Using Experience in Learning and Problem