area of
medical diagnosis, the information structure would contain the re-
lationships between signs and symptoms and the appropriate diseases.
Also included in the structure is information about the tests which
are relevant to the given diagnostic area and their associated costs.
Because of the probabilistic nature of many of the attribute-state
relationships as well as other important relationships, the informa-
tion structure must maintain a large number of individual probabilities.
The general content of the information structure will be explained
below.

The large number of state, attributes, and tests encountered
in many diagnostic areas places a premium on efficient searching of
the information base during a diagnosis. The efficiency of search
can be maintained at an acceptable level only through the proper organi-
zation of the relevant information.

A number of questions were considered in the design of the
information structure currently employed by the diagnostic system.
One of the principal questions was that of what information should
be maintained in the structure. To a large extent, the particular
diagnostic problem under investigation here determined the answer to
this question. Since the model of diagnosis makes reference only to
states, attributes, tests and various probabilities, these factors
constitute the basic information blocks in the structure. Another question is how, given the basic information blocks, these blocks should be related in order to facilitate access by the diagnostic program to the relationships which are significant in the deductive process of diagnosis. For example, the following questions typify the types of demands made on the structure.

- What are the symptoms of pneumonia?
- Which diseases exhibit a rash on the arms as an attribute?
- What is the probability that a patient will have a temperature greater than 103° given that he has pneumonia?

The information structure described here was developed through the consideration of a number of alternative forms, although there obviously are other forms which might serve as well. To a certain extent, the information structure reflects the use of the SLIP system by the diagnostic program. For example, the information structure is a SLIP list structure. While in certain instances this results in inefficient utilization of main storage, this disadvantage was more than offset by the convenience of being able to employ the full SLIP library in the development of the diagnostic system.

A basic information block in the structure is either a state, an attribute, or a test. Each of these basic blocks is represented by a SLIP list in the information structure. In what follows these
blocks will be referred to as state lists, attribute lists, or test lists. A typical state list is depicted in Figure 6; in this instance, the state list corresponding to pneumonia in a medical diagnosis problem. The list name of each attribute list relevant to pneumonia appears on the state list for this disease. There are two data pairs on the DLIST of each state list. The stored attributes are the a priori probability of the state and the print name of the state. The latter is the name by which the user of the program makes reference to the state. In order to facilitate the retrieval of the state list corresponding to a particular print name (as, for example, when the user makes a request for information about the disease pneumonia), all the state lists are grouped on a number of hash lists. Each hash list is a sublist of a list called the master state list. The retrieval of the state list corresponding to a particular print name is effected as follows: First a SLIP function is used to map the given print name onto the integers 0 to N-1, where N is the number of hash lists on the master state list. If the integer K-1 results from this mapping, the Kth hash list is searched for a state list with the desired print name. Since the same hashing function is employed in the creation of the master state list, the appropriate list will be found if one exists. Roughly speaking, this technique reduces the average search time for such requests by a factor of 1/N as compared to a search in the absence of hash lists.

An attribute list includes the list names of all the test lists corresponding to tests which can result in the given attribute.
Figure 6

A Sample State List
The DLIST for an attribute list contains a data pair for the attribute print name in addition to a special data pair for a member list. The member list for an attribute list is a standard SLIF list which contains the list name of each state list on which the name of the attribute list appears and the corresponding probability of the attribute given the state. Continuing the example above, Figure 7 depicts the attribute list for the attribute "fever." As in the case of the state lists, each attribute list is a sublist of a hash list, and each of these hash lists, in turn, is a sublist of the master attribute list.

A test list contains the cost of the test and a DLIST. The DLIST contains the print name for the test and a member list for the attribute lists which include this test. In Figure 8 a simple test list is shown with a single cost (independent of state) and a deterministic member list. This is the form of test list used in this research although it would be relatively easy to make it more complex. As above, each test list is a sublist of a hash list, which is in turn a sublist of the master test list. A schematic of a portion of the information structure is shown in Figure 9.

The presence of two-way links between attributes and states and attributes and tests results in a highly associative information structure. This associative property facilitates the accessing of information pertinent to a diagnosis. Thus a search for attributes given state and a search for states given attribute are equally efficient. Similarly the accessing of possible attributes resulting from a
Figure 7

A Sample Attribute List
<table>
<thead>
<tr>
<th>Address of Cell</th>
<th>Contents of Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>30122</td>
<td>30124 30124</td>
</tr>
<tr>
<td></td>
<td>30126 5</td>
</tr>
<tr>
<td>30124</td>
<td>30122 30122</td>
</tr>
<tr>
<td></td>
<td>100.</td>
</tr>
<tr>
<td>30126</td>
<td>30136 30130</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>30130</td>
<td>30126 30132</td>
</tr>
<tr>
<td></td>
<td>PNAME</td>
</tr>
<tr>
<td>30132</td>
<td>30130 30134</td>
</tr>
<tr>
<td></td>
<td>XRAY</td>
</tr>
<tr>
<td>30134</td>
<td>30132 30136</td>
</tr>
<tr>
<td></td>
<td>MEMBER</td>
</tr>
<tr>
<td>30136</td>
<td>30134 30126</td>
</tr>
<tr>
<td></td>
<td>30571 30571</td>
</tr>
</tbody>
</table>

**Figure 8**

A Sample Test List
Figure 9
A Portion of an Information Structure
particular test is made straightforward by the presence of the member list.

One example of the importance of this associative aspect of the information structure is its use by the diagnostic program in the initial "pruning" of the space of possible diagnoses in response to the observation of initial attributes. Generally, these initial attributes are presented as the user's statement of the problem. For the program to operate in a reasonably efficient manner, it must use this initial statement of the problem to develop a drastically reduced set of states for further consideration. This is directly analogous to the "pruning" employed by a doctor when upon the observation of a few initial signs or symptoms, he reduces the list of diseases he considers as possible causes of the problem to a very small number relative to the set of all diseases. The diagnostic program would employ the member list for a given attribute list to rapidly determine the set of all diseases which were known to exhibit the corresponding attribute. While this reduction of the search space is crucial to the success of the program, it must not be irreversible if the program is not to be led astray by spurious information or noise. Since it is unreasonable to expect that those who prepare the information structure can anticipate all variations in attribute patterns for a given state, it is expected that the program at times will be confronted with problems involving attributes which are not relevant to the principal problem. The strategies employed by the pro-
gram and the nature of the information structure have a strong effect on the program capability in such a problem environment.

The information structure currently employed by the diagnostic program associates with each state only those attributes which are relevant in the diagnosis of that state. Thus there would be no association between the state "tuberculosis" and the attribute "sore thumb" in the information structure for medicine.\(^1\) The advantage of this is that the size of the information structure is limited. Thus while there may be many attributes, only a subset is associated with any state. As will be discussed later, this creates problems in performing diagnosis in a noisy environment. Certain routines associated with the diagnostic program are responsible for making decisions about the significance of the attributes observed in a diagnosis. The function of these routines is also the subject of a later section.

The discussion of the information structure to this point has implied that the attributes for a given state are taken to be independent. Since in many cases the assumption of attribute independence is not justified, it is necessary that inter-attribute dependencies be representable in the structure. This capability is available in the current program through the use of clustering routine, the

\(^1\)Since the program does not determine what information is included in the structure, the user can associate any attributes and states. The point is that certain associations are not expected,
relation-definition routine, and the relation interpreter.

In order to provide a general capability for dealing with inter-attribute dependencies, the diagnostic program must be able to cope with a variety of relationships among attributes. The important relationships most likely vary from one diagnostic problem area to another. It does not seem advisable to attempt to catalog these relationships within the program itself, since it is extremely difficult to predict just which relationships will be required. Also, if the relationships are incorporated within the program itself, it is difficult to introduce new ones as they become of interest in a particular problem area.

What is required then is a flexible facility for the program to accept new relationships and having so accepted a relationship, to incorporate it correctly in the inference process of diagnosis. In an attempt to provide this facility, the diagnostic program provides the user with the means to define a variety of relationships among attributes. A relationship is defined by specifying as a Boolean function the conditions under which the relationship is true. This function is employed by the diagnostic program whenever it is necessary to determine whether the relationship is satisfied for a particular state.

Consider, for example, the case in which it is necessary to account for the time of the appearance of certain attributes of a particular disease. Imagine that for the disease in question the
attribute "rash" appears two days after the appearance of "fever."

Let the function BEFORE accept five arguments and be defined as

\[
\text{BEFORE (A1, A2, A3, A4, A5)} = \\
\text{(EQ (MINUS (CHAR A1 A2) (CHAR A3 A4)) A5)}
\]

Here EQ, MINUS, and CHAR are system primitives (defined by the
diagnostic program). The function CHAR is used to retrieve characteristics of attributes. For example, the value of

\[
\text{(CHAR TIME FEVER)}
\]

is the time at which the attribute fever was observed.

By specializing the function BEFORE as

\[
\text{BEFORE (TIME, RASH, TIME, FEVER, 2)}
\]

The relationship for the disease in question can be checked.

Such relationships are defined by the DEFINE subroutine which
the user can invoke as required. Relationships can also be built
into the information structure when it is first established if they
are known to be necessary. To define a relationship among the at-
tributes of a particular state, one uses the CLUSTER routine. This
routine re-organizes the state list for the state involved, producing
an attribute-cluster. Thus, for the example above, the reorganized
state list might look as that in Figure 6. As with individual at-
tributes, a conditional probability given state is associated with
each attribute cluster.
Figure 10
State List with Cluster
Any number of relationships can be defined for the structure provided that they can be expressed in the prescribed manner. Complex relationships can be specified by using functions of functions. Note that attributes remain independent for any state unless a relationship involving them is defined for that particular state. Thus, in one disease "fever" and "rash" may be related in some way, while in another they may be independent.

The diagnostic program employs an interpreter to determine the truth of relationships during diagnosis. The interpreter permits the correct incorporation of relationships in the diagnostic inference. The manner in which the interpreter is employed will be examined in detail later.

B. THE DIAGNOSTIC PROGRAM

The diagnostic program and its associated routines are the heart of the system. These programs embody the various diagnostic strategies employed by the system. When one uses the system in the solution of a diagnostic problem, he interacts with the diagnostic program. This program provides the interface between the user and the facilities of the system. There are three basic functions performed by the diagnostic program. (Although, in fact, each of these functions is delegated to a set of subroutines, it is convenient to consider them as logical functions of the diagnostic program.) In brief these three functions are:

1) The interpretation of the attributes of a particular
problem based on the information contained in the information structure. This function is called the inference function.

2) The selection of tests for the user to apply to the system being diagnosed in order to obtain further clues as to the system state. This is the test selection function.

3) The analysis of the attributes of a problem to determine whether there are irrelevant attributes present or to detect attribute patterns from more than one system state occurring simultaneously. This is the pattern-sorting function.

The design of the diagnostic program permits the alteration or replacement of any of these three functions independently of any of the others. This flexibility is important, because these functions are fundamental to this scheme for diagnosis, and it is necessary to study different versions of the functions. The possibility of changing individual functions without changing the remainder of the program greatly facilitates this study.

Before the diagnostic program can be used in a particular problem area, an information structure for that area must be established. This requires that a disk file containing all the relevant information be created. The disk file can be created using the standard input and editing facilities of the time-sharing.
formatting of the file, although specified, is quite simple, and if
the necessary information is available, the only difficulty in
creating the file is dealing with the large amount of information
which may be required. The information in the file consists of state-
attribute relationships and test cost data. An example of a portion
of such an input file is shown in Appendix 1. A system program
processes the input file and from it constructs the information
structure for the problem area.

A second file containing the loss structure for the problem
area is required by the diagnostic program. At present this loss
structure is always a matrix. Any element of this matrix, \( l_{ij} \), is
the estimate of the loss for diagnosing state \( j \) as state \( i \). The
exact manner in which this information is employed will be made
clear below.

As a preface to the discussion of the logical functions of
the diagnostic program, consider this example of a particular
application of the program. Suppose the program currently is set
up to diagnose a certain group of diseases. This means that the
appropriate information structure and loss structure have been es-
established. A user wishing to invoke the assistance of the program
does so by providing an initial problem statement. This statement
is essentially a list of the attributes which have been observed.
Assume for the example that this list is

- temperature of 102\textdegree
- severe coughing
- sore right ankle
As indicated in Figure1, the program first invokes the pattern sorting function for the current attributes. In this case, the pattern sorting function hypothesizes that the attribute "sore right ankle" is not relevant to the principal medical problem of the patient, and so removes it from the list of attributes for later investigation. After the attributes have been processed by the pattern sorting function, the set of all diseases which exhibit the relevant attributes is obtained and a probability distribution for diseases given these attributes and the "experience" in the information structure is created. The creation of this probability distribution is the task of the inference function. This distribution results from a consideration of both the current attributes and the knowledge of the various diseases. It is the current view of the diagnostic problem assumed by the program.

Now the program invokes the test selection function. The object of this function is to select a good test for the user to apply to the patient in order to gain more information. In selecting this test, the test selection function considers the current probabilities of the various diseases, the cost of each test, and the usefulness of the results expected from the test. The user is informed of the test which has been selected. The test may be as simple as asking the patient questions about his recent exposure to other sick persons, or it may be more involved, for example, a chest X-ray. In any event, when the user has obtained the results of the
Figure 11

Flow of Diagnostic Program
test, he reports them to the program. These test results are new attributes and the program again enters the loop shown in Figure 11. This dialogue with the user continues until a diagnosis has been obtained. A more detailed trace of a session with the diagnostic program is presented in Appendix 2. This brief example provides an overview of the operation of the diagnostic program. In what follows, each of the primary functions of the program will be discussed in detail.

1. THE PATTERN-SORTING FUNCTION

As explained in an earlier section, only those attributes significant to the diagnosis of a particular state are associated with that state in the information structure. Thus the attribute "sore ankle" would not be associated with the disease tuberculosis in the information structure; this means that the name of the attribute list for the attribute "sore ankle" does not appear on the state list for the disease "tuberculosis". Similarly the member list of the attribute list for "sore ankle" contains no entry for the state list of tuberculosis. If the name of a state list does not appear on the member list of a given attribute list, then the conditional probability of the attribute given the state is taken to be zero by the program. As will be discussed in the following section, the particular method of deduction employed by the program (Bayes' rule) results in a zero posterior probability for the state given the attribute. For instance,
if in the course of a diagnosis in which tuberculosis was considered a possible cause of the attributes the attribute "sore ankle" were observed, the updated probability of tuberculosis would be zero. Since the program removes from current consideration any state with zero probability, this approach makes maximum use of each attribute to reduce the set of possible diagnoses.

The problem encountered here is that while "sore ankle" is not an attribute of tuberculosis, one certainly can have tuberculosis and a sore ankle. This is but one example of the more general problem of irrelevant or noise attributes. Unless special precautions are taken, such attributes can eliminate the actual state from consideration when processed by the inference function. A number of solutions to this problem are possible.

One approach is to associate every attribute with every state, employing $\epsilon$ probabilities whenever an attribute is not considered relevant to the diagnosis of a particular state.\(^1\) As long as $\epsilon$ is greater than zero, no state will be eliminated from consideration in the manner described above. The difficulty is that this method prevents the drastic reduction in the set of possible diagnoses which is necessary for efficient operation of the program. A second approach is to employ the $\epsilon$ probabilities as above, but to eliminate

\(^1\)This probability might be taken to be the unconditional probability of the attribute. Since this probability may be quite small, the problem discussed here could still be encountered.
from further consideration those states whose posterior probability falls below a fixed threshold. This method is unsatisfactory because the posterior probabilities for the various states can undergo radical change as additional attributes are observed and employed by the inference function. Thus, there is no guarantee that a state with a very low probability in the early stages of the diagnosis will remain improbable with the observation of new attributes. This problem can be even more severe if the noise attributes are the first observed. In either event, the actual state may be removed from further consideration by this method. Another approach is to decide whether an attribute is relevant to the diagnosis or merely noise before it is processed by the inference routines. This is a very difficult task to accomplish given the particular model employed in diagnosis by the program. The model of the system being diagnosed consists principally of state-attribute relationships without any information about causal connections. Thus, the only way to evaluate the relevance of an attribute to the diagnosis is to consider some measure of its probability given the diagnosis to date. Since almost every measure of this kind depends on the current prior distribution, which, in turn, depends on the observed attributes assumed to be relevant, a cyclical argument results.

A second problem arises when attributes characteristic of two or more distinct states are observed, as in the case of an individual with more than one disease. This is more than a problem of simple
noise since the program must detect two or more patterns. Again
the methods mentioned above are inadequate to cope with this problem.

The solution to this problem which has been incorporated in
the diagnostic program involves processing a number of attribute
patterns in parallel during a diagnosis. A pattern is a subset of the
set of observed attributes which has the following two properties:
1) At least one state in the information structure exhibits all the
attributes in the pattern with a non-zero probability and 2) The
pattern is not a subset of any other pattern. If the set of observed
attributes contained a number of the attributes of tuberculosis and
the attribute sore ankle, one pattern would be the set of tuberculosis
attributes. A second pattern would be obtained by choosing a disease
for which sore ankle is an attribute and taking the intersection of
the set of attributes for that disease and the set of observed at-
tributes. Perhaps the set of attributes obtained in this way, using
a second disease on the member list of "sore ankle," might be dif-
f erent from both those previously obtained. If so, this set is still
another pattern.

Throughout the course of a diagnosis, a pattern stack is main-
tained by the pattern-sorting function. A schematic of the pattern
stack is presented in Figure 12. Each pattern is represented by a
sublist of the pattern stack, and associated with each pattern is
the probability distribution for the states of the system given the
attributes of the pattern.
Figure 12

Pattern Stack
Whenever a new attribute is obtained in a diagnosis, it is processed against every pattern in the pattern stack. The new attribute is used to update a pattern if it is relevant to at least one state in the probability distribution for the pattern. After this updating, the attribute is added to the pattern. If no state in the probability distribution of a pattern is known to exhibit the new attribute, no changes are made to either the pattern or the distribution. The actual manner in which distributions are updated to account for new attributes is discussed in detail in the next section on the inference function.

When the new attribute has been processed against all patterns, a routine called PATFRM is invoked to form new patterns if possible. PATFRM retrieves the member list of the attribute list corresponding to the new attribute. For each state on the member list, the set of probability distributions in the pattern stack is searched. If the state is found in this set, the pattern for the state is already in the pattern stack. If the state is not found, the intersection of the set of attributes denoted by the appropriate state list and the set of observed attributes is a new pattern. This pattern and the corresponding distribution for the states is added to the pattern stack. While it is conceivable that this procedure could generate many patterns for a given information structure and attribute sequence, this is not a serious problem. First in most areas the number of distinct patterns which can be formed by this procedure for a
given attribute set is quite limited, because states exist in
groups which have overlapping attribute patterns. Secondly, the
number of patterns considered can be limited by considering only
those patterns with a probability greater than some threshold.

This procedure also includes a provision for removing patterns
from the stack. If the inference function determines that the
probability of a particular pattern is zero, the pattern and its
associated distribution is eliminated from the pattern stack. The
contents of the pattern stack, then, can be quite dynamic during
the course of a diagnosis as new attributes trigger the addition
and deletion of patterns.

As an illustration of this aspect of the pattern sorting
function, consider the following example. At a given stage in a
diagnosis of a medical problem, three attributes have been observed.
These attributes are A, B and C. Also assume that of the diseases
represented in the information structure, none exhibits all three
of these attributes. A number of diseases exhibit A and B as at-
tributes, however, and so this is a pattern. The point here is that
while a disease which exhibits A and B can occur with C also present,
C is not considered relevant to the diagnosis of any of these
diseases. For the diseases for which C is a relevant attribute
A is also relevant. For this situation the pattern stack can be
represented as in Figure 13A. Here the symbol \( \exists \) denotes the dis-
tribution list for a pattern.
Now when the new attribute D is observed, it is processed through the pattern stack. Assuming that the new attribute is relevant to some of the states in distribution \( \Pi_1 \), this distribution is updated by the inference function to produce \( \Pi'_1 \) and the attribute D is added to the pattern. Attribute D is not relevant to the second pattern in the stack, and so this pattern and its associated distribution remain unchanged. Finally, the routine PATFORM is invoked to search for new patterns. Assume that no new patterns are formed. Thus, at the end of this phase in the processing of the new attribute, the pattern stack appears as in Figure 13B.

Now in the event that there is more than one pattern in the stack, the diagnostic program must make a decision as to which pattern to diagnose. Thus, the program must generate a hypothesis about the significance of the various patterns in the stack. For example, if one pattern corresponds to a majority of the attributes of tuberculosis, and the other to a single attribute "sore ankle," it is extremely important for the program to give priority to the former pattern. The problem is to establish pattern selection rules which will make the "correct" decision in such a case.

One consideration which is relevant to the selection of a pattern is the seriousness of the states suggested by the pattern. For this reason, an attribute quite specific to a very serious disease will strongly influence the course of a medical diagnosis.

In order to account for the relative seriousness of different
and $\pi_j$ = a priori probability of state $j$.

Values of $\Theta$ decrease with increasing seriousness of states. This can be seen in the following simple example.

<table>
<thead>
<tr>
<th>LOSS</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Benign tumor</td>
<td>0.7</td>
<td>0</td>
<td>1,000,000</td>
</tr>
<tr>
<td>2. Malignant tumor</td>
<td>0.3</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

While other more sophisticated measures of seriousness can be developed, this simple one was deemed suitable for the purposes of this research.

Once the seriousness of the various states has been established, the problem of pattern selection can be solved in a quite reasonable way through the use of the Bayesian model. For each pattern, a conditional distribution on states can be obtained by the inference function. For each pattern, the distribution is conditioned on the attributes of that pattern alone--all other patterns are ignored. Thus for the $k^{th}$ pattern

$$\pi_j^k = \frac{\pi_j^k P(S_{1k}, \ldots, S_{nk} | X_j, \Theta)}{P(S_{1k}, \ldots, S_{nk} | \Theta)}$$

Where $\pi_j^k$ is the conditional probability of the $j^{th}$ state ($M_j$) given the pattern $\{S_{1k} \ldots S_{nk}\}$.

The seriousness measure for the $k^{th}$ pattern is given by

$$\gamma_k = \sum_{j=1}^{n} \pi_j^k \theta_j$$
and the pattern selected is the one with minimum $\gamma$.

This measure has several desirable properties. Consider the case of an attribute which is very specific to a serious disease. If that attribute is observed, the conditional probability for the serious disease given the pattern containing the attribute will be close to one. Since the corresponding value of $\theta$ is small, the value of $\gamma$ for the pattern will be small. Hence this pattern will quite likely be selected. On the other hand, if the attribute is not specific to the serious disease, the conditional probability for the disease given the pattern will be less; and the resulting value of $\gamma$, greater.

The measure also favors a pattern which contains many attributes provided that the pattern strongly indicated one or more serious states. The posterior distribution does not have to be spiked, however, for a pattern to be chosen. For example a pattern which results in equal probabilities for six states may also be chosen if the seriousness of the individual states so warrants. This measure accounts for both the specificity of a pattern and the seriousness of states associated with the pattern. In this respect, it seems to be a good way to select patterns for investigation.

A routine called SELECT chooses the current pattern for the diagnostic program, and this pattern may change from time to time as additional information is gathered by the program. The current pattern is the one employed by the test selection function for
evaluating tests. Before each use of the test selection function, SELECT chooses the current pattern based on all information currently available.

A number of other processing routines affect the pattern stack during the course of a diagnosis. Recall that whenever the pattern sorting function produces more than one pattern in the stack, the selection of a pattern for further diagnosis constitutes a hypothesis about the significance of a group of attributes. If a consistent diagnosis for the current pattern is obtained, then the hypothesis is tentatively confirmed. If there are no other attributes to account for them a consistent diagnosis for all attributes has been obtained. Otherwise the remaining patterns must be considered. It is possible that a second pattern is being diagnosed, new attributes may prove the hypothesis about the first pattern to be incorrect. In this case, the attributes in this pattern can no longer be considered accounted for. These possibilities are dealt with in the following way by the pattern sorting function. The program maintains a list called the "unaccounted-for" list, and on it are all those attributes which have yet to be attributed to a particular system state. When the current pattern is "diagnosed" or assigned to one state, the attributes in the pattern are removed from the unaccounted-for list, and the pattern itself is marked. A marked pattern is ignored in test evaluation, although it is updated with new attributes whenever appropriate. When the current pattern has been marked, all unmarked patterns are deleted from the stack. Then PATRN is called
for each attribute in the unaccounted-for list. Patterns are formed
using the unaccounted-for list as the total attribute set. If the
unaccounted-for list is empty, a consistent diagnosis for all attrib-
utes has been obtained. Otherwise, the diagnosis continues on the new
patterns.

This means that attributes which are included in marked patterns
are not utilized in the formation of new patterns at this time. If,
for example, the total attribute set were (A, B, C, D) and (A, B, D)
had been tentatively diagnosed, the only unmarked pattern would be
(C). This is true even though there may be states which exhibit both
C and A. If, however, the test selection function chooses a test
which can detect A, A will be added to the unmarked pattern. This is
because the program always consults the history of the diagnosis be-
fore requesting the user to run a test. If on the other hand, the
program would normally account for C without employing knowledge of
A, it will do so.

If a new attribute causes the probability of a marked pattern to
become zero, a special recovery procedure is invoked. First, each
attribute of the marked pattern is transferred to the unaccounted-for
list. If one of these attributes is added to the list, it is also
processed against all the other patterns in the stack. When the stack
has been updated with such an attribute, PATTEM is invoked to check
for new patterns based on this attribute. Finally, the marked pattern
is deleted from the pattern stack, and diagnosis continued.
Thus, the contents of the pattern stack may be quite volatile during a diagnosis, although cases of extreme volatility are not expected to occur very often. In any event, the use of the pattern stack permits the program to deal with noise and multiple patterns in a reasonably efficient manner. By allowing the user to interact with the program during diagnosis, it is possible to employ his judgment with regard to the merits of pursuing particular patterns.

2. THE INFERENCE FUNCTION

In general, the observation of a new attribute provides the diagnostic program with additional information about the current state of the system being diagnosed. Based on this observation, the program may significantly alter its estimate of the likelihoods of the various states. This section discusses in detail the manner in which the program incorporates observations of attributes into its current view of the diagnostic problem. The routines which process new attributes for their effect on the current view of the problem collectively are called the inference function.

The basic analysis of attributes and inference done by the diagnostic program is based on Bayes rule. Bayes rule can be stated as follows

\[ P(M_j|S_t, \mathcal{E}) = \frac{P(M_j|\mathcal{E})P(S_t|M_j, \mathcal{E})}{P(S_t|\mathcal{E})} \]

where \( P(M_j|\mathcal{E}) \) is the probability that the current state is \( M_j \).
conditional on the total experience to date.

\[ P(S_t/M_j, \mathcal{E}) \] is the probability that the system will exhibit attribute \( S_t \) given that it is in state \( M_j \) and the diagnostic experience \( \mathcal{E} \).

\[ P(S_t/\mathcal{E}) \] is the probability of the system exhibiting \( S_t \) unconditional on state.

\[ P(M_j/S_t, \mathcal{E}) \] is the conditional probability that the state of the system is \( M_j \) \textit{given} \( \mathcal{E} \) and the newly observed attribute \( S_t \).

The quantity \( P(M_j/\mathcal{E}) \) is called the \textit{prior} probability and \( P(M_j/S_t, \mathcal{E}) \) is called the \textit{posterior} probability of the state \( M_j \). The observation of the attribute \( S_t \) increases the experience or information available on which to make a decision about the unknown state. The posterior probability is an adjustment of the prior probability to account for the new information. After this adjustment has been made, the posterior probability is the new prior probability when further attributes are observed. Consider the following example of this basic inferential process:

Suppose there are only two states relevant to the current diagnostic problem, \( M_1 \) and \( M_2 \), and three attributes \( S_1 \), \( S_2 \) and \( S_3 \). The \textit{a priori} probabilities for the two states as well as the conditional probabilities for the attributes given the states are presented in Table 2.
TABLE 2
EXAMPLE FOR BAYESIAN ANALYSIS

<table>
<thead>
<tr>
<th>A priori probability</th>
<th>S(_1)</th>
<th>S(_2)</th>
<th>S(_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(_1)</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>M(_2)</td>
<td>0.2</td>
<td>0.7</td>
<td>0.6</td>
</tr>
</tbody>
</table>

The initial experience of the program, before any attributes have been observed, is embodied in the a priori probabilities. Thus, the current distribution on states is (0.8, 0.2). Now assume that tests employed in the diagnosis reveal the presence of attribute S\(_1\). According to Bayes rule, the posterior distribution is (0.82, 0.18).

That is

\[
P(M_1/S_1, \mathcal{E}) = \frac{(0.8)(0.8)}{(0.8)(0.8) + (0.2)(0.7)} = 0.82
\]

\[
P(M_2/S_1, \mathcal{E}) = \frac{(0.2)(0.7)}{(0.8)(0.8) + (0.2)(0.7)} = 0.18
\]

Thus, the new attribute has little effect on the view of the problem taken by the program. If two more tests yield the attribute S\(_2\) and then the attribute S\(_3\), the corresponding distributions are:

\[
P(M_1/S_1, S_2, \mathcal{E}) = 0.75 \quad P(M_2/S_1, S_2, \mathcal{E}) = 0.25
\]
tern and its distribution list are removed from the stack. While Bayes rule is easily applied in principle, the inference function must include special routines to insure that inter-attribute relationships and the "history" of the diagnosis are correctly accounted for in the probabilistic analysis.

The routine UPD which performs the updating of the pattern stack based on the observation of a new attribute is to a large extent a simple encoding of Bayes rule. The routine, however, does not obtain the requisite conditional probabilities directly. Instead, it calls PIJ to obtain the conditional probability of attribute "j" given state "i" and the history of the diagnosis to date. The reason for this indirection in the accessing of probabilities is really a pragmatic one. The insulation UPD from the probability-retrieving process allows changes in this process to be made without affecting the basic inference process.

As noted, the function of PIJ is to retrieve conditional probabilities from the information structure. In the simplest case, this involves retrieving a number directly from the information structure. When the attribute of interest is involved in an attribute cluster for the given state, the process of determining the conditional probability is more involved.

The general form of an attribute cluster is either

a. \((S_1 R_1)\)

or b. \((S_1 R_1 \otimes S_2 R_2 \otimes \ldots \otimes S_n R_n)\)
where $R_j$ is an inter-attribute relationship;

$\Theta_j$ is the conditional probability of $R_j$ given the state

$\Theta$ is either "exclusive or" or "or."

Here $R_j$ can be any inter-attribute relationships (including functions of functions, etc.) as long as it does not include $\Theta$. The reason for this restriction is to eliminate ambiguity from the probability assignments. In fact, the restriction does not limit the class of logical relationships which can be defined, only the form which individual members may assume. Thus, for example, $R_j$ might be the cluster for the relationship

"Either $A_1$ precedes $A_2$ in time or $A_1$ does not appear at all."

In order to evaluate the conditional probability of an attribute involved in an attribute cluster, PLJ must be able to evaluate the truth of the relationships $R_j$. It does this by calling the routine INTERP to determine the true value of each $R_j$. INTERP is an interpreter, which retrieves the definitions of any functions involved in $R_j$ and applies these definitions to the appropriate arguments from the attribute cluster. The interpreter employs a push-down stack and recursive calls in the evaluation. All functions are reduced in this way to their component primitive functions. Routines to evaluate the primitive functions are built into the system.

The operation of the interpreter differs in certain aspects from that of a normal interpreter of Boolean functions, because this interpreter must deal with variables whose current value is unknown.
For example, suppose the relationship under consideration for a particular state $M_j$ is "$A_1$ precedes $A_2$ in time" with probability 0.5. Assume $A_1$ has just been observed and the conditional probability of $A_1$ given the state is desired. If $A_2$ has not yet been observed, the relationship is incomplete (or from a logical standpoint, undefined). From a Bayesian point of view, however, the conditional probability is well-defined; it can be obtained by assuming that $A_2$ will in fact follow $A_1$ in time. This assumption results in a value of 0.5 for the conditional probability of $A_1$ given $M_j$. If $A_2$ is observed later, then its conditional probability can be obtained in a similar manner, but the prior observation of $A$ must be taken into account. This means that the desired probability of $A_2$ is conditional on the state $M_j$ and the previously observed $A_1$. Hence the proper conditional probability is 1.0.

In general terms, the interpreter assumes the truth of any relationship which is incomplete unless that relationship is demonstrably false given the current information of the diagnosis. The interpreter must also indicate whether any attributes involved in a cluster have actually been observed. Given these modifications of the interpreter function, the routine PIJ can deduce the proper conditional probability for the given attribute-state pair. PIJ embodies a number of logical tests on the truth of the $R_{ij}$ and the number of observed attributes involved in each. For the types of relationships allowed in the information structure, these quantities are sufficient to deter-
the relevant attributes are first presented by the user. Through
the use of the interpreter, the diagnostic program is able to deal
with variety of relationships within a particular problem area.

3. THE TEST SELECTION FUNCTION

The value of heuristics for test selection in diagnostic prob-
lems has been underscored in previous sections. In this section, a
particular test selection program is discussed. This program (which
is, in fact, a number of subroutines) is the one employed in the diag-
nostic program. The nature of the program strategy and organization
is explained and some of its limitations are noted.

From the model of a diagnostic problem discussed in Chapter 3,
it will be recalled that one of the major tasks in diagnosis is the
selection of a good set of tests to apply to the system. The de-
termination of such a testing strategy involves a consideration of
both the costs of tests and the information which they are expected to
yield. Thus, any heuristic for the test selection process should re-
reflect these considerations. Another consideration involves the amount
of computation involved in applying the heuristic in a particular
diagnosis. In order to facilitate the study of a class of such test
selection heuristics, the test selection function was designed to be
in large part independent of the particular heuristics employed. While
the class of heuristics permitted is not particularly large, it does
include heuristics which lead to markedly different test selection
searched by the test selection function during a particular stage in a diagnosis are searched to the same depth.\textsuperscript{1} The limitations arising from this inflexibility will be discussed later.

The breadth of the search is controlled indirectly by the user through the use of a threshold probability. At a given decision node, only those tests which are relevant to a state with a probability greater than the threshold are considered by the test selection function. For example, if the probability distribution at a given decision node is \((0.2, 0.3, 0.5)\) for states \(M_1, M_2, M_3\) and the threshold is 0.25, only those tests relevant to states \(M_2\) and to \(M_3\) will be considered. Those tests which are relevant to \(M_1\) alone will be ignored. A test is considered relevant to a particular state only if an attribute which is associated with the appropriate state list in the information structure is a possible result of the test given the probability distribution for the current decision node. Since the control of the breadth of search is indirect, in general, the user cannot easily predict the extent of the pruning of the decision tree which will result. Some feeling for reduction in the search space can be gained from experience with the program in a particular problem area. Note that in the above example, if all the tests which are relevant to

\textsuperscript{1}An exception occurs when a particular node corresponds to a certain diagnosis. The search of the branch containing this node will terminate there.
state \( M_1 \) are also relevant to either \( M_2 \) or \( M_3 \), then the threshold probability will not result in any pruning of the decision tree. The maximum search breadth is obtained with a threshold of zero.

Like the search depth parameter, the threshold parameter can be set prior to each stage in the diagnosis. Also, these two parameters can be varied independently of one another (subject only to a practical constraint of available storage). This flexibility permits the overall selection strategy to change during the course of the diagnosis.

There are four routines in the test selection package, each performing a distinct function in the tree search. The principal routine is SEQDEC which serves as the main control for the process of test selection. The diagnostic program communicates with the test selection package through SEQDEC. It provides this routine the name of the node in the decision tree which corresponds to the current state of the diagnosis. SEQDEC then analyzes the tree to the appropriate depth and breadth to obtain the testing decision.

Because the decision tree can require considerable storage even for limited search depth and breadth, the tree is developed dynamically. That is, new levels are added only as they are needed, and levels are erased when they have been analyzed. SEQDEC is called with the name of a decision node as an argument. This decision node is represented by an empty SLIP list which has on its DLIST a list containing a probability distribution over system states. This distribution incorporates all the attributes which were observed on the
path from the beginning of the tree to the current node.

SEQDEC first determines the expected loss for an optimal decision at this node. The manner in which this value is determined will be explained below. If the level of the current node equals the required depth of search this expected loss is returned as the expected loss for the node. If not, the current loss for this node is assigned this value and if the level of the node is the topmost level of the analysis, the terminal decision and its value are stored in a special list. In any event an additional level must be "grown" on the tree. First the routine RELTST is called by SEQDEC. RELTST determines the set of tests which are relevant to the states whose probability at the current node exceeds the threshold. Excluded from this set are all those tests which have been actually run. These latter tests are known to RELTST because whenever a test is selected by the diagnostic program and run by the user, its name is placed on a list called TSTRUN in common storage. RELTST stores the names of the relevant tests on the current decision node list.

After RELTST has collected the set of relevant tests, SEQDEC processes each of these tests in turn. SEQDEC begins reading the list of tests. For each test, a routine called GROW1 is invoked. This routine determines all possible results of the given test and their respective probabilities. For each result, the routine constructs a new decision node. First the current test is placed on the top of TSTRUN to simulate the running of the test and then for each
of the possible results of the test, SEQDEC calls itself recursively to obtain the expected loss of the resulting decision node. When this value has been obtained, it is weighted by the probability of the given result and the product accumulated. The sum of the expected loss for each result is combined with the cost of the test. The current test is removed from TSTRUN and the portion of the decision tree which has just been analyzed is erased. If the analysis is at the topmost level the value of the test is saved. This means that the expected losses for all alternatives at the current level are available. In the event that the best alternative cannot be employed (e.g. a test cannot be run for some reason), the next best alternative can be chosen. In any case, the expected loss for this test is compared with that of the best decision to date for the node. If it is less, the current test becomes the best decision. The analysis then proceeds to the next test alternative. When all alternatives have been evaluated for the current decision node, SEQDEC returns the expected loss of the best decision as determined by the analysis.

The determination of the optimal terminal decision as accomplished by a routine called DLOSS. This routine employs the probability distribution, the decision node and the loss function to determine the value of the minimum expected loss terminal decision for the node. If $\pi_j$ is the probability of the state $M_j$ in the current distribution and $l_{ij}$ is a typical element from the loss function matrix, DLOSS selects state $M_k$, where
\[ \overline{E} = \sum_{j=1}^{n} l_{kj} \overline{F}_j = \min_{k} \sum_{j=1}^{n} l_{ij} \overline{F}_j \]

and \( \overline{E} \) is the expected loss of the optimal terminal decision for the node. The state selected by DLOSS and the value \( \overline{E} \) are returned to SEQDEC.

By controlling the breadth and the depth of the search employed by the test selection function, the user can generate a number of different test selection heuristics. For example, he might use a threshold close to zero and a depth of one early in a diagnosis when many states are still possible. Because the probability distribution based on only a few attributes may be quite diffuse, a low threshold is needed to insure that significant tests are not overlooked. On the other hand, the potentially large number of decision nodes requires a limited depth of search. As the diagnosis progresses and a few states become relatively probable, the threshold can be raised with less danger of missing significant tests. With the higher threshold it may be possible to improve the evaluation of tests by increasing the depth of the search.

The selection scheme above can be supplemented by the use of two additional features of the program. First, the user can restrict the set of relevant tests to those associated with the best terminal decision at a given node. In the case when the loss function is a constant for all ordered pairs of states, this corresponds to considering the tests which are relevant to the most probable state.
Since the routine DLOSS can determine the best terminal decision at a given decision, the appropriate state can be made available to RELTST. By considering only the tests relevant to this state, the user in a sense in limiting the search to those tests which will tend to prove or disprove the hypothesis that the given state is indeed the best decision. In practice, the user obtains this option by setting the threshold probability to a number greater than one.

In order to permit the user an even greater facility to test hypotheses, the program permits him to request a search for tests to prove or disprove the hypothesis that "the state of the system is \( M_k \)." If the user chooses to test such a hypothesis, the test selection function will alter its method of evaluating decision nodes. All decision losses \((1\neq j)\) are set temporarily to a certain very high value. The routine DLOSS then considers only two states in its evaluation of the loss for a given node. One state is \( M_k \) and the other is "not \( M_k \)." With these adjustments, the test selection function will rank tests according to their expected value in proving or disproving the presence of state \( M_k \).

A comparison of a number of particular selection heuristics employed in this research will be presented later in the thesis.

C. THE GENERATOR PROGRAM

The diagnostic program discussed in the previous sections is a major tool in this research. By exploiting the interactive capabilities of the program, the user can employ it directly in the solution
of actual diagnostic problems. Of equal importance, however, is the
availability of the program as a test vehicle for a variety of over-
all diagnostic strategies. By specifying the heuristics to be em-
ployed in the pattern sorting and test selection functions, one is
defining a diagnostic strategy. Since diagnostic problems tend to
be difficult and the program operation is quite complicated, it is
not an easy task to make generalizations about a given diagnostic
strategy. There are many important questions which can be asked about
a diagnostic strategy such as

- How is the performance of the program affected by noise signs?
- What is the effect of uncertainty in the probabilities on
  the performance of the program?
- How do various changes in the relevant probability distribu-
  tions affect program performance?

Questions such as these are difficult to answer based on experience with
only a few problem areas. If one is constrained to work with descrip-
tions of actual systems, it may be very difficult to establish the
conditions required for the test of a particular aspect of the pro-
gram. If, on the other hand, one can employ a wide variety of system
descriptions, the program can be exercised more thoroughly. One ap-
proach is to create an information structure with the desired proper-
ties and to test the diagnostic program with simulated problems from
this artificial problem area. Information gained from such studies of
diagnosis "in the abstract" may suggest improvements in the program.
It may also provide a deeper insight into the problems involved in solving real diagnostic problems. If such a simulation facility were available, simulated cases generated from the structure for an actual problem area could be utilized to conveniently investigate aspects of diagnosis in that area.

The diagnostic system includes such a simulation facility in the form of the generator program. This program is the third major part of the diagnostic system. Like the diagnostic program, the generator makes extensive use of the information structure. The system for which problems are to be simulated is described in the standard manner by the user. This description is converted to an information structure which is available to both the diagnostic program and the generator. The basic operation of the generator is as follows. First, a state is chosen at random from the set of possible states for the system in accordance with the a priori probability distribution. Then a certain number of initial attributes (the number being controlled by the user) are generated at random given the description of the state in the information structure. The set of initial attributes constitutes the problem presented to the diagnostic program. The latter is called to process these attributes. It selects a test in the usual manner. Given the state and the test, the generator selects a test result and conveys this response to the diagnostic program. This interaction between the generator and the diagnostic program continues until the latter arrives at a diagno-
sis. This diagnosis then can be compared with the "known" state used by the generator.

As an example of the operation of the generator, consider its use in the following simplified problem. The generator is used to simulate disease case histories for the disease-attribute probability matrix presented in Table III. The relevant tests are listed to the right of the matrix. Assume that cases are to be drawn at random from the structure and that one initial attribute is to be presented to the diagnostic program.

The generator first selects the disease. It does this by creating a list of all possible diseases and cumulative probabilities. For this example, the list would be

\[(D1 0.3 \ D2 1.0)\]

Each cumulative probability is the sum of the a priori probabilities of the diseases preceding it in the list. Then a random number between zero and one is generated. The list of diseases and cumulative probabilities, called the generation list, is searched for a disease with the property that the probability preceding it is less than and the probability following is greater than the given random number. This disease satisfying this condition is chosen for this case. Thus, if the random number generated in the example were 0.41, the disease selected would be D2. Assuming the disease D2 has been chosen, the generator now selects the initial attributes which define
### Table 3

**Disease Description for Generator Example**

<table>
<thead>
<tr>
<th>Disease</th>
<th>a priori Probability</th>
<th>P(Attribute/Disease)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0.3</td>
<td>A1 0.3</td>
<td>A2 0.7</td>
<td>A3 0.5</td>
<td>A4 1.0</td>
<td>A5 0.5</td>
</tr>
<tr>
<td>D2</td>
<td>0.7</td>
<td>A1 0.8</td>
<td>A2 0.2</td>
<td>A3 0.3</td>
<td>A4 0.2</td>
<td>A5 0.6</td>
</tr>
</tbody>
</table>

**Test** | **Attributes**
--- | ---
T1     | A1, A2
T2     | A3
T3     | A4
T4     | A5, A6
with the appropriate probabilities and returns it to the diagnostic program. This iterative process continues until the diagnostic program has completed the diagnosis.

In this example, only one attribute was generated for each test. There are tests, however, from which several attributes can be obtained. Such tests are marked in the information structure, and the generator will generate a set of test results for these tests.

The diagnostic system will record an extensive history of each diagnosis or selected aspects of that history on a history file if requested to do so by the user. A schematic of the relationships among the three major parts of the diagnostic system is presented in Figure 14. In the remainder of this section, certain features of the generator-diagnostic program interaction will be discussed in detail.

The subroutine GETSYM is the principal link between the generator and the diagnostic program. It is this routine which is called by the diagnostic program whenever the latter requires a test to be run. If the diagnostic program is being controlled by the user from the console, then GETSYM retrieves the test results from him. If the generator is in control, a routine called GENSYM is invoked to generate an appropriate response to the chosen test. The diagnostic program itself is independent of the source of responses to tests. GENSYM is also used by the generator to select the initial attributes of a problem. All system output (such as requests for test results, distributions, etc.) is processed by a special output package. This
Figure 14
Schematic of Diagnostic System
another advantage is that many cases can be simulated in a rea-
sonable amount of time.
The second problem encountered in the diagnosis of bone tumors is the large number of potentially useful attributes which can be extracted from a radiograph. Generally speaking, there are four direct kinds of information which are obtained from a radiograph of a bone tumor (R-20):

1) Destruction of bone
2) Proliferation of bone
3) Mineralization of tumor matrix
4) Location, size, and shape of tumor.

Each of these general classes of information is broken down into a number of more specific attributes. The result is the large number of attributes mentioned above. Hence, the diagnostician is confronted with a considerable amount of data which he may employ in classifying a particular tumor.

The particular study discussed here involved the diagnosis of actual cases of bone tumors, each of which was classified into one of nine histological types. These types are listed in Table 4. The evidence employed in the diagnoses consisted of fifty-three attributes obtained principally from radiographs. (The age of the patient was the only non-radiologic attribute considered.) The attributes are listed in Table 5 along with their abbreviations used in discussions of particular diagnoses.

The case histories and the disease-attribute probability matrix used in this study were obtained from Dr. G. S. Lodwick of the
University of Missouri. Dr. Lodwick and his associates developed the matrix as a result of many years experience with cases of bone tumors. Thus, the matrix represents the distillation of extensive diagnostic experience with the problem. It reflects both the statistical experience and understanding of the disease processes involved of the workers who created it. The papers cited above summarize their work and are recommended to any reader who is interested in a more authoritative view of the problem than the competence of this author permits him to present.

B. Experiments in Bone Tumor Diagnosis

The diagnostic system was used to study various aspects of bone tumor diagnoses. The disease-attribute probability matrix provided by Dr. Lodwick was used as the basis for an information structure for the system. A state was defined for each of the nine types of bone tumor. A set of thirty-two tests were defined. Some of these tests such as that of determining the age of the patient can result in one of a number of attributes. In the case of the age test, the possible attributes are: 1) age 0 to 9 years, 2) age 10 to 19 years, 3) age 20 to 29 years, 4) age 30 to 39 years, and 5) age 40 years and over. Other tests are specific for one attribute, such as the test of checking for geographic destruction of bone. The set of tests and the respective attributes which may result is presented in Table 6. Throughout the remainder of this
TABLE 4

HISTOLOGICAL TYPES FOR BONE TUMOR DIAGNOSIS

<table>
<thead>
<tr>
<th>Type</th>
<th>Abbreviation</th>
<th>Relative Incidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Chondroblastoma</td>
<td>CB</td>
<td>0.05</td>
</tr>
<tr>
<td>2. Chondrosarcoma</td>
<td>CS</td>
<td>0.17</td>
</tr>
<tr>
<td>3. Ewing's Sarcoma</td>
<td>ES</td>
<td>0.15</td>
</tr>
<tr>
<td>4. Fibrosarcoma</td>
<td>FS</td>
<td>0.10</td>
</tr>
<tr>
<td>5. Giant Cell Tumor</td>
<td>GC</td>
<td>0.15</td>
</tr>
<tr>
<td>6. Osteosarcoma</td>
<td>OS</td>
<td>0.25</td>
</tr>
<tr>
<td>7. Parosteal Sarcoma</td>
<td>PS</td>
<td>0.05</td>
</tr>
<tr>
<td>8. Reticulum Cell Sarcoma</td>
<td>RC</td>
<td>0.05</td>
</tr>
<tr>
<td>9. Chondromyxoid Fibroma</td>
<td>CF</td>
<td>0.03</td>
</tr>
</tbody>
</table>

1.00

Note: This formulation assumes that each patient has one and only one of the given diseases.
### ATTRIBUTES FOR BONE TUMOR DIAGNOSIS

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>002</td>
<td>Age 00-09 years</td>
<td>010</td>
<td>Tumor Size 51-90 Millimeters</td>
</tr>
<tr>
<td>003</td>
<td>Age 10-19 years</td>
<td>011</td>
<td>Tumor Size 91 and over</td>
</tr>
<tr>
<td>004</td>
<td>Age 20-29 years</td>
<td>012</td>
<td>Shape-Round (L LT 1.5 X W)</td>
</tr>
<tr>
<td>005</td>
<td>Age 30-39 years</td>
<td>013</td>
<td>Shape-Elongated (L GE 1.5 X W)</td>
</tr>
<tr>
<td>006</td>
<td>Age 40 years and over</td>
<td>014</td>
<td>Location-Central</td>
</tr>
<tr>
<td>007</td>
<td>Tumor Size 01-30 Millimeters</td>
<td>015</td>
<td>Location-Cortical/Paraosteal</td>
</tr>
<tr>
<td>008</td>
<td>Tumor Size 31-60 Millimeters</td>
<td>016</td>
<td>Location-Cortical</td>
</tr>
<tr>
<td>009</td>
<td>Tumor Size 61-90 Millimeters</td>
<td>017</td>
<td>Fat Bone</td>
</tr>
<tr>
<td>010</td>
<td>Tumor Size 91 MM and over</td>
<td>018</td>
<td>Small Bone</td>
</tr>
<tr>
<td>011</td>
<td>Shape-Round (L LT 1.5 X W)</td>
<td>019</td>
<td>Sacrum and Pelvis</td>
</tr>
<tr>
<td>012</td>
<td>Shape-Elongated (L GE 1.5 X W)</td>
<td>020</td>
<td>Any Bone-Epiphysis</td>
</tr>
<tr>
<td>013</td>
<td>Location-Central</td>
<td>021</td>
<td>Any Bone-Growth Plate</td>
</tr>
<tr>
<td>014</td>
<td>Location-Cortical/Paraosteal</td>
<td>022</td>
<td>Tubular Bone-Articular Cortex</td>
</tr>
<tr>
<td>015</td>
<td>Location-Cortical</td>
<td>023</td>
<td>Tubular Bone-Metaphysis</td>
</tr>
<tr>
<td>016</td>
<td>Location-Cortical</td>
<td>024</td>
<td>Tubular Bone-Shaft</td>
</tr>
<tr>
<td>017</td>
<td>Location-Cortical</td>
<td>025</td>
<td>Matrix-Clouds</td>
</tr>
<tr>
<td>018</td>
<td>Fat Bone</td>
<td>026</td>
<td>Destruction-Genealogic</td>
</tr>
<tr>
<td>019</td>
<td>Small Bone</td>
<td>027</td>
<td>Destruction-Mothen</td>
</tr>
</tbody>
</table>

- **S02** Age 00-09 years
- **S03** Age 10-19 years
- **S04** Age 20-29 years
- **S05** Age 30-39 years
- **S06** Age 40 years and over
- **S07** Tumor Size 01-30 Millimeters
- **S08** Tumor Size 31-60 Millimeters
- **S09** Tumor Size 61-90 Millimeters
- **S10** Tumor Size 91 MM and over
- **S11** Shape-Round (L LT 1.5 X W)
- **S12** Shape-Elongated (L GE 1.5 X W)
- **S13** Location-Central
- **S14** Location-Eccentric
- **S15** Location-Cortical/Paraosteal
- **S16** Long Bone
- **S17** Flat Bone
- **S18** Small Bone
- **S19** Sacrum and Pelvis
- **S20** Any Bone-Epiphysis
- **S21** Any Bone-Growth Plate
- **S22** Tubular Bone-Articular Cortex
- **S23** Tubular Bone-Metaphysis
- **S24** Tubular Bone-Shaft
- **S25** Matrix-Clouds
- **S26** Destruction-Genealogic
- **S27** Destruction-Mothen
- **S28** Matrix-Clouds
- **S29** Matrix-Solid
- **S30** Matrix-Lump
- **S31** Matrix-Clouds
- **S32** Destruction-Genealogic
- **S33** Destruction-Mothen
- **S34** Destruction-Permeated
- **S35** Margin-Regular
- **S36** Margin-Lubulated
- **S37** Margin-Ragged
- **S38** Margin-Indistinct
- **S39** Transition Sharp or Smudged
- **S40** Invasive Zone
- **S41** Special Sign-Fracture
- **S42** Special Sign-Displacement
- **S43** Proliferation-Sclerotic Rim
- **S44** Proliferation-Multiple Small Foci
- **S45** Proliferation-Endostosis
- **S46** Periosteal-Hyperostosis
- **S47** Periosteal-Buttress
- **S48** Periosteal-Trabeculae (Septae)
- **S49** Cortex Expanded
- **S50** No Codman’s Triangle
- **S51** One Codman’s Triangle
- **S52** Two or More Codman’s Triangles
- **S53** No periostosis
- **S54** Laminated Periostosis
- **S55** Amorphous Periostosis
- **S56** No Spiculation
- **S57** Sunburst Spiculation
- **S58** Hair-on-end Spiculation
- **S59** Velvet Spiculation
- **S60** Periosteal Response-Delicate
- **S61** Periosteal Response-Coarse
TABLE 6

TESTS FOR BONE TUMOR DIAGNOSIS

<table>
<thead>
<tr>
<th>Test</th>
<th>Possible Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. TEST2</td>
<td>S02, S03, S04, S05, S06</td>
</tr>
<tr>
<td>2. TEST7</td>
<td>S07, S08, S09, S10</td>
</tr>
<tr>
<td>3. TEST11</td>
<td>S11, S12</td>
</tr>
<tr>
<td>4. TEST13</td>
<td>S13, S14, S15</td>
</tr>
<tr>
<td>5. TEST16</td>
<td>S16, S17, S18, S19</td>
</tr>
<tr>
<td>6. TEST20</td>
<td>S20, N</td>
</tr>
<tr>
<td>7. TEST21</td>
<td>S21, N</td>
</tr>
<tr>
<td>8. TEST22</td>
<td>S22, N</td>
</tr>
<tr>
<td>9. TEST23</td>
<td>S23, N</td>
</tr>
<tr>
<td>10. TEST24</td>
<td>S24, N</td>
</tr>
<tr>
<td>11. TEST27</td>
<td>S27, N</td>
</tr>
<tr>
<td>12. TEST28</td>
<td>S28, N</td>
</tr>
<tr>
<td>13. TEST29</td>
<td>S29, N</td>
</tr>
<tr>
<td>14. TEST30</td>
<td>S30, N</td>
</tr>
<tr>
<td>15. TEST31</td>
<td>S31, N</td>
</tr>
<tr>
<td>16. TEST32</td>
<td>S32, N</td>
</tr>
<tr>
<td>17. TEST33</td>
<td>S33, N</td>
</tr>
<tr>
<td>18. TEST34</td>
<td>S34, N</td>
</tr>
<tr>
<td>19. TEST35</td>
<td>S35, S36, S37, S38</td>
</tr>
<tr>
<td>20. TEST39</td>
<td>S39, S40, N</td>
</tr>
<tr>
<td>21. TEST41</td>
<td>S41, S42, N</td>
</tr>
<tr>
<td>22. TEST43</td>
<td>S43, N</td>
</tr>
<tr>
<td>23. TEST44</td>
<td>S44, N</td>
</tr>
<tr>
<td>24. TEST45</td>
<td>S45, N</td>
</tr>
<tr>
<td>25. TEST46</td>
<td>S46, N</td>
</tr>
<tr>
<td>26. TEST47</td>
<td>S47, N</td>
</tr>
<tr>
<td>27. TEST48</td>
<td>S48, N</td>
</tr>
<tr>
<td>28. TEST49</td>
<td>S49, N</td>
</tr>
<tr>
<td>29. TEST50</td>
<td>S50, S51, S52</td>
</tr>
<tr>
<td>30. TEST53</td>
<td>S53, S54, S55</td>
</tr>
<tr>
<td>31. TEST56</td>
<td>S56, S57, S58, S59</td>
</tr>
<tr>
<td>32. TEST56</td>
<td>S60, S61</td>
</tr>
</tbody>
</table>

Note: The symbol "N" denotes a "normal" attribute. It means that a test may fail to reveal any of the other attributes listed. Thus, for TEST41, the possible results are S41 or S42 or neither S41 nor S47 (N).
chapter, the abbreviations for diseases and attributes presented in Table 5 and Table 6 will be used. In the initial set of experiments, all tests were assigned unit cost and the cost of all misdiagnoses (e.g. deciding the tumor is CS when it is really GC) was assumed to be 100,000. This number is quite arbitrary, and is used simply to make the decision losses much greater than the testing losses.

Experiment 1. Diagnosis Based on All Attributes

Each of the twelve case histories was presented to the diagnostic program by inputting all the attributes for the case. The diagnostic program processed the attributes through the inference function and obtained a posterior distribution for the type of tumor. The results of this experiment are presented in Table 7 along with the diagnosis of a pathologist provided with each case history. The latter is traditionally accepted as the definitive diagnosis in cases of this type.

Experiment 2. Sequential Diagnoses--Actual Case Histories

The second experiment exercised the sequential capabilities of the diagnostic program. Again, all diseases were taken to be equally serious ($\text{cost}_{ij} = 100,000, i \neq j$) and all tests were assigned unit cost. The same twelve cases were analyzed by the program. For each case, the program was presented with a set of initial attributes. This set was obtained by collecting the results of the
TABLE 7

Diagnoses Based on all Available Attributes for Actual Bone Tumor Case Histories

<table>
<thead>
<tr>
<th>Case</th>
<th>Posterior Distribution*</th>
<th>Pathology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CB 0.12</td>
<td>GC</td>
</tr>
<tr>
<td></td>
<td>GC 0.87</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OS 0.65</td>
<td>OS</td>
</tr>
<tr>
<td></td>
<td>CS 0.35</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CB 1.00</td>
<td>CB</td>
</tr>
<tr>
<td>4</td>
<td>CS 0.99</td>
<td>CS</td>
</tr>
<tr>
<td>5</td>
<td>OS 1.00</td>
<td>OS</td>
</tr>
<tr>
<td>6</td>
<td>ES .33</td>
<td>RC</td>
</tr>
<tr>
<td></td>
<td>RC .67</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CS 0.78</td>
<td>CS</td>
</tr>
<tr>
<td></td>
<td>FS 0.22</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ES 0.04</td>
<td>ES</td>
</tr>
<tr>
<td></td>
<td>ES 0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RC 0.94</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ES 1.00</td>
<td>ES</td>
</tr>
<tr>
<td>10</td>
<td>CS 1.00</td>
<td>CB</td>
</tr>
<tr>
<td>11</td>
<td>GC 0.65</td>
<td>GC</td>
</tr>
<tr>
<td></td>
<td>CF 0.35</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PS 0.99</td>
<td>PS</td>
</tr>
</tbody>
</table>

* Only types with posterior probability greater than or equal to 0.01 are shown in the tables in this chapter.
first ten tests listed in Table 6 from the case histories. Thus each diagnostic problem was defined by approximately ten attributes. (In certain cases this number was smaller, because some tests are not relevant to specific bones.)

After processing the initial attributes, for the case, the program employed the test selection function to select a test to be run. The results of the test selected were determined by consulting the given case history. The attribute or attributes resulting from this test were given to the program and the inference-test selection cycle repeated. Throughout this experiment the test selection function searched the decision tree to a depth of one and limited the breadth of search to those tests relevant to the most likely disease type.

For each case, this sequential diagnosis was continued until the diagnostic program terminated the process. This termination occurred when the program determined the expected reduction in loss for the best test at the current decision node was less than the cost of the test.

An example of a sequential diagnosis is presented in Table 8 and the results of the experiment are summarized in Table 9.

The results of Experiment 2 underscore the potential advantage of sequential analysis of attributes in diagnosis. Since all diseases were taken to be equally serious for this experiment, the program found the best terminal decision to be the most probable disease. Since these same conditions held in Experiment 1, it is easy to make comparisons between the results of the two experiments.
<table>
<thead>
<tr>
<th>Test</th>
<th>Resulting Attributes</th>
<th>Posterior Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. --</td>
<td>S05, S10, S12, S15</td>
<td>CS 0.42</td>
</tr>
<tr>
<td></td>
<td>S16, NOT S20, NOT S21</td>
<td>ES 0.13</td>
</tr>
<tr>
<td></td>
<td>NOT S22, S23, S24</td>
<td>PS 0.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS 0.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RC 0.02</td>
</tr>
<tr>
<td>2. TEST29</td>
<td>S29</td>
<td>CS 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FS 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS 0.91</td>
</tr>
<tr>
<td>3. TEST50</td>
<td>S50</td>
<td>CS 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS 0.92</td>
</tr>
<tr>
<td>4. TEST56</td>
<td>S56</td>
<td>CS 0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FS 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS 0.93</td>
</tr>
</tbody>
</table>

Terminal decision -- PS
Pathology report -- PS
**TABLE 9**

Sequential Diagnosis of Bone Tumor Cases
Summary of Results for Actual Case Histories

<table>
<thead>
<tr>
<th>Case and Pathology</th>
<th>Number of Tests Selected by Program</th>
<th>Distribution at Point of Terminal Decision</th>
<th>Distribution When all Attributes Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (GC)</td>
<td>9</td>
<td>CB 0.21, GC 0.78</td>
<td>CB 0.12, GC 0.87</td>
</tr>
<tr>
<td>2. (OS)</td>
<td>12</td>
<td>CS 0.79, OS 0.21</td>
<td>CS 0.65, OS 0.35</td>
</tr>
<tr>
<td>3. (CB)</td>
<td>0</td>
<td>CB 1.00</td>
<td>CB 1.00</td>
</tr>
<tr>
<td>4. (CS)</td>
<td>4</td>
<td>CS 0.80, ES 0.08, FS 0.08, OS 0.04</td>
<td>CS 0.99</td>
</tr>
<tr>
<td>5. (OS)</td>
<td>4</td>
<td>CS 0.03, ES 0.02, OS 0.94, RC 0.03</td>
<td>OS 1.00</td>
</tr>
<tr>
<td>6. (RC)</td>
<td>13</td>
<td>ES 0.30, FS 0.01, RC 0.68</td>
<td>ES 0.33, RC 0.67</td>
</tr>
<tr>
<td>7. (CS)</td>
<td>4</td>
<td>CS 0.74, FS 0.26</td>
<td>CS 0.78, FS 0.22</td>
</tr>
<tr>
<td>8. (ES)</td>
<td>11</td>
<td>ES 0.05, FS 0.07, RC 0.87</td>
<td>ES 0.04, FS 0.02, RC 0.94</td>
</tr>
<tr>
<td>9. (ES)</td>
<td>5</td>
<td>CS 0.02, ES 0.88, OS 0.05, RC 0.05</td>
<td>ES 1.00</td>
</tr>
<tr>
<td>10. (CB)</td>
<td>3</td>
<td>CB 0.96, CF 0.04</td>
<td>CB 1.00</td>
</tr>
<tr>
<td>11. (GC)</td>
<td>5</td>
<td>CS 0.10, ES 0.01, GC 0.81, CF 0.08</td>
<td>GC 0.65, CF 0.33</td>
</tr>
</tbody>
</table>
12. (PS)  

3  

CS 0.05  
FS 0.02  
PS 0.93

Average number of initial attributes 9.4  
Average number of test by program 7.1
With regard to "accuracy," it can be seen that the lists of
terminal decisions from the two experiments are identical and these
decisions are the same as those of the pathologist in ten of the
twelve cases. The major difference between the two sets of results
is the average number of tests performed per diagnosis. In the first
case this average is 30. (The average is less than 32 because some
test results were not available or were not relevant for a given
case and the test was not counted.) Sequential analysis of the given
cases required an average of 16.7 tests per case. This average in-
cludes 9.4 tests on the average to obtain the initial attributes.
Thus, by employing sequential analysis, the program in each case
obtained the same diagnostic decision as it obtained using all attrib-
utes, but with only slightly more than half as many tests.

The nature of diagnosis of bone tumors makes this saving seem
immaterial. That is, almost all attributes are obtained from a
radiograph, and once the radiograph has been obtained, the marginal
cost of the tests considered here is essentially zero. One can
easily imagine a situation, however, in which tests are completely
independent of one another. In such a situation, the savings from
sequential diagnosis might be quite significant. The fact that the
performance of a diagnostician should be assessed in terms of both
accuracy and cost favors the sequential mode of operation for the
program. The question of how to assess the performance of a diag-
nostician will be considered at greater length later.
Another difference between the results of the two experiments is found in the posterior distributions at the points of a terminal decision. The average value of the maximum likelihood probability for the terminal decisions can be taken as an indication of the equivocation or uncertainty in the average decision. For Experiment 1 this value is 0.85 while for Experiment 2, it is 0.80. Therefore, the sequential diagnoses terminate on slightly less "certain" decisions.

Experiment 3. Sequential Analysis—Simulated Case Histories

Table 10 presents the results of the sequential diagnoses of ten simulated case histories. The generator function was used to develop the cases and the diagnostic program employed as usual. Again, all diseases were taken to be equally serious and all tests were assigned unit cost.

Again, the marked advantage of sequential diagnosis is evident. The average number of tests required for diagnosis was 17.0. Based on a maximum likelihood terminal decision, the diagnostic program's terminal decision was correct in nine of ten cases.

On the average, the diagnostic program was more certain of its terminal decisions than in the previous experiments (average probability of terminal decision = 90.5).
<table>
<thead>
<tr>
<th>Histological Type</th>
<th>Number of Initial Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Distribution at Point of Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FS</td>
<td>14</td>
<td>9</td>
<td>CS 0.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.73</td>
</tr>
<tr>
<td>2. ES</td>
<td>7</td>
<td>9</td>
<td>ES 0.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RC 0.11</td>
</tr>
<tr>
<td>3. OS</td>
<td>11</td>
<td>0</td>
<td>OS 1.00</td>
</tr>
<tr>
<td>4. GC</td>
<td>5</td>
<td>11</td>
<td>CS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CF 0.20</td>
</tr>
<tr>
<td>5. ES</td>
<td>12</td>
<td>6</td>
<td>CS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OS 0.04</td>
</tr>
<tr>
<td>6. RC</td>
<td>5</td>
<td>8</td>
<td>CS 0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RC 0.16</td>
</tr>
<tr>
<td>7. CB</td>
<td>11</td>
<td>8</td>
<td>CB 0.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CF 0.05</td>
</tr>
<tr>
<td>8. OS</td>
<td>11</td>
<td>8</td>
<td>OS 0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CS 0.02</td>
</tr>
<tr>
<td>9. FS</td>
<td>5</td>
<td>12</td>
<td>CS 0.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.88</td>
</tr>
<tr>
<td>10. GC</td>
<td>10</td>
<td>8</td>
<td>CB 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CF 0.01</td>
</tr>
</tbody>
</table>

Average number of initial attributes: 9.1
Average number of tests by program: 7.9
Chapter 6
DIAGNOSIS OF CONGENITAL HEART DISEASE

A. The Nature of the Diagnostic Problem

A prolonged study of a group of thirty-four types of congenital heart disease has been conducted by Warner and his associates (R12, R13, R14). As a result of this study, they developed a disease-attribute probability matrix for thirty-five types (including "normal") and fifty-seven attributes. The attributes can be grouped into four main categories: murmurs, electrocardiogram findings, X-ray findings, and other symptoms and physical signs. The problem of diagnosing heart disease cases based on this matrix is more difficult than the bone tumor problem discussed in Chapter 5. One reason for the increased difficulty is simply the increased number of diseases. Also certain groups of diseases have quite similar attribute probabilities in the matrix.

As noted in Chapter 2, Warner developed a computer program to perform diagnosis of congenital heart disease patients based on a Bayesian analysis of their signs and symptoms. His program employs the matrix mentioned above, but in addition it must account for certain dependencies (such as mutual exclusion of signs or symptoms). From the performance measures presented in Chapter 2, it can be seen that Warner's program performs at the level of an experienced physician.
The experiments discussed here involved the use of the disease-attribute probability matrix prepared by Warner in the diagnosis of congenital heart disease. As before, the matrix was the basis for each of the disease types and the appropriate attribute lists created. Twenty-eight tests were also defined for the problem. Dr. Warner provided nine case histories, each with the correct diagnosis and the diagnosis obtained by his program. In this instance, the correct diagnoses were determined by follow-up studies such as heart catheterization or autopsy.

Table 11 presents the names of the thirty-five states of the information structure used in these experiments and the names of the corresponding diseases. Table 12 lists the attributes of the problem; and Table 13 the tests.

B. Experiments in Congenital Heart Disease Diagnosis

Experiment 4. Diagnosis Based on All Attributes

The first experiment tested the diagnostic capability of the program given all the known attributes for each of the actual case histories provided by Dr. Warner. The results of this experiment are summarized in Table 14. In each instance, the diagnostic program duplicated the results obtained by Warner's program for the given case history. (That is, both programs arrived at the same posterior probability distribution given all attributes.)
# TABLE 11

**Heart Disease Types**

<table>
<thead>
<tr>
<th>States</th>
<th>Diseases</th>
<th>States</th>
<th>Diseases</th>
</tr>
</thead>
<tbody>
<tr>
<td>D01</td>
<td>Normal</td>
<td>D18</td>
<td>Patent ductus arteriosus</td>
</tr>
<tr>
<td>D02</td>
<td>Atrial septal defect</td>
<td>D19</td>
<td>Pulmonary arterio-venous Pustula</td>
</tr>
<tr>
<td>D03</td>
<td>Atrial septal defect with pulmonary stenosis</td>
<td>D20</td>
<td>Congenital mitral disease</td>
</tr>
<tr>
<td>D04</td>
<td>Atrial septal defect with pulmonary hypertension</td>
<td>D21</td>
<td>Primary myocardial disease</td>
</tr>
<tr>
<td>D05</td>
<td>Atrio-ventricular communis</td>
<td>D22</td>
<td>Anomalous origin or coronary artery</td>
</tr>
<tr>
<td>D06</td>
<td>Partial anomalous pulmonary venous connection</td>
<td>D23</td>
<td>Congenital aortic disease</td>
</tr>
<tr>
<td>D07</td>
<td>Total anomalous pulmonary venous connection</td>
<td>D24</td>
<td>Ventricular septal defect with pulmonary flow (\geq 1.4) systemic flow</td>
</tr>
<tr>
<td>D08</td>
<td>Tricuspid atresia (without transposition)</td>
<td>D25</td>
<td>Coarctation of aorta</td>
</tr>
<tr>
<td>D09</td>
<td>Ebstein's anomaly</td>
<td>D26</td>
<td>Truncus arteriosus</td>
</tr>
<tr>
<td>D10</td>
<td>Ventricular septal defect with valvular pulmonary stenosis</td>
<td>D27</td>
<td>Transposition</td>
</tr>
<tr>
<td>D11</td>
<td>Ventricular septal defect with infundibular pulmonary stenosis</td>
<td>D28</td>
<td>Hypertrophic subaortic stenosis</td>
</tr>
<tr>
<td>D12</td>
<td>Pulmonary stenosis, valvular, gradient (\geq 40) mm Hg.</td>
<td>D29</td>
<td>Absent aortic arch</td>
</tr>
<tr>
<td>D13</td>
<td>Pulmonary stenosis, infundibular, gradient (\geq 40) mm Hg.</td>
<td>D30</td>
<td>Ventricular septal defect with pulmonary flow (\geq 1.4) systemic flow</td>
</tr>
<tr>
<td>D14</td>
<td>Pulmonary atresia</td>
<td>D31</td>
<td>Ventricular septal defect with pulmonary hypertension</td>
</tr>
<tr>
<td>D15</td>
<td>Peripheral pulmonary stenosis</td>
<td>D32</td>
<td>Patent ductus arteriosus with pulmonary hypertension</td>
</tr>
<tr>
<td>D16</td>
<td>Pulmonary hypertension</td>
<td>D33</td>
<td>Tricuspid atresia with transplantation</td>
</tr>
<tr>
<td>D17</td>
<td>Aortic pulmonary window</td>
<td>D34</td>
<td>Pulmonary stenosis gradient (\leq 40) mm Hg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D35</td>
<td>Ruptured sinus Valsalva</td>
</tr>
</tbody>
</table>
TABLE 12
Attributes for Congenital Heart Disease

<table>
<thead>
<tr>
<th>Sign</th>
<th>Meaning</th>
<th>Sign</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>Age, less than 1 year</td>
<td>S29</td>
<td>Post systolic</td>
</tr>
<tr>
<td>S02</td>
<td>Age, 1 year to 20 years</td>
<td>S30</td>
<td>Post continuous</td>
</tr>
<tr>
<td>S03</td>
<td>Age, 20 or more years</td>
<td>S31</td>
<td>Murmur louder than gr 3/6 (10 mm)</td>
</tr>
<tr>
<td>S04</td>
<td>Cyanosis, mild</td>
<td>S32</td>
<td>Accented F2</td>
</tr>
<tr>
<td>S05</td>
<td>Cyanosis, severe (with clubbing)</td>
<td>S33</td>
<td>Diminished P2</td>
</tr>
<tr>
<td>S06</td>
<td>Cyanosis intermittent</td>
<td>S34</td>
<td>Fixed split P2</td>
</tr>
<tr>
<td>S07</td>
<td>Cyanosis differential</td>
<td>S35</td>
<td>Femoral pulse less than brachial</td>
</tr>
<tr>
<td>S08</td>
<td>Squatting</td>
<td>S36</td>
<td>Atrial fibrillation or broad</td>
</tr>
<tr>
<td>S09</td>
<td>Apex systolic</td>
<td>S37</td>
<td>notched P wave</td>
</tr>
<tr>
<td>S10</td>
<td>Apex systolic, holo</td>
<td>S38</td>
<td>Axis, right (more than 110°)</td>
</tr>
<tr>
<td>S11</td>
<td>Apex systolic, mid</td>
<td>S39</td>
<td>Axis, left (less than 0°)</td>
</tr>
<tr>
<td>S12</td>
<td>Apex diastolic</td>
<td>S40</td>
<td>R wave greater than 1.2 mv in lead V1</td>
</tr>
<tr>
<td>S13</td>
<td>Apex diastolic, early</td>
<td>S41</td>
<td>rR' or qR in lead V1</td>
</tr>
<tr>
<td>S14</td>
<td>Apex diastolic, late</td>
<td>S42</td>
<td>R wave greater than 2.5 mv in lead V6</td>
</tr>
<tr>
<td>S15</td>
<td>L 4th systolic</td>
<td>S43</td>
<td>T wave inversion in lead V6</td>
</tr>
<tr>
<td>S16</td>
<td>L 4th systolic, holo</td>
<td>S44</td>
<td>Rib notching</td>
</tr>
<tr>
<td>S17</td>
<td>L 4th systolic, mid</td>
<td>S45</td>
<td>Peripheral vessels increased</td>
</tr>
<tr>
<td>S18</td>
<td>L 4th continuous</td>
<td>S46</td>
<td>Peripheral vessels decreased</td>
</tr>
<tr>
<td>S19</td>
<td>L 4th diastolic</td>
<td>S47</td>
<td>Hilar vessels increased</td>
</tr>
<tr>
<td>S20</td>
<td>L 4th diastolic, holo</td>
<td>S48</td>
<td>Hilar vessels decreased</td>
</tr>
<tr>
<td>S21</td>
<td>L 4th diastolic, early</td>
<td>S49</td>
<td>Main pulmonary artery large</td>
</tr>
<tr>
<td>S22</td>
<td>L 2nd systolic</td>
<td>S50</td>
<td>Main pulmonary artery not seen</td>
</tr>
<tr>
<td>S23</td>
<td>L 2nd systolic, holo</td>
<td>S51</td>
<td>Aorta large</td>
</tr>
<tr>
<td>S24</td>
<td>L 2nd systolic, mid</td>
<td>S52</td>
<td>Aorta small</td>
</tr>
<tr>
<td>S25</td>
<td>L 2nd continuous</td>
<td>S53</td>
<td>Cardiomegaly</td>
</tr>
<tr>
<td>S27</td>
<td>R 2nd systolic</td>
<td>S54</td>
<td>Snowman</td>
</tr>
<tr>
<td>S28</td>
<td>R 2nd diastolic</td>
<td>S55</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Tests</th>
<th>Possible Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST1</td>
<td>S01, S02, S03</td>
</tr>
<tr>
<td>TEST4</td>
<td>S04, S05, S06, S07, N</td>
</tr>
<tr>
<td>TEST8</td>
<td>S08, N</td>
</tr>
<tr>
<td>TEST9</td>
<td>S09, N</td>
</tr>
<tr>
<td>TEST10</td>
<td>S10, S11, N</td>
</tr>
<tr>
<td>TEST12</td>
<td>S12, N</td>
</tr>
<tr>
<td>TEST13</td>
<td>S13, S14, N</td>
</tr>
<tr>
<td>TEST15</td>
<td>S15, N</td>
</tr>
<tr>
<td>TEST16</td>
<td>S16, S17, S18, N</td>
</tr>
<tr>
<td>TEST19</td>
<td>S19, N</td>
</tr>
<tr>
<td>TEST20</td>
<td>S20, S21, N</td>
</tr>
<tr>
<td>TEST22</td>
<td>S22, N</td>
</tr>
<tr>
<td>TEST23</td>
<td>S23, S24, S25, N</td>
</tr>
<tr>
<td>TEST27</td>
<td>S27, N</td>
</tr>
<tr>
<td>TEST28</td>
<td>S28, N</td>
</tr>
<tr>
<td>TEST29</td>
<td>S29, S30, N</td>
</tr>
<tr>
<td>TEST31</td>
<td>S31, N</td>
</tr>
<tr>
<td>TEST35</td>
<td>S35, S36, N</td>
</tr>
<tr>
<td>TEST37</td>
<td>S36, S37, N</td>
</tr>
<tr>
<td>TEST38</td>
<td>S38, N</td>
</tr>
<tr>
<td>TEST40</td>
<td>S40, N</td>
</tr>
<tr>
<td>TEST41</td>
<td>S41, S42, N</td>
</tr>
<tr>
<td>TEST43</td>
<td>S43, N</td>
</tr>
<tr>
<td>TEST44</td>
<td>S44, N</td>
</tr>
<tr>
<td>TEST45</td>
<td>S45, N</td>
</tr>
<tr>
<td>TEST46</td>
<td>S46, N</td>
</tr>
<tr>
<td>TEST47</td>
<td>S47, N</td>
</tr>
<tr>
<td>TEST48</td>
<td>S48, S49, N</td>
</tr>
<tr>
<td>TEST50</td>
<td>S50, S51, N</td>
</tr>
<tr>
<td>TEST52</td>
<td>S52, S54, N</td>
</tr>
<tr>
<td>TEST54</td>
<td>S54, S55, N</td>
</tr>
<tr>
<td>TEST56</td>
<td>S56, N</td>
</tr>
<tr>
<td>TEST57</td>
<td>S57, N</td>
</tr>
</tbody>
</table>
### TABLE 14

Diagnoses Based on All Available Attributes for Actual Heart Disease Case Histories

<table>
<thead>
<tr>
<th>Case</th>
<th>Posterior Distribution*</th>
<th>Definitive Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DO3 0.91</td>
<td>DO9</td>
</tr>
<tr>
<td></td>
<td>NORMAL 0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D34 0.03</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DO5 0.84</td>
<td>DO4</td>
</tr>
<tr>
<td></td>
<td>DO2 0.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D31 0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D04 0.03</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>D32 1.00</td>
<td>D02</td>
</tr>
<tr>
<td>4</td>
<td>D20 0.41</td>
<td>NORMAL</td>
</tr>
<tr>
<td></td>
<td>D28 0.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NORMAL 0.22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D24 0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D34 0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D11 0.01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DO8 0.94</td>
<td>D33</td>
</tr>
<tr>
<td></td>
<td>D33 0.05</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>D32 0.98</td>
<td>D32</td>
</tr>
<tr>
<td></td>
<td>D29 0.02</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>D31 0.47</td>
<td>D31</td>
</tr>
<tr>
<td></td>
<td>D30 0.37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO5 0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO2 0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D32 0.02</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>D30 0.87</td>
<td>D30</td>
</tr>
<tr>
<td></td>
<td>DO2 0.12</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>D31 0.70</td>
<td>D27</td>
</tr>
<tr>
<td></td>
<td>D27 0.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D26 0.10</td>
<td></td>
</tr>
</tbody>
</table>

* Only diseases with probability greater than or equal to 0.01 are shown.
**Experiment 5. Sequential Diagnosis of Heart Disease Cases**

The actual heart disease cases were also diagnosed by the program using the sequential mode of operation. In each case, the initial attributes presented to the program were the results from a set of seven tests relating to physical signs. The diseases were assumed to be equally serious ($l_{ij} = 100,000$, $i \neq j$) and all tests were assigned unit cost. The search depth in the test selection function was one in each case.

A summary of the results of this experiment is presented in Table 15. Again, the advantage of sequential diagnosis is apparent. The program required an average of 5.8 tests to obtain a diagnosis compared to the thirty-three tests required to determine all attributes. This small number of tests is interesting. Recall the sequential diagnosis of the bone tumor cases required an average of 6.7 tests per case, although the problem involves only one quarter as many states as the heart disease problem. Several reasons might be advanced to account for this. First, the tests associated with heart disease may include a number which have little value in differentiating groups of diseases. Thus, in a given problem, the test selection function may choose a terminal decision after relatively few tests have been run. A second reason may be the relevance of more inter-attribute relationships in the heart disease problem. Such relationships may be quite useful in diagnosis, but the testing sequences for them are not examined since the
TABLE 15

Sequential Diagnosis of Actual Heart Disease Cases

<table>
<thead>
<tr>
<th>Case and Definitive Diagnosis</th>
<th>Number of Tests Selected by Program</th>
<th>Distribution at Terminal Decision</th>
<th>Distribution Based on all Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DO9</td>
<td>10</td>
<td>NORMAL 0.04</td>
<td>NORMAL 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D02 0.06</td>
<td>D03 0.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D03 0.69</td>
<td>D34 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D11 0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D18 0.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D26 0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D34 0.03</td>
<td></td>
</tr>
<tr>
<td>2. DO4</td>
<td>4</td>
<td>D02 0.08</td>
<td>D02 0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D04 0.17</td>
<td>D04 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D05 0.62</td>
<td>D05 0.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D31 0.10</td>
<td>D31 0.03</td>
</tr>
<tr>
<td>3. DO2</td>
<td>1</td>
<td>D27 0.03</td>
<td>D32 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D32 0.96</td>
<td></td>
</tr>
<tr>
<td>4. NORMAL</td>
<td>10</td>
<td>NORMAL 0.07</td>
<td>NORMAL 0.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D10 0.03</td>
<td>D28 0.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D11 0.07</td>
<td>D24 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D12 0.02</td>
<td>D20 0.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D20 0.67</td>
<td>D34 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D24 0.01</td>
<td>D11 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D28 0.10</td>
<td></td>
</tr>
<tr>
<td>5. D33</td>
<td>3</td>
<td>D08 0.92</td>
<td>D08 0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D33 0.01</td>
<td>D33 0.05</td>
</tr>
<tr>
<td>6. D32</td>
<td>0</td>
<td>D32 0.98</td>
<td>D32 0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D29 0.01</td>
<td>D29 0.02</td>
</tr>
<tr>
<td>7. D31</td>
<td>10</td>
<td>D04 0.01</td>
<td>D31 0.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D05 0.09</td>
<td>D30 0.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D31 0.86</td>
<td>D05 0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D32 0.02</td>
<td>D32 0.02</td>
</tr>
<tr>
<td>8. D30</td>
<td>8</td>
<td>D02 0.03</td>
<td>D30 0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D05 0.02</td>
<td>D02 0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D20 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D30 0.89</td>
<td></td>
</tr>
</tbody>
</table>
9, D27

6

D11 0.02  D31 0.70
D19 0.01  D27 0.20
D24 0.06  D26 0.10
D26 0.06
D31 0.77
D33 0.03

Average number of initial attributes = 7
Average number of tests by program = 5.8
depth of the tree search is limited to one level. Unfortunately, an increase in the depth of search leads to prohibitive amounts of computation in the heart disease problem. A deeper search may be possible if more powerful breadth-limiting heuristics are developed.

On the whole, the performance of the program with sequential diagnosis is comparable to that when all attributes are available. The one apparent exception to this involves case 9. Here the sequential diagnosis failed to assign a probability of greater than 0.01 to disease D27. The seriousness of this failure depends on medical considerations which are not discussed here. The general problem of measuring diagnostic performance, however, will be discussed in Chapter 8.
Chapter 7

FURTHER EXPERIMENTS WITH THE DIAGNOSTIC SYSTEM

In order to explore the potential value of the diagnostic system as a tool for the study of a variety of diagnostic problems and strategies, some further experiments were performed. The results of these experiments are reported in this chapter.

Experiment 6. The Effect of a Very Serious State

In the experiments discussed in Chapters 5 and 6, it was assumed that the loss for misdiagnosis was the same for all pairs of diseases. For each experiment, the elements of the loss function matrix were taken to be 0 for $l_{ij}$ and 100,000 for $l_{ij}$, $i 
eq j$. For this reason, the diagnostic program always selected the most likely disease as its terminal decision. One can easily imagine situations, however, in which the assumption of a constant loss for misdiagnosis independent of the actual disease is unrealistic. For example, it may be far more serious to diagnose pneumonia as a common cold than vice versa. Since the diagnostic program incorporates such considerations in its rules for selecting a terminal decision, changes in the loss function matrix can result in pronounced changes in its decisions.
This effect was observed in two different situations. In the first, the loss function matrix is presented in Table 16. Note that it is very costly to miss the diagnosis of CB. The misdiagnosis of either CS or ES as a disease other than one of these two or CB is quite serious, but it is not particularly serious to diagnose CS as ES or CB or ES as CS or CB. Failure to diagnose one of the remaining diseases results in a loss which is independent of the diagnosis made.

The generator was used to generate seven case histories of bone tumor cases. Each case was diagnosed by the diagnostic program in the light of the new loss function. The results of this experiment are summarized in Table 17. From this table, it can be seen that the new loss function affects only one decision, that of case 3. In this case, the diagnostic program selected CB as the terminal decision in spite of the fact that GC (the actual disease) was more than three times as probable. The loss for diagnosing CB as GC is 1,000 times that of diagnosing GC as CB, however, and this fact dominates the decision of the program. The relative seriousness of CB does not affect the diagnoses of the remaining cases because the observed attributes excluded CB as a possibility in each case.

The effect of a serious disease on diagnosis can be made even more pronounced if the serious disease is not easily distinguished from other less serious ones. For example, the disease CS often
TABLE 16
Loss Function Matrix for Bone Tumor Diagnosis
(in thousands)

<table>
<thead>
<tr>
<th>Diagnosis</th>
<th>CB</th>
<th>CS</th>
<th>ES</th>
<th>FS</th>
<th>GC</th>
<th>OS</th>
<th>PS</th>
<th>RC</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CS</td>
<td>100</td>
<td>0</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ES</td>
<td>100</td>
<td>0.1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FS</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GC</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OS</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PS</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RC</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>CF</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
### TABLE 17
Sequential Diagnosis of Cases for Loss  
Function of Table 16

<table>
<thead>
<tr>
<th>Case and Disease</th>
<th>Number of Initial Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (PS)</td>
<td>15</td>
<td>1</td>
<td>PS* 1.00</td>
</tr>
<tr>
<td>2. (GC)</td>
<td>8</td>
<td>7</td>
<td>GC* 0.90, FS 0.09, CS 0.01</td>
</tr>
<tr>
<td>3. (GC)</td>
<td>9</td>
<td>3</td>
<td>CB* 0.24, GC 0.76</td>
</tr>
<tr>
<td>4. (ES)</td>
<td>10</td>
<td>0</td>
<td>ES* 0.99, RC 0.01</td>
</tr>
<tr>
<td>5. (ES)</td>
<td>8</td>
<td>2</td>
<td>ES* 0.96, CS 0.02</td>
</tr>
<tr>
<td>6. (OS)</td>
<td>13</td>
<td>0</td>
<td>OS* 1.00</td>
</tr>
<tr>
<td>7. (GC)</td>
<td>8</td>
<td>12</td>
<td>GC* 0.89, FS 0.09, CS 0.02</td>
</tr>
</tbody>
</table>

* Terminal decision by program.
appears in a terminal distribution when the actual disease is another. This means that CS has not been excluded as a possible diagnosis when a terminal decision is made. By making CS very serious relative to the other diseases, the decisions of the program can be strongly influenced.

The loss function matrix presented in Table 18 represents just this situation. A series of simulated cases was diagnosed by the program using this loss function. The results of this experiment are summarized in Table 19. Here the seriousness of CS dominates all decisions, and the terminal decision is CS in all cases. Note also that the terminal decision is made after relatively few tests have been run and while the posterior distribution is relatively diffuse. The predominance of terminal decisions for disease CS is a result of the seriousness of that disease. The decrease in the number of tests per case and the diffuse terminal distributions reflect the difficulty finding a single test which promises to significantly alter the expected loss. Since the diagnostic program employed a one level look ahead in searching the decision tree for these cases, it did not consider possible sequences of several tests to resolve this problem. This point will be discussed in more detail later in the thesis.

The above example is but one in which the loss function has a significant effect on the terminal decisions made by the diagnostic program. Because the test selection strategy also accounts for the loss function, it, too, is affected by changes in the matrix. There-
# TABLE 18

Loss Function Matrix for Bone Tumor Diagnosis  
(in thousands)

<table>
<thead>
<tr>
<th>Actual Disease</th>
<th>CB</th>
<th>CS</th>
<th>ES</th>
<th>FS</th>
<th>GC</th>
<th>OS</th>
<th>PS</th>
<th>RC</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CS</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ES</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FS</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GC</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OS</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PS</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RC</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>CF</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
### TABLE 19

Sequential Diagnoses of Cases for Loss Function of Table 18

<table>
<thead>
<tr>
<th>Case and Disease</th>
<th>Number of Initial Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (FS)</td>
<td>14</td>
<td>1</td>
<td>CS* 0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GF 0.07</td>
</tr>
<tr>
<td>2. (OS)</td>
<td>8</td>
<td>2</td>
<td>CS* 0.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.02</td>
</tr>
<tr>
<td>3. (OS)</td>
<td>8</td>
<td>4</td>
<td>CS* 0.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GS 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RC 0.02</td>
</tr>
<tr>
<td>4. (OS)</td>
<td>7</td>
<td>3</td>
<td>CS* 0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GS 0.91</td>
</tr>
<tr>
<td>5. (CB)</td>
<td>6</td>
<td>2</td>
<td>CS* 0.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CB 0.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.48</td>
</tr>
<tr>
<td>6. (GC)</td>
<td>7</td>
<td>2</td>
<td>CS* 0.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CB 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ES 0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC 0.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GS 0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.06</td>
</tr>
<tr>
<td>7. (OS)</td>
<td>15</td>
<td>5</td>
<td>CS* 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GS 0.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FS 0.06</td>
</tr>
</tbody>
</table>

* Terminal decision by program.
fore, an important facility in the study of diagnostic strategies for a particular application is the ability to assess the sensitivity of these strategies to the loss function. Although the current version of the diagnostic system restricts the loss function to a matrix form, it is still possible to employ wide ranges of the values of the matrix elements in a given application study. This facility coupled with the capabilities of the generator makes it possible to study the performance of different versions of the diagnostic program with a variety of matrix loss function.

Experiment 8. Studies of a Test-Selection Heuristic

The experiments discussed in Chapters 5 and 6 indicate the value of sequential diagnosis in reducing the number of tests required for a diagnosis. Therefore, it is worth some effort to improve the operation of the test-selection function.

One problem which can arise in the use of the test-selection function of the current system is the appreciable amounts of computation required to evaluate all the relevant tests at a given decision node. It would be quite desirable to reduce the amount of computation devoted to test selection provided that the diagnostic capability of the program were not impaired. As an example of the amount of computation involved in test selection, consider the following. In the diagnosis of congenital heart disease, there can be as many as thirty-five states with non-zero probabilities in the current distribution. If there are twenty relevant tests at a given
decision node, each with two possible results, a one-level evaluation of these tests could require the creation of forty distributions, each requiring the computation of thirty-five updated probabilities. This is a significant amount of processing for a highly interactive program, and the example cited does not represent a particularly large set of alternatives. Since the test-selection function may be performed many times during a diagnosis, there is a good reason to reduce the time required to perform it. An obvious approach is to improve the efficiency of the code for the function. While this would no doubt lead to improvements, it was not attempted. Attention was focused on attempting to reduce the number of tests considered, rather than reducing the time devoted to the evaluation of an individual test.

This approach was motivated by the results of the experiments with sequential diagnosis. There it was observed that relatively few tests were required for diagnosis by the program. The particular set of tests employed for a given diagnosis is determined dynamically by the program, and varies from one diagnosis to another. If one could guess which tests would be relevant to a particular diagnosis, the total number of tests considered could be reduced significantly. A guess about the relevance of certain tests must not be irreversible, however, because the value of some tests will become apparent only after other tests have been run.

At any stage in a diagnosis, the current distribution provides
the most logical basis for a hypothesis about the future relevance of particular tests. One heuristic which incorporates this view is the one which restricts the set of tests considered to those which are relevant to the state which is the best terminal decision at the current node. This heuristic favors those tests which tend to confirm or disprove the current "best guess" about the problem. It also had the property of reversibility mentioned above. When the terminal decision changes, the set of relevant tests changes correspondingly.

This heuristic was employed in a number of experiments with both congenital heart disease problems and bone tumor problems. In the cases studied it resulted in the same number of tests selected as the standard function which employs no such heuristic. This heuristic does reduce the average number of decision nodes considered per diagnosis. This reduction is not great, however, because in both problem areas the diseases share many attributes in common, and hence many relevant tests. Thus, at any decision node, almost all the tests are relevant to the state determined to be the best terminal decision.

A second heuristic which offered a potentially greater reduction in the number of decision nodes considered per diagnosis was also considered. This heuristic employs the current distribution to "guess" which tests will not be useful in the remainder of the diagnosis. Tests which are thought to have little value are temporarily removed
from consideration. At a later point in the diagnosis these tests may be released for further consideration.

The actual operation of this heuristic is as follows. At a given decision node, the set of relevant tests is evaluated by the test selection function. Then the set of tests is partitioned into two disjoint subsets. In the first are all those tests with the property that the sum of the cost of the test plus the expected loss of a terminal decision after the test has been run exceeds the expected loss of the current terminal decision. These tests are said to be dominated. The second set consists of all the remaining undominated tests. The heuristic hypothesizes that the tests in the dominated set will remain dominated for the remainder of the diagnosis. This set of tests is placed on the top of a push-down stack. At each decision node the push-down stack is examined prior to evaluating each test. If the test is found in the stack it is not considered at the decision node.

In general, then, each iteration of the test selection function produces a new set of dominated tests which are pushed onto the stack. This means the set of relevant tests is generally decreased at each stage of the diagnosis. Whenever there are no undominated tests at a given decision node (i.e. whenever the terminal decision is selected), the program releases the set of dominated tests (if one exists) on the bottom of the stack. This corresponds to re-evaluating those tests which were tentatively discarded earliest in the diagnosis.
The reason for this choice is that it is desirable to reconsider tests which were dominated when the distribution was quite different from the present one. If the distribution has changed little, tests which were formerly dominated are apt to be currently dominated. Actually, there is no guarantee that this method will produce the desired effect. It is used primarily as an example of a possible approach, and additional discussion will be devoted to the subject below.

The "dominated-test" heuristic was tested in the sequential diagnosis of both the heart cases and bone tumor cases. The nine heart disease cases and the twelve bone tumor cases were used as the testing sample. The same initial attributes for a given case were given to both the "dominated-test" heuristic and the standard version of the diagnostic program. The number of tests by the program, the number of decision nodes considered during diagnosis, and the distribution at the terminal decision were all recorded. These results are summarized in Tables 20 through 23. A number of these results have an interesting interpretation.

In both the heart disease cases and the bone tumor cases, the dominated-test heuristic results in a substantial reduction in the average number of decision nodes considered per diagnosis. In the heart disease problem, this heuristic results in a larger average number of tests performed per diagnosis. In situations when the cost of an average test exceeds the value of the computation saved, this is an undesirable effect. The reason for this reduction in diagnostic efficiency can be seen from the following interpretation of
### TABLE 20
Sequential Diagnosis of Heart Disease Cases—Standard Test Selection Function

<table>
<thead>
<tr>
<th>Case and Diagnosis Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Number of Decision Nodes Considered</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DO9</td>
<td>10</td>
<td>541</td>
<td>NORMAL 0.04&lt;br&gt;DO3 0.69&lt;br&gt;DO4 0.03&lt;br&gt;DO2 0.06&lt;br&gt;D18 0.05&lt;br&gt;D26 0.03</td>
</tr>
<tr>
<td>2. DO4</td>
<td>4</td>
<td>287</td>
<td>DO2 0.08&lt;br&gt;DO4 0.17&lt;br&gt;DO5 0.62&lt;br&gt;D31 0.10&lt;br&gt;D27 0.03&lt;br&gt;D32 0.96</td>
</tr>
<tr>
<td>3. DO2</td>
<td>1</td>
<td>133</td>
<td>D27 0.03&lt;br&gt;D32 0.96</td>
</tr>
<tr>
<td>4. NORMAL</td>
<td>10</td>
<td>523</td>
<td>NORMAL 0.07&lt;br&gt;P10 0.03&lt;br&gt;D11 0.07&lt;br&gt;D12 0.02&lt;br&gt;D20 0.67&lt;br&gt;D24 0.01&lt;br&gt;D28 0.10</td>
</tr>
<tr>
<td>5. D33</td>
<td>3</td>
<td>248</td>
<td>DO8 0.92&lt;br&gt;D33 0.01</td>
</tr>
<tr>
<td>6. D32</td>
<td>0</td>
<td>66</td>
<td>D32 0.98&lt;br&gt;D29 0.01</td>
</tr>
<tr>
<td>7. D31</td>
<td>10</td>
<td>513</td>
<td>DO4 0.01&lt;br&gt;D05 0.09&lt;br&gt;D31 0.86&lt;br&gt;D32 0.02</td>
</tr>
<tr>
<td>Program</td>
<td>Test Cases</td>
<td>Decision Nodes</td>
<td>Average Tests</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>D30</td>
<td>7</td>
<td>8</td>
<td>457</td>
</tr>
<tr>
<td>D02</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D05</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D20</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D30</td>
<td>0.89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D27</td>
<td>7</td>
<td>6</td>
<td>379</td>
</tr>
<tr>
<td>D11</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D19</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D24</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D26</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D31</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D33</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average number of tests by program = 5.8
Average number of decision nodes considered = 350
<table>
<thead>
<tr>
<th>Case and Diagnosis</th>
<th>Initial Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Number of Decision Nodes Considered</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DO9</td>
<td>7</td>
<td>11</td>
<td>283</td>
<td>NORMAL 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D03 0.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D05 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D02 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D11 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D18 0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D26 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D34 0.03</td>
</tr>
<tr>
<td>2. DO4</td>
<td>7</td>
<td>5</td>
<td>163</td>
<td>D02 0.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DO4 0.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D05 0.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D31 0.03</td>
</tr>
<tr>
<td>3. DO2</td>
<td>7</td>
<td>1</td>
<td>66</td>
<td>D27 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D32 0.96</td>
</tr>
<tr>
<td>4. NORMAL</td>
<td>7</td>
<td>16</td>
<td>345</td>
<td>NORMAL 0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D11 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D15 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D20 0.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D28 0.02</td>
</tr>
<tr>
<td>5. D33</td>
<td>7</td>
<td>3</td>
<td>176</td>
<td>D08 0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D33 0.01</td>
</tr>
<tr>
<td>6. D32</td>
<td>7</td>
<td>0</td>
<td>66</td>
<td>D32 0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D29 0.01</td>
</tr>
<tr>
<td>7. D31</td>
<td>7</td>
<td>11</td>
<td>269</td>
<td>D04 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D05 0.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D30 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D31 0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D32 0.07</td>
</tr>
</tbody>
</table>
8. D30  7  10  301
    D02  0.04
          D04  0.02
          D05  0.02
          D18  0.03
          D20  0.04
          D30  0.70
          D31  0.08
          D32  0.02

9. D27  7  6  216
    D11  0.02
          D31  0.77
          D26  0.06
          D26  0.06
          D33  0.03

Average number of tests by program = 7
Average number of decision nodes considered = 208
TABLE 22

Sequential Diagnosis of Bone Tumor Cases
Standard Test Selection Function

<table>
<thead>
<tr>
<th>Case and Pathology</th>
<th>Initial Attributes</th>
<th>Number of Tests Selected by Program</th>
<th>Number of Decision Nodes Considered</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (CC)</td>
<td>7</td>
<td>9</td>
<td>269</td>
<td>GC 0.78 CB 0.21</td>
</tr>
<tr>
<td>2. (OS)</td>
<td>10</td>
<td>12</td>
<td>425</td>
<td>OS 0.35 CS 0.65</td>
</tr>
<tr>
<td>3. (CB)</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>CB 1.00</td>
</tr>
<tr>
<td>4. (CS)</td>
<td>70</td>
<td>4</td>
<td>223</td>
<td>CS 0.99</td>
</tr>
<tr>
<td>5. (OS)</td>
<td>10</td>
<td>4</td>
<td>194</td>
<td>OS 1.00</td>
</tr>
<tr>
<td>6. (RC)</td>
<td>10</td>
<td>13</td>
<td>406</td>
<td>RC 0.68 ES 0.30 FS 0.01</td>
</tr>
<tr>
<td>7. (CS)</td>
<td>8</td>
<td>4</td>
<td>228</td>
<td>CS 0.78 FS 0.22</td>
</tr>
<tr>
<td>8. (ES)</td>
<td>8</td>
<td>11</td>
<td>475</td>
<td>ES 0.05 FS 0.07 RC 0.87</td>
</tr>
<tr>
<td>9. (ES)</td>
<td>6</td>
<td>5</td>
<td>278</td>
<td>ES 0.88 RC 0.05 OS 0.05 CS 0.02</td>
</tr>
<tr>
<td>10. (CB)</td>
<td>10</td>
<td>3</td>
<td>109</td>
<td>CB 0.96 CF 0.04</td>
</tr>
<tr>
<td>11. (GC)</td>
<td>10</td>
<td>5</td>
<td>169</td>
<td>GC 0.81 CS 0.10 CF 0.08 ES 0.01</td>
</tr>
<tr>
<td>12. (PS)</td>
<td>10</td>
<td>3</td>
<td>142</td>
<td>PS 0.93 FS 0.02 CS 0.05</td>
</tr>
</tbody>
</table>

Average number of tests by program = 7.1
Average number of decision nodes considered = 243
### TABLE 23
Sequential Diagnosis of Bone Tumor Cases
Dominated Test Heuristic

<table>
<thead>
<tr>
<th>Case and Diagnosis</th>
<th>Initial Attributes</th>
<th>Number of tests Selected by Program</th>
<th>Number of Decision Nodes Considered</th>
<th>Distribution at Terminal Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (GC)</td>
<td>7</td>
<td>7</td>
<td>151</td>
<td>CB 0.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GC 0.26</td>
</tr>
<tr>
<td>2. (OS)</td>
<td>10</td>
<td>17</td>
<td>211</td>
<td>CS 0.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OS 0.34</td>
</tr>
<tr>
<td>3. (CB)</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>CB 1.00</td>
</tr>
<tr>
<td>4. (CS)</td>
<td>10</td>
<td>5</td>
<td>148</td>
<td>CS 0.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ES 0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FS 0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OS 0.05</td>
</tr>
<tr>
<td>5. (OS)</td>
<td>10</td>
<td>3</td>
<td>139</td>
<td>OS 0.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ES 0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CS 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RC 0.03</td>
</tr>
<tr>
<td>6. (RC)</td>
<td>10</td>
<td>14</td>
<td>218</td>
<td>RC 0.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ES 0.29</td>
</tr>
<tr>
<td>7. (CS)</td>
<td>8</td>
<td>4</td>
<td>180</td>
<td>CS 0.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FS 0.26</td>
</tr>
<tr>
<td>8. (ES)</td>
<td>8</td>
<td>15</td>
<td>294</td>
<td>RC 0.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FS 0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ES 0.03</td>
</tr>
<tr>
<td>9. (ES)</td>
<td>6</td>
<td>5</td>
<td>137</td>
<td>ES 0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CS 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FS 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OS 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RC 0.04</td>
</tr>
<tr>
<td>10. (CB)</td>
<td>10</td>
<td>3</td>
<td>97</td>
<td>CB 0.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CF 0.04</td>
</tr>
</tbody>
</table>
11. (GC)  10  5  119
          GC  0.81
          CS  0.10
          ES  0.01
          CP  0.08

12. (PS)  10  3  106
          PS  0.92
          CS  0.05
          FS  0.02

Average number of tests by program = 6.6
Average number of decision nodes considered = 150
the heuristic.

This heuristic simulates to a certain extent the diagnostic strategy of one who seizes upon an initial view of the problem and later yields that view with considerable reluctance. Thus, the program makes a guess as to which tests will prove important at an early stage in the diagnosis, and thereafter restricts its attention to those tests as long as some appear to be useful. The difficulty is that the view on which the guess was made may not be an accurate one. Although the tests being considered may be of some value, there may be other tests, temporarily disregarded, which may be of greater value. Unfortunately, the heuristic is not sufficiently sensitive to changes in the current distribution, and it may cause relatively unfruitful paths to be pursued to an unnecessary extent. When it eventually abandons such a path and re-evaluates the formerly dominated tests, it may already have incurred unnecessary testing costs. The heuristic exhibits a "single-mindedness" which results in less than satisfactory performance.

In the bone tumor cases, this heuristic reduced both the average number of decision nodes considered and the average number of tests run. Here its failing is a loss of accuracy. This effect is extremely interesting. Apparently in its pursuit of an informative series of tests, the program succeeds in obscuring much of the information implicit in the initial attributes. As a result, when the undominated tests are finally released for consideration, the
current distribution is sufficiently altered that the program does not find additional tests worthwhile. This effect may be the cause of the results for case 1. Here the dominated test heuristic selected fewer tests in arriving at a less satisfactory diagnosis than the standard test selection function.

While the heuristic in question has some shortcomings, it does indicate a certain amount of promise. What it seems to lack is an awareness of changes in the current distribution which should cause certain dominated tests to be released for consideration. One possible solution is to save the current distribution with a set of dominated tests. This would allow the program to compare the present distribution with one in the stack to determine whether the view of the problem has changed sufficiently to warrant the release of the tests. This comparison could also account for the relative seriousness of states in deciding whether a given change were significant.

This example is but one of a number of heuristics which can be studied in the diagnostic system. Because very large decision trees may be encountered in future applications, a variety of tree-pruning heuristics should be studied.

Experiment 9. Exercise of the Pattern-Sorting Capability

A small example was constructed with which the pattern-sorting capability could be tested. This example consisted of six states and fifteen attributes. The matrix for the example is presented in
Table 24. The states in this example can be partitioned into two sets which have the property that certain attributes are specific to the states in a group and other attributes are shared by the two groups. The generator was employed to simulate case histories with noise attributes. That is, a case history for a state in the first group included one or more attributes selected from those specific to the states in the second group.

Consider the following diagnostic problem with the loss function as specified in Table 25. The initial attributes are $S_{10}, S_{12}, S_{13}$ and $SO_4$. These attributes cannot be attributed to a single state, and so the pattern-sorting function produces more than one pattern. In this case the patterns formed are ($SO_4$) and ($S_{10}, S_{12}, S_{13}$). For each of these patterns the distribution over states is obtained assuming that the given pattern is the only one. These distributions are:

1) ($SO_4$): DONE 0.24 DTWO 0.11 DIHREE 0.65
2) ($S_{10}, S_{12}, S_{13}$): DFOUR 0.42 DFIVE 0.02 DSIX 0.57

Based on these distributions, the pattern-sorting function selects the current pattern. Here the choice is pattern 1 although it contains only one attribute. From the loss function matrix, it can be seen that state DONE is very serious. Since state DONE can exhibit $SO_4$, the posterior probability of DONE given $SO_4$ is non-zero (0.24). By considering both posterior probabilities and losses, the pattern-
<table>
<thead>
<tr>
<th>A priori Probability</th>
<th>Disease</th>
<th>S01</th>
<th>S02</th>
<th>S03</th>
<th>S04</th>
<th>S05</th>
<th>S06</th>
<th>S07</th>
<th>S08</th>
<th>S09</th>
<th>S10</th>
<th>S11</th>
<th>S12</th>
<th>S13</th>
<th>S14</th>
<th>S15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>DONE</td>
<td>.10</td>
<td>.50</td>
<td>.40</td>
<td>.30</td>
<td>.70</td>
<td>.10</td>
<td>.20</td>
<td>.10</td>
<td>.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>DTWO</td>
<td>.15</td>
<td>.35</td>
<td>.50</td>
<td>.10</td>
<td>.20</td>
<td>.70</td>
<td>.05</td>
<td>.90</td>
<td>.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>DTHREE</td>
<td>.60</td>
<td>.20</td>
<td>.20</td>
<td>.80</td>
<td>.05</td>
<td>.00</td>
<td>.05</td>
<td>.50</td>
<td>.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>DFOUR</td>
<td>.10</td>
<td>.80</td>
<td>.10</td>
<td>.30</td>
<td>.10</td>
<td>.25</td>
<td>.30</td>
<td>.05</td>
<td>.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>DFIVE</td>
<td>.25</td>
<td>.35</td>
<td>.40</td>
<td>.15</td>
<td>.25</td>
<td>.05</td>
<td>.10</td>
<td>.60</td>
<td>.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>DSIX</td>
<td>.40</td>
<td>.35</td>
<td>.25</td>
<td>.35</td>
<td>.90</td>
<td>.25</td>
<td>.35</td>
<td>.55</td>
<td>.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE 25

Loss Function Matrix for Six State Problem
(in thousands)

<table>
<thead>
<tr>
<th></th>
<th>DONE</th>
<th>DTWO</th>
<th>DTHREE</th>
<th>DFOUR</th>
<th>DFIVE</th>
<th>DSIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>DONE</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DTWO</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DTHREE</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DFOUR</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DFIVE</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DSIX</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
sorting function selects pattern 1 as the more serious, and hence it becomes the current pattern. Tests are selected relative to this pattern, but any new attributes are processed through the entire pattern stack as discussed in Chapter 4. In this particular example, the program continued diagnosis until the following situation was obtained:

1. \((S02, S04, \text{NOT S06, S07, NOT S08, NOT S09})\)
   
   \begin{align*}
   \text{DONE} & \quad 0.92 \\
   \text{DTHREE} & \quad 0.08
   \end{align*}

2. \((S10, S12, S13, S02)\)
   
   \begin{align*}
   \text{DFOUR} & \quad 0.62 \\
   \text{DFIVE} & \quad 0.01 \\
   \text{DSIX} & \quad 0.37
   \end{align*}

The program then tentatively attributed pattern 1 to state DONE.

This left \(S10, S12, \) and \(S13\) unaccounted for. At this point, the user terminated the diagnosis. Had he wished, he could have pursued the investigation, the original pattern was shown to be invalid, the attributes in it would be returned to the unaccounted-for set and the pattern would be removed from the stack.

A variety of such experiments were run with the pattern-sorting function and the results indicated that the particular scheme embodied in the function exhibits the desired properties. This function needs to be studied more extensively, however, especially in more complicated situations. Although this area was somewhat slighted in this research the environment provided by the diagnostic system should be a good one in which to pursue such a study.
Chapter 8

DISCUSSION OF THE RESEARCH

The research discussed in the preceding chapters suggests a number of questions and issues which merit additional comment. In this chapter an attempt is made to draw together a number of results and to consider their potential generality. Also of interest here are some of the possible extensions of this research which aim at developing a more sophisticated system for the study and performance of diagnosis.

One of the more obvious questions involves the evaluation of the performance of the current diagnostic program. This question is important for two reasons. First, one of the principal hypotheses considered in this research was that in a variety of problem areas, a computer program could prove a competent or superior diagnostician. The current program has been applied to a number of cases, simulated and actual, of bone tumor and congenital heart disease. Hence a reasonable question is how well did it perform. A second reason for establishing a meaningful performance measure is so that it can be used in studies of various diagnostic strategies. If one test selection heuristic is to be judged superior to another, the judgment must be based on a measure of performance, and that measure should reflect diagnostic capability. So there is a very
real need for a good measure of diagnostic performance.

Unfortunately, while the need for a performance measure is clear, the precise nature of such a measure is open to a number of questions. Perhaps the best way to approach the problem is to catalog those qualities for which a diagnosis is generally judged to be a good one. The most obvious of these qualities is the accuracy of the diagnosis. The object of diagnosis as stated in the beginning of this thesis is to ascertain the state of a system. All other things being equal, the more accurate the determination of the state of the system, the better the diagnosis. By itself, however, this quality has relatively little meaning. One desires to know the state of a system in a diagnostic problem because this knowledge is an input to a subsequent decision (e.g., the decision about a treatment plan for a medical problem). Accuracy is not sought for its own sake, but rather for its improvement of decisions which result from the diagnosis. If these latter decisions are independent of any particular alternative in a group of diagnostic decisions, then there is no benefit to be accrued from distinguishing one of this group from another. From the point of view of further decisions, the states corresponding to these decision alternatives constitute an equivalence class. If a doctor knows that a patient has one of three viruses, all of which would be treated in the same manner, there may be no value attempting to deduce the "actual" virus.

If one were interested in accuracy as the chief quality of good
diagnosis, he could contend that in the above example, the doctor was accurate in diagnosing the problem as one of three viruses and that this can be thought of in identifying the state of the patient. A simple extension of this example makes this objection less forceful, however. Suppose that each of the three viruses are treated in a different manner and that there is a loss of diagnosing any one as another, but in each case this loss is less than the testing loss required to distinguish one from another. Again the identification of the goal of diagnosis as accuracy seems incomplete. The point is that accuracy is sought only to an extent commensurate with the expected consequences of a diagnostic decision about the system and the expected cost of obtaining greater accuracy.

This view of the diagnostic process has been the basis for this research. From the point of view of the diagnostician, the goal of diagnosis is to minimize the sum of the testing loss and the expected decision loss. Conceivably a diagnostician could correctly ascertain the state of a system at such a testing cost that his diagnosis would be judged inferior.

While it is appropriate for a diagnostician to consider expected loss for misdiagnosis as a factor in determining the course of a diagnosis this quantity is not necessarily relevant to the judgment of his diagnostic performance. The principal reason for this is that the expected loss depends on the probability distribution over states which is held by the diagnostician at the time of a terminal decision.
Since the diagnostician chooses tests, this distribution reflects his testing strategy as well as the actual problem. Basing a performance measure on expected loss ignores the relative merits of different testing strategies. It is as though a doctor were to be given a high performance rating simply because he believed very strongly that he had discovered the patient's problem. This strong belief may well be founded on incomplete or irrelevant information.

A more satisfactory way of assessing diagnostic performance is to simply add the testing loss to the actual decision loss. That is, judge the act rather than the intent. Ideally, one could determine the actual decision loss by comparing the actual state of the system (when it becomes known) with the diagnostic decision and determining the loss attributable solely to the difference between the two. By this standard, a diagnostician who consistently minimized the sum of testing and decision losses would be judged to be superior. Some of the problems inherent in this measure are rather obvious. First, the actual state of the system may never be known with certainty. A patient who is diagnosed and treated may never return for further examination, and hence a serious misdiagnosis may never be uncovered. A second problem is the difficulty in apportioning the decision loss to various diagnostic decisions. Also, the loss itself may be very difficult to ascertain. Nonetheless, this measure does seem to subsume the desired properties, and although it may be difficult to apply, it does seem to be a standard to be sought.
Another consideration in evaluating diagnostic decisions couched in terms of probabilities is the interpretation of probability distributions. For example, what are the implications of a diagnosis of (0.75, 0.25) for the states S1 and S2 for a performance measure? To a large extent, it depends on the actions which are taken based on this diagnosis. Suppose the actual state is S2. How does this affect the evaluation of this diagnosis? If only a single action can be taken on this diagnosis and it is based on the belief that the state is S1, the problem is even more difficult. The influence of such a distribution on a human decision maker may be quite subtle. If individuals react differently to such distributions, the problems will be compounded.

Finally, some effort should be made to normalize performance measures. Certain problems may be inherently more difficult to diagnose than others. For this reason, it is important to obtain an understanding of the limitations placed upon even the most expert diagnostician by the very nature of the problem before him.

The evaluation of the performance of the diagnostic program in the particular problem areas of bone tumors and congenital heart disease is made more difficult by the lack of well-defined loss structure for these problems. This precludes the use of the total loss measure discussed above. An alternative approach is to compare the program performance with standards based on the performance of experienced doctors. Even this approach is somewhat indirect in this case. Since no studies of doctor performance with the particular
case histories used were performed, no immediate comparisons based solely on the results of this research are possible. Some indication of program performance, however, can be obtained in the following way. The problems of bone tumor diagnosis and heart disease diagnosis have been studied extensively by Lodwick and Warner respectively. Both developed computer programs to perform diagnosis and have compared the performance of these programs with that of experienced physicians. These comparisons suggested that the programs performed diagnosis of a quality comparable to that of an experienced physician when all attributes were presented to both physician and program. The fact that the current diagnostic program duplicates the results of these programs on the cases studied suggests that the current program would fare equally well in a comparison with physicians. In the absence of a performance measure, this is the strongest statement which the experimental evidence will support.

If one tentatively accepts this suggestion, then a second significant conclusion can be derived from the results of these experiments. The diagnostic program was able to solve problems in two different areas of medical diagnosis. These areas differ in both the number of diseases and the complexity of inter-attribute relationships which are considered. The latter aspect is particularly important because it was handled without changing the program. Since the experiments involved only two problem areas and both were medical, the applicability of the program for a wide class of problems has
not been established. Its success in the two areas mentioned, however, strengthens the belief that it does have wider applicability.

The fact that the program is independent of the content of the information structure might be of significant value in the use of the program with hierarchical structures. Consider, for example, the problem of diagnosing a very large set of diseases. One possibility would be to create a hierarchical structure in which many sub-structures exist. The structures for bone tumors and congenital heart disease might such sub-structures. At the higher levels, the states would be classes of diseases, such as heart disease. The goal of diagnosis at higher levels would be to determine the proper class of disease. When this determination had been made, a more detailed sub-structure for that disease class would be employed for a "finer" diagnosis. The same diagnostic program could deal with all sub-structures. This would be a great improvement over a large set of programs, one for each sub-structure.

Again, considering the results of diagnosing actual case histories, one can readily appreciate the advantage of sequential diagnosis. In the particular problems studied, the program was able to arrive at a diagnosis with the use of relatively few tests. This capability is very important since the testing cost for a diagnosis may be a significant part of the total cost. Tests which are unnecessary or uninformative may exact a high price, and an effort
should be made to restrict the tests run to those essential to the
diagnosis. The sequential test selection facility permits the pro-
gram to dynamically assess the potential usefulness of each possible
test. This results in efficient testing strategies, an important
component of good diagnosis.

In a problem area in which the tests relevant to different
groups of states are relatively disjoint, the value of sequential
testing should be even greater. Once the appropriate group of states
has been established, the tests considered can be restricted to the
set of tests associated with that group. In the absence of a sequen-
tial testing capability, it may be necessary to perform all tests to
obtain information which could have been obtained from a few. The
striking reduction in the number of tests required for diagnosis of
bone tumors and congenital heart disease effected by sequential testing
strongly suggests the potential value of this approach in other diag-
nostic problems.

The existence of a diagnostic system rather than just a diagnost-
ic program has proved quite important in this research. Many of
the strategies which were considered are quite complicated, and it is
difficult to predict a priori the manner in which they will perform.
The generator has been very useful in testing these strategies under
a variety of problem conditions. Also of use has been the facility
for selectively monitoring particular diagnostic functions such as
pattern-sorting and test selection by collecting detailed data on
their operations.

One virtue of the inclusion of a generator in the diagnostic system is that it makes it possible to study the performance of the diagnostic program in problems derived from a wide range of information structures. The simulation capability frees the researcher from dependence on actual case histories. Thus he can create structures and simulated cases specifically designed to test some aspect of the diagnostic program. The use of the simulation facility with an information structure corresponding to an actual diagnostic problem may also be very useful in the study of that particular problem.

Complementing this capability is that of operating the diagnostic program in an interactive mode. Thus a user can employ the program in actual diagnostic problems. This "open end" of the system permits the independent testing of strategies developed through research, as well as making the diagnostic program a practical aid to problem solving. The experience gained in this research indicated the value of such a system which permits the study of both actual and artificial diagnostic problems. It seems that this type of system would prove most useful in further development of sophisticated strategies for computer-aided diagnosis.

Finally, the modularity of the system is very important. On the one hand, the insulation of the system functions from one another permits one to study a wide variety of diagnostic strategies since the functions can be changed independently of one another. Also as
better versions of these functions are developed, they can be incorporated into the system without restructuring it. In this sense, the performance of the system can be improved as additional experience with it is obtained.

The experience obtained with the diagnostic system has pointed to a number of areas for further research. A number of these areas are discussed here. Some pertain to specific improvements in the diagnostic capabilities of the program, while others have more general ramifications.

In Chapter 7, certain experiments to study the effect of the loss function on diagnosis were discussed. While these experiments are by no means exhaustive, they do indicate the strong effect the loss function can exert on diagnoses obtained by the program. Two major questions need to be investigated in this regard. The first is how such a loss function can be developed for a particular problem area, and the second is in what ways is diagnosis sensitive to the actual values of a loss function.

The first question is a very difficult one to answer. Assuming for the moment that the matrix form of the loss function is retained, the problem is to determine the "seriousness" of each possible misdiagnosis in some appropriate units. For example, in the context of medical diagnosis, one must answer questions such as "How serious is the diagnosis of pneumonia as influenza and vice versa?" This answer must be in such terms as to permit the comparison of a wide variety
of misdiagnoses in an orderly manner. If one considers the extreme range of consequences resulting from misdiagnoses in medicine, he can appreciate the magnitude of this task. As stated, the problem required the establishment of a common scale for such extremes as the failure to diagnose a simple cold and the failure to diagnose cancer.

In many instances, the loss for a misdiagnosis depends on many extraneous factors, such as whether a patient will return to the doctor when his symptoms persist. The loss may also depend on decisions made after the diagnosis which are difficult to predict. Compounding the problem of the loss function is the need to convert the testing loss to the same scale. In particular areas, one may be confronted with further complications in this regard. For example, the question of a loss function for medical diagnosis is also a question of whose loss function should be employed. One could answer that the loss function should be that of the patient. The loss function of the doctor, and that of society, however, are also possible answers to this question. If a diagnostic system were created for general use in medical diagnosis, questions such as these would have to be considered.

Although the problems of determining the loss function for an area as complex as medical diagnosis would be very great, they may well prove worth the effort of solution. If the value of a program for diagnosis in a given area can be clearly demonstrated to be considerable, this would be strong motivation for work on an ap-
appropriate loss function. As currently conceived, such a diagnostic program would make extensive use of losses in directing a diagnosis. These losses should reflect the best understanding of the consequences of possible decisions. In some areas, the development of a loss function might be a valuable exercise independent of the implementation of a diagnostic program. In areas where sophisticated diagnosis is currently being performed by human beings, a loss function is often implicit. The attempt to quantify this loss function may reveal inconsistencies and reveal implicit losses of questionable merit. To the extent that this situation obtains in a particular area, there is additional motivation for research into this problem.

Such research would involve investigation of means of quantifying and scaling diverse consequences as well as considerations of the best form which the loss function should take. To a large extent, a framework for these investigations has already been established. A number of workers in the areas of statistical decision theory, game theory, and economics (R21, R22) have considered many of the problems associated with the attempt to scale decision alternatives. While this work is far from complete, it does provide a reasonable basis for some of the initial studies. This whole area is rich with problems of interest and importance.

Another important area for research is the development of a diagnostic program which includes improved solutions to a number of different problems, some of which are discussed here.
As previously noted, the test selection function merits particular attention. This function serves a central purpose in the overall diagnostic strategy of the program, and as a result, significant improvements in this area would directly promote the diagnostic capability of the program. More sophisticated test selection heuristics are required if the program is to deal successfully with problems involving large numbers of decision and testing alternatives. All the test selection heuristics employed in this research is "fixed-depth" in the sense that they explore all branches away from a given decision node to a fixed depth in the decision tree. Most likely a better test selection function would explore branches to varying depth, pursuing further those branches which appeared more promising. The difficulty yet to be overcome in this regard is the establishment of some measure of "promise" for branches in the decision tree. This problem has been encountered in other applications of heuristic programming, and it can be expected that significant results in the diagnostic problem would be of more general applicability. Similarly, if powerful test selection heuristics can be developed, they might be of considerable value in a variety of sequential decision problems.

Another improvement to the diagnostic program would allow it to take advantage of various relationships among tests. For example, if one is going to perform a certain test, it may be advantageous to perform another test as well because it is inexpensive when run
in conjunction with the first test. The inclusion of more complete information about tests in the information structure might allow the program to exploit various inter-test relationships and to select groups of tests to be run during diagnosis.

The pattern-sorting function needs to be bolstered by the addition of facilities for assessing the accuracy of the attributes provided it by the user. Just as it is important to detect noise attributes, it is equally important that the presence of false information be discovered. Undoubtedly only partial solutions to this problem are possible, but additional capabilities of this kind, even if somewhat limited, would be of considerable value in applications of the program to actual diagnostic problems. For example, the program could include a means for incorporating estimates of the reliability of tests into both the pattern-sorting and inference functions.

A number of improvements can be made in the inference function of the program. One of these is the incorporation of a learning scheme within this function. Such a scheme would permit the program to learn the a priori probabilities for the various states as well as the conditional probabilities of attributes of given states. Bayesian framework provides a convenient structure within which a learning scheme can be developed. Learning of this type is especially important if the relevant probabilities vary with the specific application. For example, if the information structure for congenital
heart disease were employed in a region of the country other than that in which it was developed, the probabilities might require adjustment to reflect changes in the characteristics of the population of potential patients. The program can obtain the information required for such an adjustment from the actual diagnoses which it performs on patients from the new population provided that other means of obtaining diagnoses are available. Thus in certain applications, the diagnostic program may require a training period in which it can alter the contents of the information structure to more accurately reflect the relevant behavior of the given system. A variety of learning schemes should be investigated to develop a scheme which will be suited for this problem.

Some of the considerations involved in research of this kind are apparent at the outset. If the probabilities of interest are relatively stable, then a rather prolonged learning period may be acceptable in the hope that these probabilities will be learned accurately. On the other hand, if the probability structure of the problem is relatively dynamic, then more rapid learning may be required. One difficulty with the latter situation is that rapid learning implies a greater weighting of recent experiences and if the environment is noisy, this may lead to poor probability estimates, and hence to poor diagnosis. One possibility is to exploit the ability of the human diagnostician to perceive patterns and trends by allowing him to influence probability estimates dynamically. For
instance, a doctor might be better able to detect the early stages of an epidemic and hence adjust the a priori probability of the prevalent disease to reflect its increased incidence.

Some Comments on the Diagnostic Model

When one devotes considerable attention to the problem of diagnosis, he may experience a tendency to generalize his definition of the problem so as to encompass an increasingly wide circle of problems. The danger of this tendency is that it may result in the extensive discussion of diagnostic programs and systems of impressive capabilities which are founded more on wishful thinking than on practical experience. Because the appeal of such an intellectual exercise is strong, it is important to consider carefully the model of the diagnostic problem being employed in order to obtain a realistic view of both its potential and limitations. Some of the important characteristics of the model employed in this research are investigated here with this intention.

A diagnostic model based on attribute-state relationships has understandable appeal. In many diagnostic problems the most visible aspect of an expert's attack on a problem is his gathering of attributes on which to base his decision. In many instances he may appear to relate these attributes directly to the possible states of the system. When the difficulty of diagnostic problems in general is considered, however, it seems unlikely that the human expert performs only a simple association of attributes and states to arrive
at a diagnosis. Diagnosis, as performed by humans, seems to be a subtle and often complex process of association and deduction.

The model employed in this research, on the other hand, is very explicit in the way in which it relates attributes and states. Associations in the information structure are relatively direct, and deduction is performed in a uniform manner for all problems. In one sense, the model employed by the diagnostic program appears quite rigid and simple. Even this brief comparison with human diagnosis suggests an important question. Can this relatively simple model be sufficient for a diagnostic program to perform effectively? A derivative of this question is the following. To what extent can a program based on this model be successful in performing diagnosis in a variety of problem areas? Although the evidence gathered from this research is far from sufficient to allow definitive answers to these questions, it does permit some insights into the problems to which these questions are addressed.

The author believes that the basic functions developed in this work reflect aspects of a diagnostic program which has both potential generality and power. At present, the functions are quite crude in their structure and capabilities, but the conception of diagnosis in terms of these functions (or their more sophisticated successors) is believed to be both a useful and viable one. One problem may be that the current separation of functions is somewhat restrictive, but this has the advantage of emphasizing the principal objectives and problems of each. This emphasis is very important in the initial phases of
research in this area, and the separation permits the study of differ-
ent versions of one function more or less independently of the others.

In broad outline, the model incorporates the principal features
of diagnosis as performed by human beings. The inference function
coupled with the information structure allows the consideration of
both past experience and current information in a particular diagnosis.
Bayesian inference provides an orderly way for balancing these two
elements in the deductive process. The test selection function pro-
vides the program with a rational means for choosing tests which
accounts both for their cost and their potential value in furthering
the diagnosis. Finally, the pattern-sorting function provides a
means for performing diagnosis in the presence of noise.

While it is unlikely that the human diagnostician employs this
particular division of the diagnostic function, the total capability
incorporated in the functions seems to approximate that required.
It is also important to note that there is no particular reason to
require a diagnostic program to simulate the processes employed by
humans. A more appropriate requirement is that a diagnostic pro-
gram should allow the exploitation of the comparative advantages of
a computer in order that the total diagnostic capability of a man-
machine partnership may exceed that attainable by either above.

For example, it has been noted that doctors do not organize
their diagnostic experience into large lists of symptoms and diseases,
but rather associate their experience with and through their under-
standing of the human body and its processes. It would be extreme to conclude from this that such an organization is a necessary one for diagnosis, particularly if the diagnostician is a computer program. The fact that a doctor does not order his experience primarily in terms of attribute-disease lists may simply be evidence of the difficulty he encounters in attempting to deal with and maintain information of this form. A computer program would have less of a problem in this regard, and, in fact, this may be a useful structure to impose on the experience employed by a diagnostic program.

While in very general terms, the functions of the program correspond to those apparently required for diagnosis, there remain certain questions about limitations arising from their current realizations. In a sense these are questions about the generality of the model. Since the program was designed to solve the model diagnostic problem, it is reasonable to expect that the generality of the program will be determined by the extent to which real diagnostic problems can be described by the model. (Also, the appropriate statistical data must be available.)

For example, a major difficulty in applying the program to program debugging is developing a proper characterization of states. One can see in theory how this can be accomplished, but a practical solution would be extremely difficult. Also, an extremely useful strategy in program debugging is changing the state of the program (by
changing instructions, etc.) Here tests may very well change the state of the system. Because one can save a copy of the program, one can also use destructive testing. While one could probably change the model (and program) to reflect these possibilities, the current model does not account for them. Hence, the use of the program in this area is severely limited.

Also, there may be areas in which the diagnostic experience may not fit the statistical model employed in this work. In these areas, the inference function would have to be redone for non-Bayesian inference.

On the other hand, there seem to be a number of real problems which can be described by the model, including many machine failure and medical diagnosis problems. While the evidence is limited, the performance of the current diagnostic program in the areas of congenital heart disease and bone tumors should not be overlooked. At the very least these results must be termed promising. The model on which the program was based and the program itself were developed independently of considerations of these particular diagnostic problems, and yet the program demonstrated potential value in both areas. There seems reason to believe that other problems of medical diagnosis will also prove susceptible to such a program. The diagnostic system permits the study of alternative strategies developed in the light of such experiments, and this, too, should ease the problems of increasing the extent of its capabilities.
Some of the difficulty in applying the program to new areas can be traced more directly to a lack of adequate data for an information structure than to an inherent intractability to this approach. If continued research yields further indications of the value of a computer program for diagnosis, it may well be worth the considerable effort required to reformulate a number of diagnostic problems in terms of this model or an extension of it. Certainly, the results of this research do not preclude this possibility.
References


Appendix 1

Sample of an Input File
(STATE D01 0.05 S01 0.01 S02 0.10 ... S17 0.90)

(CLSTR D01 EXOR 0.05 S06 0.07 S07)

...

...

(ATRIB (S01 S02 S03) TEST1 S04 TEST4 ... (16 S17) TEST16)

...

...

...(TESTS TEST1 10. ... TEST16 15.)
Appendix 2

Trace of a Session with the Diagnostic Program
User responses = small letters
Program responses = capital letters

1. r system
2. NAME OF DIAGNOSTIC AREA PLEASE
3. bone tumors
4. NAME OF LOSS STRUCTURE FILE
5. bone losses
6. INFORMATION STRUCTURE ESTABLISHED
7. generate brief
8. YOU OR ME
9. me
10. HISTORY FILE
11. bone case
12. CODES
13. 3 2 2 2 3 2 3
14. NEW CASE
    WHAT ARE THE INITIAL ATTRIBUTES OF THE PROBLEM. Q.
15. s05 s07 s11 s14 s17 s20 not s21
16. CONDITIONAL PRIOR STATE PROB
    CB 0.26
    CS 0.09
    GC 0.62
    CF 0.02
    TRACE 0.01
17. ANY IDEAS. Q. TYPE 'DONE' IF SATISFIED.
18. c.r.
19. SET SEARCH DEPTH, THRESHOLD, AND HEURISTIC CONTROL

20. 1 0.10 0

21. THE TEST SELECTED IS TEST43

22. s43

23. CONDITIONAL PRIOR STATE PROB
   CB  0.55
   CS  0.04
   GC  0.37
   CF  0.04

24. THE TEST SELECTED IS TEST50

25. s50

26. CONDITIONAL PRIOR STATE PROB
   CB  0.21
   GC  0.78
   TRACE 0.01

27. GC TENTATIVE DIAGNOSIS FOR THIS PATTERN

28. CONSISTENT DIAGNOSIS FOR ALL ATTRIBUTES
Notes

A. Line 7 through line 14. The user sets controls for the run. These controls include a history file and instructions as to what information is to be collected in this file during the run (line 13).

B. Line 15. These are the initial attributes of the problem.

C. Line 16. The inference function reports the current distribution.

D. Lines 17 and 18. The user is given the option of testing his hypothesis about the problem. He declines this option (line 18).

E. Lines 19 and 20. Here the user sets the depth and threshold for the test selection function. He also chooses the standard version of this function.

F. Lines 21 and 22. The program selects a test and the user responds. This dialogue continues through line 25.

G. Line 27 and line 28. The program makes a terminal decision for the pattern. This decision accounts for all attributes and the case is completed.
Appendix 3

Listings of Diagnostic System
ROLONL.(TEMP)
N1=PODP01.(TEMP)
N2=PODP01.(TEMP)
SELST(FAULT,INIT,ADJUST.IN2)

PRINT COMMENT INFORMATION STRUCTURE ESTABLISHED.

ROLONL.(TEMP)
CODE=PODP01.(TEMP)
W$CODE.E=$DEFINER
LIST=DEFAGF.(TEMP)
Q$CODE.E=GENER8.OR; CODE.E=GEN8
GENER8.(TEMP)
PRINT COMMENT RETURN FROM GENER8.
Q$CODE.E=SCLUSTRS
LIST=CLUSTA.(TEMP)
PRINT OCTAL RESULTS LIST
E*$
MILST.(TEMP)
T'D TOP
E#N

---

DIAG - MAD
EXTERNAL FUNCTION (CONTROL)
W$R
P'T.P,SELECT,FANS,FANS1
B$N LEMPTV
INSERT FILE COMMON
EQUIVALENCE (IP,P),(FANS,ANS),(FANS1,ANS1)
R
R
R THIS FUNCTION IS THE CONTROL ROUTINE FOR THE
R DIAGNOSIS IT MANAGES THE MACRO ASPECTS OF
R THE DIAGNOSIS.
R
R
E'O DIAG.
COUNT=0
MLIST=(CELL1111)
LIST.(TEMP)
W$R STAND.E=1 OR STAND.E=3, OUTPUT.(STAND.O, BLNK)
R
R GET AND PROCESS THE INITIAL SYMPTOMS WHICH DEFINE
R THE PROBLEM.
R
W$R CBIT.E=1, P'T ILINE
W$S ILINE=OH/WHAT ARE THE INITIAL SYMPTOMS OF THE PROBLEM/*$
W$R GETSYM.(TEMP).E=0, F'N
W$R LEMPTV.(TEMP), F'N
MLIST.(PASTIK)
MLIST=(UNACDF)
MLIST.(TSTRUN)
MLIST=(SYMSTL)
MLIST.(TREF)
W$S BLNK=6/H NEW CASE/*$/
R
R PROCESS THESE SYMPTOMS TO FORM SYMPTOM PATTERNS.
R
LOOP W$R LEMPTV.(TEMP), T'D GETPAT
SYMP=PODP01.(TEMP)
TEST=BOT.(SYMP)
NEWBOT.(TEST,ISTRUN)
TEST=ITSVAL.(EXCLUS,T,TEST)
VAR.TEST,MR=O,NEWBOT.(TEST,ISTRUN)
SYRSAV.(SYMP,BINIT)
T=O LOOP
R
R
R HERE IS WHERE THE MOST SERIOUS PATTERN IS
R CHOSEN. DURING THIS ITERATION, TESTS WILL BE EVALUATED
R RELATIVE TO THIS PATTERN.
R
GETPAT
P=SELECT.(O)
OUTPUT.(CPAT,O,ENFRM)
V'S INFRM=8/H/THE CURRENT PATTERN IS.../#8
V'S CLINIC=AN/HEIGHT OF THIS PATTERN IS /5.46,2.1M,#
PDUMP1.(CPAT,PDAT)
OUTPUT.(CPAT,P,C,LBE,P)
DUMP1.(CPAT,CRST)
OUTPUT.(ALLPAT,O,OCAT)
V'S OTPT=8/H/OTHER PATTERNS.../#8
DUMPP.(ALLPAT)
R
R
R HERE CHECK THE CURRENT STATE PRIOR TO A SUCCESSFUL
R DIAGNOSIS OF THE CURRENT PATTERN.
R
R=READRV.(CURST)
TL
IF=ADVLER.(R,F)
MAG.P.O
IRANDR.(R)
T=O DQSTRT
OVR P.L.,99
T=O TL
E=I
NAME=ITSVAL.(SNMRS,CONT.(LMKL,CONT.(LPNTR.(R))>1))
OUTPUT.(STAND,1,ANSFRM,NAME)
V'S ANSFRM=8/H/THE CURRENT PATTERN IS ATTRIBUTED TO /,
1 C6=/>8
R
R CHECK FOR MORE SYMPTOMS TO EXPLAIN.
R
SUCR
GOTPAT.(O)
VAR.CODE,E=O,T=O GETPAT
OUTPUT.(STAND,0,OKYS)
V'S OKYS=8/H/CONSISTENT DIAGNOSIS FOR ALL SIGNS./#8
FAK.ORE
R
R
DQSTRT
LCOUE=O
W=FR CBIT,E=O,OR.CPRIOR,E=2
NSTATE=O
WORD=O
T=O $86K
E=I
PRINT.COMMENT.SANY.IDEAS.O. 'DONE' IF SATISFIED.
RRD
CONTINUE
R'T C6=O.WORD
V'S C6=$C6=#$
SEQDEC IS THE TEST SELECTION ROUTINE.

PRINT COMMENT BSET-DEPTH,THRESHOLD, AND HEURISTIC-CONTROL
W&R NOT. LENEMPTY.(RODONL.TEMP)
DEPTH=POPTOP.(TEMP)
THRESH=POPTOP.(TEMP)
CONTROL=POPTOP.(TEMP)

SEEK NODIES=0
STATE=STATE
W&R STANDGE.2 OUTPUT.(2,2,CFRM.DEPTH,THRESH)
SEQDEC.(TREE,0,STATE)
W&R ALLSTK.E.0 Q GETSTK
RDN=SEQLR.(ITSVAL.(VALUES$TREE))
STATE=SEQLR.(RDN,1)
IP=SEQLR.(RDN,1)
W&R WORD,NE.0
W&R STATE,E.0 DUMMY8
PRE=0N0$4
E$E
PRE=68
E1L
E$E
WORD=ITSVAL.(0$NAME$STATE)
PRE=68
E1L
OUTPUT.(ALLSTK,3,RDN,PRE,WORD,P)
V'S TRMD=S$ BEST TERMINAL DECISION AT THIS POINT IS /c3/,
1g6f, WITH EXPECTED LOSS /f8.2$4
OUTPUT.(ALLSTK,0,THEAD)
V’S THEAD=NOW TEST COST E1LOSS$/$6
TLOOP TEST=SEQLR.(RDN,1)
W&R =未
ANS=SEQLR.(RDN,1)
ANS=0$1.(TEST)
NAME=ITSVAL.(0$NAME$TEST)
OUTPUT.(ALLSTK,3,THEAD,NAME,FANG1,FANG$FANG3)
V’S TLIN=5G0,3S,4S,15S,F8.2$8
T0 TLOOP
E1L
OUTPUT.(ALLSTK,1,SCORE,NODES)
V’S SCORE=S16,H, DECISION NODES CONSIDERED.//$6
R
R SELECT THE BEST TEST
R
GETSTK W&R CONTROL.G=0

W&R WORD.E. SDSONE
OUTPUT.(STAND,0,UTERM)
V’S UTERM=SH USER TERMINATED DIAGNOSIS OF PATTERN/*$
LC->TOPTH.((TEST,STATE)
COUNT=COUNT+LC
T'0 CKS
AGAIN
TOPT.((TEST,STATE)
CKS
W.R.STATE,NE.0
LCOUNT=LCOUNT+LC
W.R.LCOUNT.L.COUNT
POPBOT.((CELL(1))
COUNT=COUNT+1
MTLIST.((ITSVASLS,(VALUES$)),TREE$)
T'0 SEEK
E'1
DECIDE
OUTPUT.((STANDARD,DF,ITSVASLS,(FILENAME$)),STATE$)
V'S DF=BC0,0/H TENTATIVE DECISION FOR THIS PATTERN./S
EQ.SUCR
0'E
NEWBOT.((TEST,ISTRUN)
TEST=ITSVASLS,(SEXCLUS$),TEST)
W.R. TEST,NE.0,NEWBOT.((TEST,ISTRUN)
E'1
R
R TEST HAS BEEN SELECTED. NOW RUN IT.
R
MTLIST.((TEMP)
W.R.LBIT.E.0,NEWTOP.((TEST,TEMP)
OUTPUT.((TEST,1,TEMP,ITSVASLS,(FILENAME$)),TEST
GEDIT.((TEMP)
W.R.LEMPY.((TEMP),T'0 AGAIN
TRES)
SYMP=POPOT.((TEMP)
SYMPASV.((SYMP,TEST)
TEST=BOT.((SYMP)
NEWBOT.((TEST,ISTRUN)
TEST=ITSVASLS,(SEXCLUS$),TEST)
W.R. TEST,NE.0,NEWBOT.((TEST,ISTRUN)
W.R HEMPY.((TEMP),T'0 TRES)
T'0 GETPAT
V'S CFROM=SH/DEPTH=1,12,H/AND THRESH=1/F4.2K
V'S TEMP=SH/ THE TEST SELECTED IS /1,C6$3
ETR

GENERB MAD
EXTERNAL FUNCTION IX)
N.R
F.2 RANGQ,OLRQ,P,PR,TESTP
B.2 DMARK,LEMPY
INSERT FILE COMMON
EQUIVALENCE (IPR,PRA)
E+0 GENERB
R
R GENERA IS THE SIMULATOR FOR THE
R DIAGNOSTIC SYSTEM.
R
LIST.((WORK)
W.R NOT-LEMPY.(X)
POPOT.((X)
NOTRMS=FAST.((CONTRL)
T'0 OKTODG
GETINF

PRINT COMMENT 6 FOR EACH OF THE FOLLOWING, RESPONDS

PRINT COMMENT 8 IF YOU WISH A CONSOLE TRACE,

PRINT COMMENT 9 IF YOU WISH A HISTORY RECORD,

PRINT COMMENT 10 IF YOU WISH BOTH, AND 101 IF NEITHER,

RDINF

PRINT COMMENT 81. CURRENT DISTRIBUTIONS

PRINT COMMENT 82. CURRENT PATTERNS

PRINT COMMENT 83. PATTERN STACKS

PRINT COMMENT 84. TESTS AND VALUES

PRINT COMMENT 85. TEST SELECTED

PRINT COMMENT 86. SIGNS OF THE PROBLEM

PRINT COMMENT 87. STANDARD INFORMATIONS

PRIVATE=POTOP(I).IRDOLNL(WORK)

CPAT=POTOP(I).WORK)

ALLPAT=POTOP(I).WORK)

ALLLIST=POTOP(I).WORK)

CTEST=POTOP(I).WORK)

SIGNS=POTOP(I).WORK)

STAND=POTOP(I).WORK)

OKTOGO

PRINT COMMENT 88

MR_FILE.NOK.0. DWRITE=(FILE,HEAD,ERRIORD,CPAT,ALLPAT,

1) ALLLIST,CTEST,SIGNS,STAND)

MR.CSIT.6.1

T'O DDIOAG

O'R DMARK

T'O START

E'T

R

R SET UP DISEASE SELECTION LIST.

R

P=O.

LIST(GENLST)

SLOOP

HSFLST=SEQLR(RDR(I))

R=SEQDRM(RDR(I))

NEXT=SEQLR(RDR(I))

T'O SLOOP

IP=ITRVAL(SPROB,NEXT)

P=P+PR

R=GENLST(NEXT)

T'O HLOOP

E'T

R

R WARN UP RANNO.

T'H RIM, FOR J=1;1;J,G.20

RANNO(RANNO(I))

R

R CONTROL LOOP FOR THE GENERATOR.

R

START

T'H GENO., FOR J=1;1;J,G.NORUNS

GLOOP

OLOOP=O.

OAT.COM.

V'S COMM=SH/CASE 1,12+8

W'R DMARK, F'O GOTTIT

TEST=RANNO(I)

GLOOP

DISEAS=SEQLR(RDR(I))

M'R.1.6.1

OUTPUT((STAND,0,BUG)

V'S BUG=SH/BUG IN GENLST/*8

CHN.COM.(10)

HERE THE USER IS IN CONTROL. SIMPLY RETRIEVE
THE NEXT SYMPTOM FROM HIM (WITH TRANSLATION
AND CHECKING). RECORD SYMPTOM AS CALLED FOR.

OUTPUT.(DS,0,FIRST)

LOOP

W'R NOT. EMPTY.(WORK)

NAME=POPTOP.(WORK)

W'R NAME.E.$NOT. OR. NAME.E.$NO

NL=POPTOP.(WORK)

W'R NAMST.(NL)

STRANS.(0)

O'E

R=SEORDR.(NL)

SL

NL*SEQLR.(R,F)

W'R F.NE.1

STRANS.(0)

T'Q SL

E'L

INTERNAL FUNCTION (DM)

T'Q STRANS.

WORD=TRANS.(NL,2)

W'R WORD.E.O, T'Q ERRMRK

NEWTOP.(-WORD,T,LST)

OUTPUT.(DS,1,NRM,NL)

F'N

E'M

O'R NAME,E.$NORMAL

R=SEORDR.(POPTOP.(WORK))

TL

NEXT=SEQLR.(R,F)

W'R F.NE.1

OUTPUT.(DS,1,NT,NEXT)

W'E NR=SM(NORMAL,C6=F)

RI=SEORDR.(ITSVAL.(SMEMBERS,TRANS.(NEXT,3)))

TLL

SYMP=SEQLR.(R,F)

W'R F.LE.1, T'O TL

NEWTOP.I=SYMP,LST

T'Q TLL

E'L

O'E

WORD=TRANS.(NAME,2)

W'R WORD.E.O, T'Q ERRMRK

NEWTOP.(WORD,LST)

OUTPUT.(DS,1,POS,NAME)

E'L

T'Q LOOP

E'L

R

R WHEN THE CURRENT LIST IS EMPTY, INITIAL SYMPTOMS
MUST BE GENERATED.

O'R CURRST.E.O

OUTPUT.(SIGNS,0,INFRM)

COUNT=RELST.(WORK,TMP)

IRALST.(TEMP)

R

R HERE THE INITIAL TESTS ARE CHOSEN AT RANDOM
R TO OBTAIN THE INITIAL SYMPTOMS.
SWITCH=0
T+1 GLOOP, FOR J=1, J<=N NINITS
W'R COUNT=0, T'0 OUT
K=0
GEOI
RDR=SEQDRD(WORK)
TEST=SEQLR(RDR, I)
K=K+1
W'R K,KTH, T'0 GETI
COUNT=COUNT+1
MEN TOP( TEST, LST )
SYNGEN(LST)
REMOV EB ( LPTR, RDR )
W'R TOP+LST=G0, SWITCH=1
GLOOP CONTINUE
OUT W'R SWITCH=G0
OUTPUT( STAND, O, NOS )
E'RE=0
E'L
R
R A RESPONSE TO A PARTICULAR TEST IS
R REQUIRED. THE TEST IS ON THE TOP OF 'LST'.
R
R
E'L
SYNGEN(LST)
R
BACK TRASH ( WORK )
F'N RET
ERRMAK OUTPUT ( STAND, O, ERR )
T'0 LOOP
R
R THIS FUNCTION SELECTS A RESPONSE AT RANDOM
R TO THE TEST ON THE TOP OF 'LST' GIVEN
R THE KNOWN DISEASE 'DISEASE'.
R
R INTERNAL FUNCTION ( X )
E'R SYNGEN,
TEST=POPTOP(LST)
W'R I$VAL=(SPTEST$TEST).E$YES$ SP$TEST=1B
O'E SP$TEST=OB
E'L
TEST=RA$NO.( $X )
POLO$0,
F'N SEQDRD( I$VAL.( $MEMBER$, TEST ) )
GLOOP NEXT=SEQLR( R, S )
W'R S,E=1
GLOOPS1 NEXT=SEQLR( R, S )
W'R S,E=1, F'N
NAME=I$VAL.( $PNAME$, NEXT )
OUTPUT( SIGNS, 1, NAM, NAME )
MEN TOP( -NEXT, LST )
T'0 GLOOPS1
O'E LOC=MEMBER,'DISEASE',I$VAL.( $MEMBER$, NEXT, 0)
W'R LOC=E0, T'0 GLOOP
PR=PIJ.( NEXT, CONT, tc$MNKR, ( CONT, ( LOC)+1 ) )
P$NEW=POLO+PR
W'A POLD.L.TESTP.AND.TESTP.L.PNEW
NAME=ITSCAL.($PNAME$,$NEXT$)
OUTPUT.(SIGNS,$1$,$POS$,$NAME$)
WRITE.(NEXT,$1$,$ST$)
W'R SPLIT
POLD=O.
T'D GLOOP
E'L
F'N
O'D
POLD=PNEW
T'D GLOOP
E'L
E'N
R

Y'S FIRST=$SH/USER RESPONSE /;C6=*$
Y'S NOS=$SH/INITIAL SIGNS ARE ALL 'NORMAL' SIGNS./;*$
Y'S ERR=$SH/SIGN NOT RECOGNIZED. IGNORED./;*$
Y'S INF=$SH/THE INITIAL SIGNS OF THE PROBLEM ARE/;*$
Y'S ARMS=$SH/OBSERVED SIGN 'NOI.' /;C6,2H=*$
E'N

PDUMP MAD
EXTERNAL FUNCTION (MARK)
R
FIT WGT.
EQUIVALENCE (WGT,WGT)
INSERT FILE COMMON
E'O DUMP.
R=M.R.ALLPAT,E,'O E'N
OUTPUT.(ALLPAT,0,BLANK)
R=SEQOR.(PAISIK)
COUNT=O
NEXT=SEQOR.(R,E)
F'E

W'R F.E.1
W'R COUNT=E-1; OUTPUT.(ALLPAT,0,ONLY)
OUTPUT.(ALLPAT,0,BLANK)
F'N

E'L
COUNT=COUNT+1
W'R NEXT=L;0; T'D LOOP
W'R NEXT=E;CURST. AND CPAT.E;ALLPAT;1; T'D LOOP
ITSCAL.(NEXT,0)
SEND.(ALLPAT,0,SYMP$)$
OUTPUT.(ALLPAT,1,CLINE,WGT)
SEND.(ALLPAT,NEXT)
T'D LOOP
F'L
R
R

Y'S BLANK=6/$*$
Y'S ONLY=$SH/CURRENT PATTERN IS THE ONLY ONE./;*$
Y'S CLINE=$SH/PATTERN WEIGHT=./;6A.2/$*$
E'N
DUMP1 MAD

EXTERNAL FUNCTION (MARK,LST)
W'R
F'S R,PDT
EQUIVALENCE (IP,P)
INSERT FILE COMMON
E'O PDOUMP1,
W'R MARK,E,O, F'N
CNT=0
R=SEGRDR,(LST)
SYMP=SELR,(R,F)
W'R F,E,1

W'R CNT,G,0, OUTPUT.(MARK,CNT,SYLIN,SARRAYS)
OUTPUT=(MARK,O,BLANK)
F+N
E'L
W'R SYMP,L,0
CNT=CNT+1
STACK(CNT)=SNOT $
E'L
CNT=CNT+1
STACK(CNT)=ITSVAL.($NAME$,SYMP),V,$000,00$
W'R CNT,G,17
OUTPUT.(MARK,CNT,SYLIN,SARRAYS)
CNT=0
E'L

T'O LOOP1
R
E'O PDOUMP1,
W'R MARK,E,O, F'N
PIOT=0
R=SEGRDR,(LST)
OUTPUT.(MARK,O,OLINE)
STATE=SELR,(R,F)
W'R F,E,1
OUTPUT.(MARK,O,OLINE,TRACE,1,-PIOT)
OUTPUT=(MARK,O,BLANK)
F+N
E'L
IP=SELR,(R,F)
W'R F,E,1,E=2, T'O LOOP1
OUTPUT.(MARK,O,OLINE,ITSVAL.($NAME$,STATE),IP)
PIOT=PIOT+P
T'O LOOP1
R

R
V'S BLANK=$/+$
V'S TRACE=$TRACES$
V'S SYLIN=$1864+$
V'S OLINESH/CONDITIONAL PRIOR STATE PROB/;/$$
V'S LINE=$205,C6,F=2/$$
E'N

OUTPUT MAD
EXTERNAL FUNCTION (MARK,NARGS,FMT,A1,A2,A3)
SELECT

R

THIS FUNCTION EXAMINES ALL THE PATTERNS IN THE
R

PATTERN STACK. IT RETURNS THE NAME OF THE
R

PATTERN WHICH HAS MINIMUM EXPECTED LOSS AS THE
R

CURRENT PATTERN.
R

UPDATE THE TREE

NEWVAL-NEXT

NEWVAL-WEIGHT

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVAL-MGT-PSAVE

NEWVA
E\*L
T\*O TLOOP

R

R THIS FUNCTION UPDATES THE UNACCOUNTED-
R FOR LIST AFTER A SUCCESSFUL DIAGNOSIS OF A PATTERN.
R IT CONTROLS THE FORMATION OF NEW PATTERNS FROM THE
R SYMPTOMS REMAINING ON THE UNACCTD LIST.
R
E\*D GOFPAT
CODE=1
RDR=SEQDR.(UNACCTD)
LOOP
SYMP=SEQLR.(RDR,1)
CHECK
W*R L=E,1
T\*O PRUNE
LOC=MEMBER.(SYMP,PATLST,0)
W*R L=G,0
W*R SYMP,G,0, CODE=0
T\*O LOOP
D*E
ADD=UNTR.(RDR)
SYMP=SEQR.(RDR,1)
REMOVE.(ADD)
T\*O CHECK
E\*L
R

PRUNE
W*R CODE,E,1, F=M
RDR=SEQDR.(PATLST)
LIST.(TEMP)
LOOP1
NULST=SEQR.(RDR,1)
W*R L=E,1
T\*O RESTOR
D*R NULST,L,0
NEWBOT=(NULST,TEMP)
D*R NULST,E,CURLST
NEWBOT=(-NULST,TEMP)
E\*L
T\*O LOOP1
R

RESTOR
RDR=SEQDR.(UNACCTD)
NLIST.(PATLST)
LOOP2
SYMP=SEQR.(RDR,1)
W*R L=E,1
INSTR.(TEMP,PATLST)
INLIST.(TEMP)
F=M
E\*L
W*R SYMP,G,0, PAFRM.(SYMP)
T\*O LOOP2
E\*N

PAFRM = MAD
EXTERNAL FUNCTION (SYMP)

R THIS FUNCTION FORMS ALL THE DISTINCT PATTERNS
R FOR A GIVEN SYMPTOM, 'SYMP'. IT PROCESSES
R ALL PATTERNS SO FORMED AGAINST THE CURRENT
R PATTERN STACK. IF THE PATTERN IS A NEW ONE, IT
IS RETAINED. OTHERWISE IT IS DISCARDED.
R
W=R
R'M SUBSET
INSERT FILE COMMON.
F'T P, UPDI
E'Q PATRN.
MEMST=ITSVAL.1$MEMBER, SYMP)
R
R PROCESS THE SYMPTOM PATTERN FOR EACH STATE ON THE
R MEMEBR LIST OF SYMP.
R
ROR=SEQROR.(MEMST)
R
LOOP
STATE=SEQLR.(RDR, I)
R=R I, 5, 1, P0

R
R CHECK FOR THIS STATE IN THE CURRENT PATTERN STACK.
R IF IT IS THERE THEN ITS SYMPTOM PATTERN MUST ALSO
R BE THERE, AND IT SHOULD BE IGNORED.
R
SEQLR.(RDR, I)
R=R MEMBER.(STATE,PATSTK,1),NE.0, T'O LOOP
R
R STATE NOT FOUND IN PATTERN STACK, SYMPTOM PATTERN
R FOR THIS STATE MAY BE A NEW PATTERN.
R GET THE 'PARTIAL SYMPTOM PATTERN' FOR THIS
R STATE GIVEN THE CURRENT SYMPTOM LIST.
R
INSCRT.(UNACTO,STATE,LIST,TEMP))
R IS THIS PARTIAL PATTERN A SUBSET OF AN EXISTING PATTERN Q.
R=SEQROR.(PATSTK)
R
CLOOP
NEXT=SEQLR.(R, F)
R=R P, NE. 1
R
W=R SUBSET.(TEMP, ITSVAL.1$SYMP, , NEXT))
R=ALST.(TEMP)
R=T'O LOOP
R=0, T'O CLOOP
EL
R
R 'TEMP' NOW CONTAINS THE PARTIAL SYMPTOM
R PATTERN. CREATE THE STATE PRIOR FOR THIS PATTERN
R AND ADD IT TO THE PATTERN STACK.
R
NULST=CONT.(NEWBOT.(LIST,9), PATSTK)+1
NULST.1$SYMP. , ITSVAL. (PATSTK)
R=SEQROR.(MEMST)
R
INLOOP
STATE=SEQLR.(RDR, II)
R=R II, 1, T'O PROC
R
SEQLR.(RDR, II)
R
W=R MEMBER,(STATE,PATSTK,1),NE.0, T'O INLOOP
R=MANY.(NULST,STATE,ITSVAL.1$PROB,STATE)
R=T'O INLOOP
R
R 'NULST' NOW CONTAINS THE STATES AND A PRIORI
R PROBABILITIES FOR THE PATTERN IN 'TEMP'.
R
PROC
P=0.
PLOOP
PRDR=SEQDR.(TEMP)
SYMP=SEQSR.(PRDR,P1)
W=PR.NE.1
UPD1.(SYMP,NULST,NULST)
T0 PLOOP
E=P
IRALST.(TEMP)
T0 LOOP
E=N

UPD - MAD
EXTERNAL FUNCTION (SYMP)

R THIS FUNCTION SUPERVISES THE UPDATING
R OF THE PATTERN STACK GIVEN THE NEW 'SYMP'.
R EACH OF THE STATE PRIOR LISTS IN THE STACK
R IS UPDATED - PROVIDED THAT THE 'SYMP' IS
R RELEVANT TO SOME STATE IN THE LIST).
R WHENEVER THE PROBABILITY OF A PATTERN GOES
R TO ZERO, THE PATTERN IS DELETED FROM THE
R PATTERN STACK.
R
R N=1
R N=EMPT, RELEV
INSERT FILE COMMON
F1 T P,UPD1
E1=UPD,
NEWBOTT.(SYMP,UNACID)
R=SEQSR.(PASTSTK)
LOOP
ST=ABS.SEQSR.(RDR,P1)
CHECK
W=ST.E=1, T0 FINISH
R
R CHECK THE RELEVANCE OF THE SYMPTOM TO
R THE PRIOR IN STLS.
R
W=ST.L, NOT. RELEV.(SYMP,STLS)
W=ST.R, L, O, NEWBOTT.(SYMP, ITSVAL, ISSYMPS, STLST)
T0 LOOP
E'=1
R
R THE SYMPTOM IS RELEVANT. USE IT TO
R UPDATE THE PRIOR.
R
P=UPD1.(SYMP,STLS,STLST)
W=PR.E=0
W=STST.L, O, UNDO.(STLST)
ADD=LPNTR.(RDR)-
STLS=SEQSR.(RDR,P1)
SYMPS=ITSVAL, ISSYMPS, REMOVE,(ADD)

PLOOP
W=PR.EMPT, SYMPS, T0 CHECK
TSYM=PRPPOP.(SYMPS)
W=PR.TSYM.G, O, PATRN,(TSYM)
T0 PLOOP
O=E
NEWVAL.(4PRDRR,P-STLST)
NEWBOTT.(SYMP, ITSVAL, ISSYMPS, STLST)
T0 LOOP
E'=1
HERE THE SYMPTOM IS PROCESSED BY TAPFAM.
TO SEE IF ANY NEW PATTERNS CAN BE FORMED.

FINISH
W'R SYMP.G.O, PATFAM.(SYMP)
F'N
E'N

UPD MAD
EXTERNAL FUNCTION (SYMP,LST1,LST2)
R THIS FUNCTION UPDATES THE STATE PRIOR
R IN LST1 TO ACCOUNT FOR THE NEW
R SYMPTOM 'SYMP'. THE SIGN OF 'SYMP' DENOTES
R THE PRESENCE OR ABSENCE OF 'SYMP'.
R 'LST2' IS WHERE THE UPDATED PRIOR IS STORED.
R
W'R
W'S EPSI=LE.A
INSERT FILE COMMON
FIJ R,P,J, EPSI,PR,PROB
EQUIVALENCE (IPROB,PROB)
EN SAME
E'O UPD
W'R LST1,LST2
SAME=1B
E1
SAME=0B
E'L
P'=D.
MEMVAR.LT(SVAL.(MEMVAR,SYMP))
R
R PROCESS EACH STATE ON THE MEMBER-LIST OF 'SYMP'.
R
RDR=SEQDRD.(LST1)
LOOP STATE=SEQLR.(RDR,I)
CHECK W'R I,E.I
W'R P,L,EPSI, E'O 0.
W'R L, E.E., E'R, P
AGAIN IPROB=ADVLR.(RDR,I)
W'R I.E.I, E'R, P
ADD=LPTR.(RDR)
SUBST.(PROB,P,ADD)
T'0 AGAIN
E'L
IPROB=SEQLR.(RDR,I)
LDR=MEMBER.(STATE,MEMLR,0)
W'R LOC.E.O
PA=0
O'E
E'PR=PJR.(SYMPCONT.(LNKR.(CONT.(LOC))+1)).
E'L
W'R SYMP,L,O
PROB=PR(l-PR)
O'E
PROB=PROB+PR
E'L
R
R CHECK FOR 'ZERO' POSTERIOR FOR THIS
R STATE. IF 'ZERO', DELETE IT FROM THE LIST.
R
W&R PROBF=EPST, T&D GRAP
P=P PROBF
W&R SAME
ADD=LPNTK(RDR)
SUBST=F PROBF, ADDF
OFF
E=L
T&D LOOP
R
R HERE IS WHERE A STATE IS REMOVED FROM 'LIST'.
R
GRAP W&R NOT SAME, T&D LOOP
ADD=LPNTK(RDR)
ADDI=LMKL(ADD)
STATE=SEQLR(RDR, 1)
REMOVE(ADD)
REMOVE(ADD)
T&D GHEGR
E=N

PIJ M&D
EXTERNAL FUNCTION (SYMP,CLUSTER)
W&R.
INSERT FILE COMMON
B=N LEMPY, NAMST
F*T PI,PZ
EQU=AMIXE (P1,P11,P2, P2)
E=T PIJ.
R
R THIS FUNCTION OBTAINS THE PROBABILITY OF SYMPT 
R 'SYMP'. 'CLUSTER' IS EITHER THIS PROBABILITY OR 
R THE NAME OF A CLUSTER WHICH CONTAINS 'SYMP'.
R
W&R NAMST, CLUSTER), F*N CLUSTER
LIST(TMP)
LIST(OPSTK)
RDR=SEQKD.(CLUSTER)
NEXT=SEQLR(RDR, 1)
LOOP W*R T=EV
R
R SHOULD NEVER GET HERE
R
F=N-1
O*R NEXT=EPAREN
T&D LOOP
O*R NEXT=RPAREN
R
R END OF A TRIPLE: PROCESS OPERATOR AGAINST 'TEMP'
R
IP1=POPTOP, (TEMP)
FIRST=PoPTOP, (TEMP)
W&R LEMPY, (OPSTK)
R
R END OF THE EVALUATION
IRALST.(TEMP)
IRALST.(DPSTACK)
PAN.P1

P2=POPTOP.(TEMP)
SECOND=POPTOP.(TEMP)
OPERA=POPTOP.(DPSTACK)
R

R PROCESS OPERATOR HERE
R

W+R OPERA.E.4OR
W R FIRST.E.1.OR.SECOND.E.1
P1=P1+P2
BMARK=1
Q*E
P1=0.
BMARK=0
E*F

BMARK=FIRST+SECOND
W R BMARK.G.1

P1=0.
BMARK=0
Q*E
P1=P1+P2
E*F
E*F

R CHECK FOR AN OPERATOR HERE
R

O R NEXT.E.4OR. OR.NEXT.E. &EXOR
NEWTOP.(NEXT,DPSTACK)
T*O LOOP

R

R PROCESS SUBCLUSTER HERE
R

Q*E
BMARK=INTERP.(NEXT)
W R SYMCH.E.0
P1=0.
BMARK=0
O R BMARK.E.0
P1=0.
O R MEMBER.(SYMP,NEXT,0).NE.0
W R SYMCH.E.1
IP=ITSVAL.(SPROB,NEXT)
Q*E
P1=1.
E*F
O*F
P1=0.

E*F
MANY.(TEMP,P1,BMARK)
T*O LOOP
YES LPAREN=$1$
YES RPAREN=$1$
E*N
EXTERNAL FUNCTION (TEST,PRIOR,LST)

NR
P=UPD1
E'O NSCOMP.
R=SEQDR.(ITSVAL,1,MEMBERS,TEST))
LOOP
SYMP=SEQDR.(R,F)
WR F,NE=1
P=UPD1,(-SYMP,PRIOR,LST)
WR P=GOV,TO LOOP
E'L
E'N

DEFINE MAD
EXTERNAL FUNCTION (TEMP)
NR
B'N LEMPTY, OPER
INSERT FILE COMMON
E'O DEFINE.
LST=0
LIST.(RDRTK)
NAME=POPTOP.(TEMP)
NUM=UFUNC(0)
T'H LOOK, FOR J=1,2,...,NUM
WR UFUNC(J),NE,NAME, TO LOOP
WR UFUNC(J+1),NE,0
PRINT COMMENT #RELATION ALREADY DEFINED. REPLACE Q.
WR ANS,NE,YES$, TO DONE
LST=LIST.(UFUNC(J+1))
TO START
E'L

LOOK CONTINUE
NUM=NUM+2
UFUNC(NUM)=NAME
LST=LIST.(UFUNC(NUM+1))
PRINT OCTAL RESULTS LST
UFUNC(0)=NUM
START
OPER=10
ARGST=POPTOP.(TEMP)
RDR=SEQDR.(TEMP)
LOOP
ELEM=SEQDR.(RDR,1)
WR I.E=1
PCOUNT=PCOUNT-1
WR LEMPTY, (RDRTK)
WR PCOUNT,G.O
PRINT COMMENT $HOT WELL FORMED. TRY AGAIN.$
UFUNC(J+1)=0
E'L
TO DONE
O'M
NEWBOT.(#$), LST
RDR=POPTOP.(RDRTK)
TO LOOP
LIST.(TEMP)
STATE=TRANS.(POPTOP.(LIST),1)
  WR STATE,E,0,T'& ERR
  NLIST=CONS.(NEWBOT,(LIST,9),STATE+)1
  NEWVAL.(SRELAT,SLCSTR,S,1,NULST)
  RDA=SEQDR.(LIST)
  NEWBOT.(LPAREN,NULST)
  SUB=O
  LOOP
    IP=SEQLR.(RDR,I)
    W*R I=E,1
    NEWBOT.(LPAREN,NULST)
    WR LEMPTY.(TEMP)
    IRALST.(TEMP)
    E'=L
    NLIST=NULST
    RDR=POPTOP.(TEMP)
    SUB=O
    D*E I=E,O
    W'R SUB,E,1
    D+L((IP)
    D*E
    NEWTOP.(ADR,TEMP)
    RDA=SEQDR.(IP)
    NEWBOT.(LPAREN,NULST)
    E'=L
    D*E IP,E,0 OR,OR,IP,E,SEXOR,便是
    NEWBOT.(IP,NULST)
    O*E
    SUB=1
    PSAVE=P
    E'=L
  T'& LOOP
  R
  INTERNAL FUNCTION (SLST)
  E=O DO1=
  SUBLST=CONS.(NEWBOT,(LIST,9),NULST)+1
  NEWVAL.(SPROB,PSAVE,SUBLST)
  NLIST=CONS.(SRELAT,POPTOP,(SLST),SUBLST)
  DORD=SEQDR.(SLST)
  DLOOP
    NEXT=SEQLR.(DRDR,DI)
    W'R DI,E,1,F'N
    NEXT=TRANS.(NEXT,2)
    W'R NEXT,E,0,T'& ERR
    LOC=MEMBER.(NEXT,STATE,0)
    REMOVE.(LOC)
    LOC=MEMBER.(STATE,ITSVAL.(MEMBER,NEXT,STATE),0)
    ADD=LNKR.(CONS.(LOC))
    SUBLST=NULST,ADD
    NEWBOT=(NEXT,SUBLST)
    T'& DLOOP
  E'N
  R
  ERR
  PRINT COMMENT $ERROR IN FORMATS
  F'N=1
  V'S LPAREN=S(1
  V'S RPAREN=S(1
  E'N
RETAC.(NEXT)
F'N STACK(0)
O'R NEXT.E.$($
NEWTOP.(SPPOINT,OPSTCK)
NEXT=SQLR.(RDR,1)
NEWTOP.(NEXT,OPSTCK)
O'R NEXT.E.$($
NEXT=POPTOP.(OPSTCK)
W'R NEXT,A77K10,E54K10
ARGF.(NEXT)
T'0 KEEP
E'7L
PUSH.(NEXT)
KEEP
SAVE DATA RDR, BASE
SAVE RETURN
BASE=POPTOP.(OPSTCK)
CODE=EVAL.1O
RESTORE RETURN
RESTORE DATA BASE, RDR
W'R CODE.LO.OR, CODE.E.$FALSE$, F'N CODE
O'R NEXT,A77K10,E54K10
ARGF.(NEXT)
O'E
PUSH.(CONST(NEXT))
E'7L
T'0 LOOP
E'7N

CONTROL MAD
EXTERNAL FUNCTION (SPOT)
N'R
INSERT FILE COMMON
E'O RETAC.
STACK(BASE)=SPOT
SPPOINT=BASE
E'7N
E'O PUSH.
SPPOINT=SPPOINT+1
STACK(SPOINT)=SPOT
E'7N
E'O POP.
SPPOINT=STACK(SPOINT)
SPPOINT=SPPOINT-1
E'7N

APRIM MAD
EXTERNAL FUNCTION (DUMMY)
N'R
STATEMENT LABEL X
B'N FIRST=SECOND, AGM ECK, BV
V'5 NRMBIT=400000000K
INSERT FILE COMMON
E'0 L.
W'R ACHECK (BSTORE) BV=FTEMP2.LE.FTEMP1
O'E BV=TEMP2.LE.FTEMP1
E'L T'O BSTORE E'O LE.
W'R ACHECK (BSTORE) BV=FTEMP2.LE.FTEMP1
O'E BV=TEMP2.LE.FTEMP1
E'L T'O BSTORE E'O EQ.
W'R ACHECK (BSTORE) BV=FTEMP1.LE.FTEMP2
O'E BV=TEMP1.LE.FTEMP2
E'L T'O BSTORE E'O GE.
W'R ACHECK (BSTORE) BV=FTEMP2.GE.FTEMP1
O'E BV=TEMP2.GE.FTEMP1
E'L T'O BSTORE E'O LE.
W'R ACHECK (BSTORE) BV=FTEMP2.GE.FTEMP1
O'E BV=TEMP2.GE.FTEMP1
E'L BSTORE RETAC (BV)
F'N E'O PLUS.
W'R ACHECK (BACK) FTEMP1=FTEMP1+FTEMP2.
O'E TEMP1=TEMP1+TEMP2
E'L T'O BACK E'O MINUS.
W'R ACHECK (BACK) FTEMP1=FTEMP2-FTEMP1
O'E TEMP1=TEMP2-TEMP1
E'L T'O BACK E'O TIMES.
W'R ACHECK (BACK) FTEMP1=FTEMP1*FTEMP2
O'E TEMP1=TEMP1*TEMP2
E'L T'O BACK E'O DIVIDE.
W'R ACHECK (BACK) FTEMP1=FTEMP2/FTEMP1
O'E TEMP1=TEMP2/TEMP1
BACK
RETAC.(TEMP1)
F'N 0
R

INTERNAL FUNCTION (X)
E'Q ACHECK.
POP.(TEMP1)
POP.(TEMP2)
WR TEMP1.E.SK+1,OR,TEMP2.E.SK+1
TEMP1.E.SK+1
BY = 18
T'0 X
E'Q
WR TEMP1.A, NRMBIT.E.0
F'N FIRST = 0B
O'E
F'N FIRST = 1B
WR TEMP2.A, NRMBIT.E.0
SECOND = 0B
O'E
SECOND = 1B
E'Q
WR FIRST.A, AND, SECOND
F'N 1B
O'E FIRST.EQ. SECOND
F'N 0B
O'E
F'N 1B
O'E FTEMP1 = TEMP1
F'N 1B
E'N
E'N

LPRIM MAD
EXTERNAL FUNCTION (DUMMY)
NR
INSERT FILE COMMON
B'N TEMP1, TEMP2, BTEST, BV
E'Q AND.
WR .NOT.BTEST(10), F'N -1
BV = TEMP1, AND, TEMP2
T'0 STORE
E'Q OR.
WR .NOT.BTEST(10), F'N -1
BV = TEMP1, OR, TEMP2
T'0 STORE
E'Q EQV.
WR .NOT.BTEST(10), F'N -1
BV = TEMP1, EQV, TEMP2
T'0 STORE
E'Q NOT.
POP.(TEMP1)
BV = .NOT.TEMP1
T'0 STORE
STORE  RETAC.(BV)
  FN 0
R
INTENDED FUNCTION (X)
E=0: BTEST
T1=STACK(SPOINT)
T2=STACK(SPOINT-1)
WR T1.E=0.OR.T1.E=1
POP.(TEMP)
0*R T1.E=INC*8
POP.(TEMP)
TEMP=1B
0*E
FIN 0B
EIL
NEXT  WR T2.E=0.OR.T2.E=1
POP.(TEMP2)
0*R T2.E=INC*8
POP.(TEMP2)
TEMP2=1B
0*E
FIN 0B
EIL
FIN 1B
EIL
EIL
SYNATS  MAD
EXTERNAL FUNCTION (DUMMY)
WR
B=NV BV
INSERT FILE COMMON
E=0: PREL
POP.(TEMP)
LOC=MEMBER.(TEMP,SYMLST,0)
WR LOC.E=0
RETAC.(MARK)
0*E
BV=TEMP.G=0
SYMCNT=SYMCNT+1
RETAC.(BV)
EIL
FIN
W=MARK=INC*8
R
E=Q ATTRIB.
R
LIST.(LST)
POP.(SYMP)
POP.(ATT)
WR SYMP.L=0
CODE=FALSE
VAL=FALSE
EIL
RY=RET
LOC=MEMBER.(SYMP,SYMLST,0)
DO LOOP
  $R P.LE.THRESH, T'O LOOP
  $YR SEQDR.(STATE)
  $YRP=SEQLR.(SYR, $Y1)

END

TOPT MAD
  EXTERNAL FUNCTION (TEST, STATE)
  $R
  EXE-EMPTY
  INSERT FILE COMMON
  EXEC LSAVE, DSAVE, LS
  EQUIVALENCE (ILSAVE, LSAVE), (ILS, LS)
  E+O-TOPFH-
  $LST-INDOQD(4)
  T'O START
  E+D-TOPF-
  SWITCH=2

START
  $R SEQDR.(ITSAV, (4 VALUEES, TREE))
  RET=O
  STATE=SEQLR.(R,F)
  LSAVE=SEQLR.(R,F)
  DSAVE=LSAVE
  ADD=LPMTR.(R)

LOOP
  TEST=SEQLR.(R,F)
  $R F.E=1, T'O END(SWITCH)
  LREF=SEQLR.(R,F)
  $R L.E.G.DSAVE
  T'O SAVE(SWITCH)
  $O=LS.l.LSAVE
  STATE=O
  LSAVE=LS
  ATEST=TEST
  ADD=LPMTR.(R)

END LOOP

SAVE(2) T'O LOOP
SAVE(1) NEWTOP(.TEST,NOGOOD)
T'Q LOOP
END(1) N'R NOT. LENTY.(NOMGOOD)
ABS-
NEWTOP.(NOMGOOD,CELL(1))
E'L.
IRALST.(NOMGOOD)
END(2) REMOVE.(N MNL.(CONT.+ADD))
REMOVE.(ADD)
FIN SET
E'N

SETUP.I AD
EXTERNAL FUNCTION (N1,N2)
N1
F'P PROB
E'Q SETUP
E'O SETUP
MANY.(STRUCT,list.(STLIST),LIST.(SYMS),LIST.(TESTS))
NUM=ITSVAL.(SHSHNUM,STRUCT)
TIM. FOR.J=1..J,J=2..JU
NEWLOT.(LIST.(0),STLIST)
NEWLOT.(LIST.(9),SYMS)
NEWLOT.(LIST.(9),TESTS)
ML CONTINUE
LIST.(TEMP)
LOOP N'R DISKLST.(ML,N2,TEMP),E.BOOM, T'Q END
WORD=POPTOP.(TEMP)
M'R WORD=E,STATE
T'Q ST
Q'R,WORD=E,SYMPSS
T'Q SYMP
Q'R,WORD=E,TESTSS
T'Q QSTL
Q'R,WORD=E,DEFINES
DEFINE.(TEMP)
Q'R,WORD=E,SPTESTS
T'Q SPS
Q'R,WORD=E,EXCLUS.
T'Q EXL
Q'S PRINT COMMENT $ERROR IN BCD TAPES
CHNSIVL.(Q)
E'L
R
R.
ST NAME=POPTOP.(TEMP)
STATE=LOOKUP.(NAME,1)
NEWVAL.(I#PROB,POPTOP.(TEMP),STATE)
STLOOP W'R LENTY.(TEMP), T'Q LOOP
SYMP=POPTOP.(TEMP)
IPROB=POPTOP.(TEMP)
VAR,PROB.E,D.. TAD..$LOOP
SYMP=LOOKUP.(SYMP,2)
MANY.(ITSVAL.1#MEMBERS,SYMP),STATE,PROB)
NEWLOT.(SYMP,STATE)
T'0 STLOOP
R
R
SYMPL
W; R LEMTY, (TEMP), T'0 LOOP
F; R KK, TOP, (TEMP), NWTOP, (TEMP, 2)
V; R KK, 0011, 0121, 00121
NAME=TOP, (TEMP)
TEST=LOOKUP, (TEMP, 2), 31
W; R HARTST, (NAME)
SYM=LOOKUP, (NAME, 3)
NEWBOT, (SYM, ITSVL, (MEMBER, TEST))
NEWBOT, (TEST, SYMP)
T'0 SYMPL
E'
REMLST=ITSVL, (MEMBER, TEST)
R; R SEORDR, (NAME)
RLOOP
SYM=SEODR, (F, F)
W; R F, EVT, T'0 SYMPL
SYM=LOOKUP, (SYM, 2)
NEWBOT, (TEST, SYMP)
NEWBOT, (SYM, REMLST)
T'0 REMOOP
TESTL
W; R LEMTY, (TEMP), T'0 LOOP
NAME=TOP, (TEMP)
COST=NWTOP, (TEMP, 2)
W; R HARTST, (NAME)
NEWBOT, (COST, LOOKUP, (NAME, 3))
T'0 TESTL
E'
R; R SEORDR, (NAME)
TLOOP
NEXT=SEODR, (F, F)
W; R F, EVT, T'0 TESTL
NEWBOT, (COST, LOOKUP, (NEXT, 3))
T'0 TLOOP
SPTST
W; R LEMTY, (TEMP), T'0 LOOP
TEST=LOOKUP, (POPTOP, (TEMP), 3)
NEWVAL, (SPTST, SYE5, TEST)
T'0 SPTST
ENDD
IRALST, (TEMP)
EAN
SYMPL1
POPTOP, (TEMP)
POPTOP, (TEMP)
T'0 SYMPL
TESTL
POPTOP, (TEMP)
POPTOP, (TEMP)
T'0 TESTL
EXL
W; R LEMTY, (TEMP), T'0 LOOP
F; T TRAN, (POPTOP, (TEMP), 3)
T1 TRAN, (POPTOP, (TEMP), 3)
NEWVAL, (SSEXCLUSS, T1, T2)
NEWVAL, (SSEXCLUSS, T2, T1)
E'N
E=N

LOOKUP MAD

EXTERNAL FUNCTION, (WORD, LCODE)
W; R
INSECT FILE COMMON
E'O LOOKUP
H LIST=0
ADD=LOCATE (.O)
W'R ADD=0
NEWBOT.(LIST(.O),H LIST)
LIST=BOT.(H LIST)
W'R LCODE.C.1, NEWVAL.(#MEMBER#,LIST(.O),LST)
NEWVAL.(#SPNAMES,WORD,LST)
E'I
F'N LST
E'O TRANS,
F'N LOCATE(.O)
INTERNAL FUNCTION (X)
E'O LOCATE.
HSNUM=VISUAL.(#HSNUMS,STRUCT)
H LIST=NTHTOP.(NTHTOP.(STRUCT,LCODE),HASH.(WORD,HSNUM)+1)
RDR=SEQRDR.(H LIST)
LIST=SEQLR.(RDR,1)
W'R I=0.E.1, F'N 0
R ITSVAL.(#SPNAMES,LST).E.WORD, F'N LST
END LOOP
E'I
E'N

INSECT MAD
EXTERNAL FUNCTION(L1,L2,L3)
W'R
W'M LEMPTY
E'O INSECT.
R
R THIS FUNCTION DETERMINES THE INTERSECTION
R OF L1 AND L2 AND PLACES THE ANSWER IN L3.
R
R RDR=SEQRDR(L1)
R LIST.(RORSTK)
LOOP ELEM=SEQLR.(RDR,1)
W'R I.E.1
W'R LEMPTY.(RORSTK)
IRALST.(RORSTK)
F'N
E'O
RDR=PORTOP.(RORSTK)
E'I
D'R I.E.0
W'R ITSVAL.(RELATS,ELEM).NE.0
NEWTOP.(RDR,RORSTK)
RDR=SEQRDR.(ELEM)
D'R MEMBER(.ABR,ELEM,L2,0).NE.0
NEWBOT.(ELEM,L3)
E'I
E'O LOOP
E'I

MEMBER MAD 12/12/66 2321.6 92 00000
EXTERNAL FUNCTION (GOAL, LST, LEVEL)

END

E0 MEMBER.
ADD=0

DESCEND
RDR=LRADDY (LST)
NAME=ADVSNR (RDR, I)
W'R I.E.1, T0 RETURN
W'R LCNR (RDR) I.LEVEL I T0-Descend

COMPAR
W'R NAME.E.GOAL, T0 FOUND
NAME=ADVNR (RDR, I)
W'R I.E.1, T0 COMPARE

ASCEND
W'R LCNR (RDR) E.O, T0-RETURN
LVLVR (RDR)
ADVNR (RDR, I)
W'R I.E.1, T0 ASCEND
T0-DESCEND

FOUND
ADD=LSPNR (RDR)
RETURN
IRARDR (RDR)
F'N ADD
E'N

UNDO MAD
EXTERNAL FUNCTION (LST)
N'R
INSERT FILE COMMON
E0 UNDO.
RDR=SEQDR, ITSVAL (ratsym$ LST)
LOOP SYMR=SEQDR, (RDR, I)
W'R I.E.1, F'N
RDR1=SEQDR, IPATSTK
LOOP1 NEXT=SEQDR, (RDR, I)
W'R I.E.1
NEWORG (SYMP, UNACTD)
T0 LOOP
D'A MEMBER (SYMP, ITSVAL (ratsym$ NEXT), Q).NE.0
T0 LOOP
D'F
T0 LOOP1
E'F
E'N

DSKR09 MAD 12/11/66 2074.2 144 00000
EXTERNAL FUNCTION (FIRST, SECOND, LST)
N'S INTEGER
INSERT FILE COMMON
O'A INT (2), NAME (1), OTHER (21)
E0 DSKLT.
V'S MODE=1
START (1)
NAME (0) = JUST (FIRST)
NAME (1) = JUST (SECOND)
BFOPEN (FILE, NAME (0), NAME (1), BUF (0), 4321).
BUF (0) = 0, ERR
MODE = 2
START (2)
BFREAD (NAME (0), NAME (1), INT (0) *** 1, EOF, EOFCT, ERR)
COUNT = LNKR (INT (0))
DREAD.(NAME(0),NAME(1),INT(COUNT) ... COUNT,EOF,EDFCT,ERR)
T'IN SWITCH, FOR I=COUNT,-1,1,1,E-0

SWITCH
OTHER(COUNT-1)=INT(I)
OTHER(I)=COUNT
X=LIST(OTHER,LST)
WR.E.E.SM NYES, T'Q START(2)
F'N K
EOF
BFCLOS.(NAME(0),NAME(1),ERR)
MODE=I
EIN DDONES

ERR
PRINT COMMENT $GOOF ON READING FILES
MODE=I
F'N DDONES
EIN

LEV RELEV

EXTERNAL FUNCTION (SYMP,LST)
N'R
EID RELEV

LOOP
TEST=ADVNR.ADR,11
W'1 I.E.1, F'N 0B
W.R MEMBER.(TEST,LST,0).NE.0, F'N 1B
T'O LOOP
EIN

LOSS MAD

EXTERNAL FUNCTION (A1,A2)
N'R
FIT PR,LOSS,MGT,SAVE,FCONS,MTOT,PI
B'M LE MPTY
EQUVALENCE (IPR,PR1,LWGT,MGT)
D'M PI(10),LOSS100,AD
VIS AD=2,1,10
W'S SUBS=0,10,20,30,40,50,60,70,80,90
B'M LE MPTY
INSERT FILE COMMON
EID SETLOS.
RETO
LIST BUFFER
DSKLST.(A1,A2,BUFFER)
SIZE=POPTOP.(BUFFER)
DSKLST.(A1,A2,BUFFER)
T'H LOOP, FOR J=1,1,J.G.SIZE
W'R LE MPTY.(BUFFER), T'O ERL
NUM=POPTOP.(BUFFER)
NAME=TRANS.(POPTOP.(BUFFER),1)
W'R NAME,E.O, T'O ERL
IPR=ITSVAL.(NAME)
PI=(NUM)=PR
LOOP
NEWVAL.(INDEX,NUM,NAME)
MAX=0
T'H LOOP1, FOR J=1,1,J.G.SIZE
IND=SUBS(J)
W'R DSKLST.(A1,A2,BUFFER).E.$DOMES, T'O ERL
### WEIGHT - MAD

**EXTERNAL FUNCTION (LST)**

```plaintext
NAR

F'T HGT, ANS, PR
EQUIVALENCE (HGT, INGT, (IPR, PR))
E'O WEIGHT.
R=SEQR=HGT
AN=0.

LOOP
STATE=SEQR(1, F)
N'R F.E.1, F=N ANS
IPR=SEQR(1, F)
INGT=ITVAL(SEIGHT, STATE)
AN=AN+RP+NGT
T'O LOOP
E'M
```

### FAST - MAD

**EXTERNAL FUNCTION (CONTR)**

```plaintext
N'R
BIN=EMPTY
INSERT FILE COMMON
END=FAST
LIST(TEMP)
PRINT COMMENT $YOU OR N$S
N'T $C$A$B, $ANS
W'N $AN$E $YOU$
CBIT=0
PRINT COMMENT $CONTROL LIST$A
RDONL(TEMP)
DEPTM=POPTOP(TEMP)
THRESH=POPTOP(TEMP)
MINIS=POPTOP(TEMP)
NOISE=POPTOP(TEMP)
CONTROL=POPTOP(TEMP)
PRINT COMMENT SCASES
RDONL(TEMP)
NUM=POPTOP(TEMP)
```

### D'IL

```plaintext
CBIT=1
E'IL
PRINT COMMENT $HISTORY FILES$
RDONL(TEMP)
W'R $NOT-EMPTY (TEMP)$
FILE=JJUST(POPTOP(TEMP))
FILE2=RJUST(POPTOP(TEMP))
ASSIGN(FILE1, BUF1, BUF2)
E'IL
PRINT COMMENT $SCODES$
RDONL(TEMP)
CPAT=POPTOP(TEMP)
CPAT=POPTOP(TEMP)
ALLPAT=POPTOP(TEMP)
ALLLST=POPTOP(TEMP)
CST=POPTOP(TEMP)
SIGNED=POPTOP(TEMP)
STAND=POPTOP(TEMP)
W'R CRIPION.E.2 AND CBIT.E.1
```
PRINT COMMENT $DEPTH,THRESH,HEURISTIC CONTROL$
DEPTH=POPTOP.$(RDEQLN+1)(TEMP)$
THRESH=POPTOP.$(ITEM)$
CONTROL=POPTOP.$(ITEM)$
E'N
IRALST.$(ITEM)$
F'N NUM
E'N

FIRMUP MAD
EXTERNAL FUNCTION (PATSTK)
N'R
R+R SUBSET
F'R PR
EQUIVALENCE (PR,IPR)
E'O FIRMUP.
PQO
R=SEQDR.(PATSTK)
PARGS=R
LOOP NEXT=SEQLR.$(R,F)$
CHECK W'R F.E.1, F'N P
CURPAT=ITSVAL.$($SYMP$,$NEXT)$
R'=PFR
LOOP1 CAN=SEQLR.$(R,1)$
W'R F.E.1
(PR=ITSVAL.$($PROB$,NEXT))
PR+PR
T'O LOOP
Q'R CAN=E.NEXT
T'O LOOP1
P'R SUBSET={CURPAT,ITSVAL.$($SYMP$,$CAN$)}
ADD=LPNTA.$(R)$
NEXT=SEQLR.$(R,F)$
REMOVE.=ADD
T'O CHECK
O'N
E'N
E'N

SUBSET MAD 12/26/66 1718.4 44 00000
EXTERNAL FUNCTION (L1,L2)
N'R
E'O SUBSET.
R='SEQDR.(L1)
LOOP NEXT=SEQLR.$(R,F)$
W'R F.E.1
F'N 1B
O'R MEMBER.(NEXT,L2,0).E.0
F'N 0B
O'N
E'N
E'N
SYMSAV  MAD
  EXTERNAL FUNCTION (SYMP,TEST)
  M'xR
  INSERT FILE COMMON
  E'G SYMSAV.
  M'xR MEMBER.(SYMP,SYMLST),0,NE,0, F'N
  MEMBER.(SYMP,SYMLST)
  UPD.(SYMP)
  F'N
  E'N
<table>
<thead>
<tr>
<th>COMMON</th>
<th>FAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUF1</td>
<td>COMMON 433</td>
</tr>
<tr>
<td>BUF2</td>
<td>COMMON 433</td>
</tr>
<tr>
<td>CBIT</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>SIGNS</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>GPAT</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>ALLPAT</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>CPRION</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>CTEST</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>ALLLIST</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>DISEAS</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>STAND</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>FILE1</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>FILE2</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>DEPTH</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>XNAMES</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>NIMITS</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>NOISE</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>NODES</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>BASE</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>TREE</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>CURLIST</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>PTLST</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>STRUC</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>SYMCNT</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>SYMLST</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>VARACTO</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>IRRUN</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>PATSTR</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>CODE</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>OPSTCK</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>STACK</td>
<td>COMMON 21</td>
</tr>
<tr>
<td>UFUNC</td>
<td>COMMON 21</td>
</tr>
<tr>
<td>ARG5</td>
<td>COMMON 11</td>
</tr>
<tr>
<td>PRIM</td>
<td>COMMON 31</td>
</tr>
<tr>
<td>CONST</td>
<td>COMMON 31</td>
</tr>
<tr>
<td>SP.INT</td>
<td>COMMON 1</td>
</tr>
<tr>
<td>NPRIM</td>
<td>COMMON 31</td>
</tr>
<tr>
<td>CELL</td>
<td>COMMON 21</td>
</tr>
</tbody>
</table>

**MACRO**  FAP

* STACK-MANAGEMENT MACROS

**PUSH**  MACRO  ARGS

- IRP    ARGS
- TXI    **+1,1,1**
- CLA    ARGS
- SIO    STACK,1
- IAP    

**PUSH**  END

**POP**  MACRO  ARGS
IRP  ARGS
CLA  STACK,1
STO  ARGS
FIN  ***TRAY1

IRP
POP
END

* LIST-READING MACROS HERE
*

SEQORD MACRO A,B
CLA  A
GET LIST HEADER
STO  B
STORE IN READER-CELL
SEQRD END
*

SEQLR MACRO A,B,C
LAG  B,4
READER-LINK
CLA  1,4
GET DATUM FOR CELL
STO  A
SAVE DATUM
CLA  0,4
ADVANCE READER
STO  B
ANA  -D700000
SET FLAG
ARS  19
SUB  *1
STO  C
SEQLR END

-NTEST - FAP
ENTRY NAMTST
NAMTST SRA  SVA,4
CLA  1,4
STO  GAND
TSX
GETMEM,4
TXH  *
STO  LIMIT
CLA  GAND
SSP  
STA  LINK
ARS  18
CAS  LIMIT
TRA  NO
TRA  **2
TRA  NO
CLA  LINK
CAS  LIMIT
TRA  NO
TRA  **1
CLA  LINK
STO  HEAD
ANA  0700000
CAS  0200000
TRA  NO
TRA  **2
TRA  NO
CLA  HEAD
ARS  10
CAS  LIMIT
TRA  NO
TRA  **1
STA ++1
CLA **
ANA =07777
CLA = LINK
TRA NO
IRA YES
NO CLA =1
IRA ++2
YES CLA =0
SVC AXT **4
IRA 2,4
CAND PZE
HEAD PZE
LINK PZE
LIMIT PZE
END

SLF FAP
* DEPTH TO OBTAIN THE BEST TEST TO RUN THE ROUTINE
* 'GROW1' IS USED TO GROW NEW BRANCHES ON THE TREE IF
* NECESSARY.

* STACK MANAGEMENT MACROS
PUSH MACRO ARG5
IRP ARG5
TXL ++1,1,1
CLA ARG5
STO STACK,1
IRP
PUSH END

POP MACRO ARG5
IRP ARG5
CLA STACK,1
STO ARG5
TXL ++1,1,1
IRP
POP END

* LIST READING MACROS HERE

SEQDRD MACRO A,B
CLA A GET LIST HEADER
STA B STORE IN READER CELL
SEQDRD END

SEQLR MACRO A,B,C
CLA 1,4 GET DATUM FOR CELL
STA A SAVE DATUM
0,4 ADVANCE READER
STA B
ANA =0700000 SET FLAG
ARS 15
SUB =1
STA C
SEQLR END
ENTRY SEQDEC
SEQDEC SXA RET+1
SXA RET+1,2
SXA RET+2,4
* INDEX REGISTER 1 IS THE POINTER TO THE TOP OF THE STACK
  ZER0,1
* INDEX REGISTER 2 IS THE LEVEL COUNTER FOR THE SEARCH
  ZER0,2
CLA* 1,4
STO LIST
CLA* 3,4
STO STATE
CLA DEPTH
SUM ONE
ALS 16
STO LTEST
TSX BITSVAL,4
TXH VALUE= VALUE LIST FOR TOP LEVEL
TXH LIST
STO VALUES

* THIS IS THE MAIN SEARCH LOOP.
* FIRST GET THE DECISION LOSS OF THE CURRENT PRIOR
LOOP CLA NODES COUNT DECISION NODES
ADD 1
ST0 NODES TSX BITSVAL,4 GET DISTRIBUTION FOR THIS NODE
TXH PRIOR
TXH LIST ST0 PLIST SAVE NAME OF PRIOR LIST
CLA STATE
ST0 DECIDE NOTERM TSX SOLOSS,4 DECISION LOSS FOR DISTRIBUTION
TXH DECIDE TXH LIST
ST0 LSAVE TXH LIST
TXH SMANY,4 SAVE DECISION VALUES IF AT LEVEL ZERO
TXH VALUES TXH DECIDE
TXH LSAVE
LTEST TXL DOWN+2,2** CHECK LEVEL AGAINST DEPTH
TXK CONTINUE2,1 RETURN

* HERE THE LEVEL IS LESS THAN THE REQUIRED DEPTH.
* THE TREE IS DEVELOPED TO THE NEXT LEVEL AND THE SEARCH
* CONTINUES.

DOWN TSX BSELST,4 GET RELEVANT TESTS FOR THIS LEVEL
TXH LIST
TXH PLIST

* PROCESS THE BRANCHES AWAY FROM THE NODE DENOTED BY "LIST".
* EACH BRANCH CORRESPONDS TO A DIFFERENT TESTING ALTERNATIVE
* AT THE NOPE DENOTED BY "LIST".
SEQRD LIST,RDR ESTABLISH READER FOR LIST
SEQRL TEST,ROR,I GET NEXT TEST
READ  CLA   I
       CAS   ONE
       TRA   NOHEAD  NOT A HEADER
TRA   **2  
TRX   **2, 2, 0
TRA   RET   END
TXI   CONTIN, 2, 1  NOT THE END OF THE ANALYSIS

* PROCESS A SINGLE TEST BRANCH HERE

NOHEAD CLA   ZERO
STO   ELOSS  EXPECTED LOSS FOR THIS TEST
TSX   $GROW1, 4  *GROW1 RESULT-LIST FOR THIS TEST
TXH   RDR
TXM   PLIST
STO   RESLIST  NAME OF RESULTS LIST
* SAVE VARIABLES HERE
PUSH (RDR, LSAVE, RESLIST, PLIST)
TSX   $MENTS, TOP, A  PUT THIS TEST ON TEST STACK
TXH   TEST
TXH   TSTMN

* PROCESS ALL POSSIBLE RESULTS FOR THE TEST CURRENTLY BEING EVALUATED

SEQRD  RE COVID, RDR, 11  READER FOR RESULTS LIST
READ  SEOQR  LIST, RDR, 11  GET LIST FOR NEXT RESULT
CLA   II   CHECK FOR HEADER
       CAS   ONE
       TRA   GOON
TRA   **2   HEADER
TRA   GOON

* ALL RESULTS FOR THIS TEST PROCESSED. RESTORE VARIABLES FOR TEST EVALUATION

PDP  (PLIST, RESLIST, LSAVE, RDR)
TSX   $POPTOP, 4  GET THE TEST NAME
TXH   TSTMN
STO   TEST
TSX   $BOIT, 4  GET TEST COST
TXH   TEST
STO   ELOSS  COMBINE WITH ELoss
STO   ELOSS
TXH   CHECK, 2, 0
TSX   $MANY, 4  SAVE VALUES IF LEVEL IS ZERO
TXH   VALUES
TXH   TEST
TXM   ELOSS
TXM   ELOSS
CHECK  CLA   ELOSS
       FSB   LEAVE  IS THIS THE BEST TO DATE
TPL   DEL   NO
CLA   ELOSS  BEST SO FAR
STO   LSANE
DEL  SEOQR  TEST, RDR, 1  REMOVE THIS BRANCH
LXD   RDR, 4
SXA   TEMP, A
TSX   $REMO, 4
TXH   TEMP
TRA   READ
PROCEDURE A SINGLE TEST RESULT HERE

SAVE VARIABLES

G O D N  PUSH  ( A D R 1 , E L O S S , L I S T )
        T R I  LOOP  2 , 1  CYCLE

FOLD THIS BRANCH BACK IN TERMS OF EXPECTED VALUE

CONTIN Pop  ( L I S T , E L O S S , A D R 1 ) RESTORE VARIABLES
        T S X  $ I T S V A L , 4
        T X H  P R O B  S T O  P R O B
        T X M  L I S T
        L D Q  L S A V E  EXPECTED LOSS
        P M P  P R O B  T R A  R E A D I

B E T  A X T  1
        A X T  2
        T R A  1

T E S T  B C I - - - T E S T
        P L I S T
        P R I O R  B C I - - - P R I O R
        T E M P
        T E S T
        R D R
        A D R 1

I
        L I S T
        E L O S S
        O N E  O C T  1
        R E S U L T
        L S A V E
        Z E R O  O C T  0
        P R O B
        P R O B  B C I - - - P R O B
        S T A T E
        D E C I D E

V A L U E S  B C I  1 , V A L U E S
        I N S E R T  C O M M O N
        C O M M O N  P A C K A G E
        E N D

U P D I  F A P

THIS FUNCTION UPDATES THE PRIOR DISTRIBUTION IN
**LIST** BASED ON THE SIGN "SWAP" THE NEW DISTRIBUTION
**IS STORED IN "LIST2"**.

ENTRY  U P D I
SUBST  MACRO  MACROS
        READER,DATUM  READER,4
LAC    0,4
ARS    18
PAC    0,4
CLA    DATUM
STQ    1,4
SUBST  END

LMDI  SCA  REI,A
       STO  SYMP
       CLA*  4,4
       STO  LST1
       CLA*  3,4
       STO  LST2
       CAS  LST1
       TRA  DIF
       TRA  SAME
DIF   STZ  SWITCH
       TRA  ++2
SAME  STL  SWITCH
       CLA  FZERO
       STO  P
       TSX  SITVAL,4
       TXH  MEMPQ
       TXH  SYMP
       STO  MEMLST
LOOP  SEQDR  LST1,RDR
      SEQLR  STATE,RDR,I
CHECK  CLA  I
       CAS  ONE
       TRA  MORE
       TRA  ++2
       TRA  MORE
       CLA  P
       FSB  FZERO
       TPL  NOZERO
       CLA  P
RET   AXT  REI,A
       TRA  *4

*  NOZERO  SEQDR  LST2,RDR
AGAIN  SEQLR  STATE,RDR,I
       CLA  I
       CAS  ONE
       TRA  ++2
       TRA  RET-1
       SEQLR  PR,RDR,I
       CLA  PR
       FDP  P
       STQ  PROB
SUBST  RDR,PROB
       TRA  AGAIN

*  MORE  SEQLR  PROB,RDR,I
       TSX  MEMBER,4
       TXH  STATE
       TXH  MEMLST
       TXH  ZERO
       TNZ  ++4
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>FZERO</td>
</tr>
<tr>
<td>STA</td>
<td>PR</td>
</tr>
<tr>
<td>TRA</td>
<td>RSTEST</td>
</tr>
<tr>
<td>TAC</td>
<td>0,4</td>
</tr>
<tr>
<td>CLA</td>
<td>0,4</td>
</tr>
<tr>
<td>TAC</td>
<td>0,4</td>
</tr>
<tr>
<td>CLA</td>
<td>1,4</td>
</tr>
<tr>
<td>STA</td>
<td>PR</td>
</tr>
<tr>
<td>TSX</td>
<td>$PI,1,4</td>
</tr>
<tr>
<td>TXH</td>
<td>$SMP</td>
</tr>
<tr>
<td>TXH</td>
<td>PR</td>
</tr>
<tr>
<td>STA</td>
<td>PR</td>
</tr>
<tr>
<td>TRA</td>
<td>SYMP</td>
</tr>
<tr>
<td>TPL</td>
<td>MULT</td>
</tr>
<tr>
<td>CLA</td>
<td>$1.E0</td>
</tr>
<tr>
<td>FSB</td>
<td>PR</td>
</tr>
<tr>
<td>STA</td>
<td>PR</td>
</tr>
<tr>
<td>TRA</td>
<td>PROB</td>
</tr>
<tr>
<td>FNS</td>
<td>PR</td>
</tr>
<tr>
<td>STA</td>
<td>PROB</td>
</tr>
<tr>
<td>FSB</td>
<td>FZERO</td>
</tr>
<tr>
<td>TPL</td>
<td>2**2</td>
</tr>
<tr>
<td>TRA</td>
<td>SCRAP</td>
</tr>
<tr>
<td>CLA</td>
<td>P</td>
</tr>
<tr>
<td>FAD</td>
<td>PROB</td>
</tr>
<tr>
<td>STA</td>
<td>P</td>
</tr>
<tr>
<td>ZET</td>
<td>SWITCH</td>
</tr>
<tr>
<td>TRA</td>
<td>2**2</td>
</tr>
<tr>
<td>TRA</td>
<td>DIFPRO</td>
</tr>
<tr>
<td>SUBST</td>
<td>RDR,PROB</td>
</tr>
<tr>
<td>TRA</td>
<td>LOOP</td>
</tr>
<tr>
<td>DIFPRO</td>
<td>TSX</td>
</tr>
<tr>
<td>TSX</td>
<td>$ANY,4</td>
</tr>
<tr>
<td>FXH</td>
<td>$ST2</td>
</tr>
<tr>
<td>TXH</td>
<td>STATE</td>
</tr>
<tr>
<td>TXH</td>
<td>PROB</td>
</tr>
<tr>
<td>TRA</td>
<td>LOOP</td>
</tr>
<tr>
<td>SCRAP</td>
<td>$SWITCH</td>
</tr>
<tr>
<td>TRA</td>
<td>LOOP</td>
</tr>
<tr>
<td>LAC</td>
<td>RDR,4</td>
</tr>
<tr>
<td>CLA</td>
<td>0,4</td>
</tr>
<tr>
<td>ARS</td>
<td>18</td>
</tr>
<tr>
<td>STA</td>
<td>ADD</td>
</tr>
<tr>
<td>CLA</td>
<td>RDR</td>
</tr>
<tr>
<td>ARS</td>
<td>18</td>
</tr>
<tr>
<td>STA</td>
<td>ADD</td>
</tr>
<tr>
<td>SEQUR</td>
<td>STATE,RDR,1</td>
</tr>
<tr>
<td>TSX</td>
<td>$REMOVE,4</td>
</tr>
<tr>
<td>TXH</td>
<td>ADD</td>
</tr>
<tr>
<td>TSX</td>
<td>$REMOVE,4</td>
</tr>
<tr>
<td>TXH</td>
<td>ADD</td>
</tr>
<tr>
<td>TRA</td>
<td>CHECK</td>
</tr>
</tbody>
</table>

233000000000
1
0
CHECK
RDR
<table>
<thead>
<tr>
<th>STATE</th>
<th>LST1</th>
<th>LST2</th>
<th>SYMP</th>
<th>MEMO - BC1</th>
<th>MEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td>ADD1</td>
<td>P</td>
<td>RA</td>
<td>PROB</td>
<td>MEMLIST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBER - FAP</th>
<th>ENTRY</th>
<th>MEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBER SFA</td>
<td>RET+4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SFA</td>
<td>RETA+1</td>
</tr>
<tr>
<td></td>
<td>STA</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>STA</td>
<td>LIST</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>INZ</td>
<td>LEVELL</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td>LIST</td>
</tr>
<tr>
<td></td>
<td>STA</td>
<td>NEXT</td>
</tr>
<tr>
<td></td>
<td>TSX</td>
<td>ONCE+1</td>
</tr>
<tr>
<td></td>
<td>RET</td>
<td>AXT</td>
</tr>
<tr>
<td></td>
<td>AXT</td>
<td>AXT+4</td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>4.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL1</th>
<th>SEQR</th>
<th>LIST</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOOP</td>
<td>SQR</td>
<td>NEXT</td>
<td>ADDR</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>ONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>GOOM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>AXT+2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>GOOM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>TSX</td>
<td>ONCE</td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>LOOP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ONCE</th>
<th>SEQR</th>
<th>NEXT</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLOOP</td>
<td>SQR</td>
<td>CAND</td>
<td>R,F</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>ONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>MORE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>AXT+2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>MORE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>MORE</td>
<td>CLA</td>
<td>CAND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CAS</td>
<td>GOAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRA</td>
<td>OLOOP</td>
<td></td>
</tr>
</tbody>
</table>
TRA   **2
TRA   ODDP
LAC   R,4
GLA   0,4
ARS   10
ANA   =:D7777
TRA   RET

CANL
NEXT
RDR
1
R
F
GOAL
LIST
ONE
CFT   1
END

UN
FAP
ENTRY   UPDI
ENTRY   NSCOMP
INSERT   MACROS
SUBST   MACRO   READER, DATUM
LAG   READER,4
CLA   0,4
ARS   10
PAC   0,4
GLA   DATUM
STD   1,4
SUBST   END

HCHECK   MACRO   LAB1, LAB2, FLAG
CLA   FLAG
GAS   ONE
TRA   LAB1
TRA   LAB2
TRA   LAB1
HCHECK   END

* "UPDI" DOES THE STANDARD UPDATE OF LST1 INTO LST2
* WHEN A SYMPTOM IS THE "AGENT".

UPDI
STI   INDIC
RIR   17
TRA   START

* "NSCOMP" DOES THE NORMAL UPDATE WITH A TEST AS THE AGENT.

NSCOMP
STI   INDIC
RIR   17
STI   1
START   SXA   RET+4
SXA   RET+1,1
CLA   1,4  "AGENT"
STO   AGENT
CLA   2,4  FIRST LIST
STO   LST1
CLAD 3, 4
STO LST2
CAS LST1
TRA ++2
STA 2
CLA ZERO
STO P
TSX 8554
THX MEMO
THX AGENT
STO MEMLST

* PROCESS EACH STATE ON LST1 *

SEQRDR LST1, RDR
LOOP SEQLR STATE, RDR, I
CHECK HCHECK MORE, NORM, I

CLA P
RET AXT **, 4
AXT **, 1
LDI INDIC
TRA 4, 4

NORM SEQRDR LST2, RDR
AGAIN SEQLR STATE, RDR, I
HCHECK DIIV, RET-1, I
DIV SEQLR PR, RDR, I
CLA PR
FDI P
STO PROB
SUBST RDR, PROB
TRA AGAIN

MORE SEQLR PROB, RDR, I
RTF 1 TEST PROCESS SWITCH
TRA NC 'NSCOMP'
TSX GEPTP, 1 GET P(AGENT/STATE)
THX MEMLST

THX PR
CLA AGENT
TPL MULT
CLA =1, 60
FSD PR
STO PR
MULT CLA =1, E-6
THX CHECK FOR 'ZERO' PROB
FSD PR
THI OK
RTST RTF 2
TRA SCRAP
TRA LOOP
OK LDO PROB
FDP PR
STO PROB
CLA P
FAD PROB
STO R
THX PROB
STO 2 AGAIN TEST LISTS
THX SAME
TSX $MANY, 4
\texttt{TXH \hspace{0.5cm} LST2} \\
\texttt{TXH \hspace{0.5cm} STATE} \\
\texttt{TXH \hspace{0.5cm} PROB} \\
\texttt{TRA \hspace{0.5cm} LOOP} \\
\texttt{SAME SUBST RDR,PROB} \\
\texttt{TRA \hspace{0.5cm} LOOP} \\
\texttt{*} \\
\texttt{NG \hspace{0.5cm} CLA \hspace{0.5cm} \#1.0} \\
\texttt{STD \hspace{0.5cm} PR} \\
\texttt{SEQDR \hspace{0.5cm} MEM,LIST,4 \hspace{0.5cm} READ MEMBER LIST OF ISSY} \\
\texttt{NLODP SEQLR AGENT,R,F \hspace{0.5cm} NEXT SIGN} \\
\texttt{HCHECK \hspace{0.5cm} GOON,MULT,F \hspace{0.5cm} MEMBER LIST OF SYMP} \\
\texttt{GOON TSX \hspace{0.5cm} SITVAL,4 \hspace{0.5cm} MEMBER LIST OF SYMP} \\
\texttt{FXH \hspace{0.5cm} MEMO} \\
\texttt{TXH \hspace{0.5cm} AGENT} \\
\texttt{STG \hspace{0.5cm} SYMEM} \\
\texttt{TSX \hspace{0.5cm} GETP,1} \\
\texttt{FXH \hspace{0.5cm} SYMEM} \\
\texttt{TXH \hspace{0.5cm} TEMP} \\
\texttt{CLA \hspace{0.5cm} PR} \\
\texttt{FSB \hspace{0.5cm} TEMP} \\
\texttt{STD \hspace{0.5cm} PR} \\
\texttt{FSB \hspace{0.5cm} \#1.0 \hspace{0.5cm} TEST FOR ZERO} \\
\texttt{TPL \hspace{0.5cm} NLODP} \\
\texttt{TRA \hspace{0.5cm} RTEST} \\
\texttt{*} \\
\texttt{GET THE PROBABILITY OF A SIGN GIVEN A SYMP} \\
\texttt{*} \\
\texttt{GETP CLA \hspace{0.5cm} \#1.1} \\
\texttt{STG \hspace{0.5cm} HOLD} \\
\texttt{TSX \hspace{0.5cm} MEMBER,4} \\
\texttt{TXH \hspace{0.5cm} STATE} \\
\texttt{TXH \hspace{0.5cm} HOLD} \\
\texttt{TXH \hspace{0.5cm} JEO} \\
\texttt{TIZ \hspace{0.5cm} \#44 \hspace{0.5cm} FOUND} \\
\texttt{CLA \hspace{0.5cm} ZERO} \\
\texttt{BACK} \\
\texttt{STG \hspace{0.5cm} 2.1 \hspace{0.5cm} STORE RESULT} \\
\texttt{TRA \hspace{0.5cm} 3.1} \\
\texttt{PAC \hspace{0.5cm} 0.4 \hspace{0.5cm} GET PROB CELL} \\
\texttt{CLA \hspace{0.5cm} 0.4} \\
\texttt{PAC \hspace{0.5cm} 0.4} \\
\texttt{CLA \hspace{0.5cm} 1.4} \\
\texttt{STG \hspace{0.5cm} HOLD} \\
\texttt{STA \hspace{0.5cm} RIGHT} \\
\texttt{ARS \hspace{0.5cm} 18 \hspace{0.5cm} FAST CHECK FOR NAME} \\
\texttt{ANA \hspace{0.5cm} 077777} \\
\texttt{CAS \hspace{0.5cm} RIGHT} \\
\texttt{TRA \hspace{0.5cm} NONAM \hspace{0.5cm} NOT A NAME} \\
\texttt{TRA \hspace{0.5cm} \#2} \\
\texttt{TRA \hspace{0.5cm} NONAM \hspace{0.5cm} POSSIBLY A NAME} \\
\texttt{TSX \hspace{0.5cm} \#P12,4 \hspace{0.5cm} AGENT} \\
\texttt{TXH \hspace{0.5cm} HOLD} \\
\texttt{TRA \hspace{0.5cm} BACK} \\
\texttt{NONAM CLA \hspace{0.5cm} HOLD} \\
\texttt{TRA \hspace{0.5cm} BACK} \\
\texttt{*} \\
\texttt{SGRAP LXD \hspace{0.5cm} RDR,4} \\
\texttt{SKA ADD,4} \\
\texttt{SEQDR STATE,RDR,1} \\
\texttt{LXD RDR,4}
<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXA</td>
<td>ADDI, 4</td>
</tr>
<tr>
<td>TSX</td>
<td>REMOVE, 4</td>
</tr>
<tr>
<td>TXH</td>
<td>ADD</td>
</tr>
<tr>
<td>TSX</td>
<td>REMOVE, 4</td>
</tr>
<tr>
<td>TXM</td>
<td>ADD</td>
</tr>
<tr>
<td>TRA</td>
<td>CHECK</td>
</tr>
</tbody>
</table>

- SYMEN
- ADD
- ADDI
- INDIC
- RDR
- L
- F
- STATE
- LST1
- LST2
- RIGHT
- HOLD
- TEMP
- MEMLST
- MENO BCI 1 MEMBER
- ZRDG OCT 0
- AGENT
- P
- PR
- PROB
- ONE OCT 1

END
Biographical Note

George Anthony Gorry was born in Glens Falls, New York on November 16, 1940. He attended public schools there, graduating from Glens Falls High School in June, 1958. He entered Yale University in September, 1958, where he studied chemical engineering. He received a Bachelor of Engineering degree with high honors in June, 1962. He entered the University of California at Berkeley in September, 1963, and received a Master of Science degree in chemical engineering in September, 1963. In September, 1963, he entered the Sloan School of Management at M.I.T. In September, 1965, he was married to the former Lucinda Jean Paulsen of Belmont, Massachusetts.

Mr. Gorry joined the staff at M.I.T. as a teaching assistant in the Sloan School in September, 1964, and was appointed as Instructor in Management in July, 1965. He has taught courses in operations research and heuristic programming. During the summer of 1964, Mr. Gorry worked at Project MAC, and in the summer of 1965, he became associated with the Boston Programming Center of the IBM Corporation. Since 1966, he has been a consultant to a number of organizations concerned with computer technology.
CS-TR Scanning Project
Document Control Form

Report # LCS-TR-44

Date: 12/14/95

Each of the following should be identified by a checkmark:

- Artificial Intelligence Laboratory (AI)
- Laboratory for Computer Science (LCS)

Document Type:

- Technical Report (TR)
- Technical Memo (TM)
- Other: ____________________________

Document Information

Number of pages: 254

Not to include DOD forms, printer instructions, etc.; original pages only.

Intended to be printed as:

- Single-sided or
- Double-sided

Print type:

- Typewriter
- Offset Press
- Laser Print
- InkJet Printer
- Unknown
- Other: ____________________________

Check each if included with document:

- DOD Form
- Funding Agent Form
- Spine
- Printers Notes
- Cover Page
- Photo negatives
- Other: ____________________________

Page Data:

Blank Pages (by page number): ____________________________

Photographs/Tonal Material (by page number): ____________________________

Other (note description/page number):

Description: Page Number:

IMAGE MAP (1-254) UNNUMBERED TITLE PAGE 1-IX UNNUMBERED BLANK

(255-260) SCANNER, COVER, DOD, TREATS(3)

Scanning Agent Signoff:

Date Received: 12/14/95  Date Scanned: 12/19/95  Date Returned: 12/28/95

Scanning Agent Signature: ____________________________
A System for Computer-Aided Diagnosis

Ph.D Thesis, Sloan School of Management, June 1967

Gorry, George A.

September 1967

Office of Naval Research, Nonr-4102(01)
Project No. NR 048-189
ER 003-09-01

Distribution of this document is unlimited.

This thesis describes a model diagnostic problem and a computer program designed to deal with this problem. The model diagnostic problem is an abstract problem. A major contention of this thesis, however, is that this problem subsumes the principal features of a number of ostensibly different real diagnostic problems including certain problems of medical diagnosis and the diagnosis of machine failures. A second major contention of this thesis is that strategies for the solution of the model diagnostic problem can be formulated in terms sufficiently explicit to permit their incorporation in a computer program.

The diagnostic program was implemented on the time-sharing system at Project MAC. It was applied to two medical problems, the diagnosis of congenital heart disease, and the diagnosis of primary bone tumors. The results obtained here suggest 1) that a computer program can be of considerable value as a diagnostic tool, and 2) that it is quite advantageous for such a program to perform sequential diagnosis as it interacts with the user.

Computers  Multiple-access computers  Real-time computers
Computer-aided diagnosis  On-line computers  Time-sharing
Machine-aided cognition  On-line diagnosis  Time-shared computers
Scanning Agent Identification Target

Scanning of this document was supported in part by the Corporation for National Research Initiatives, using funds from the Advanced Research Projects Agency of the United States Government under Grant: MDA972-92-J1029.

The scanning agent for this project was the Document Services department of the M.I.T Libraries. Technical support for this project was also provided by the M.I.T. Laboratory for Computer Sciences.

Scanned
Date: 12/19/1995
M.I.T. Libraries
Document Services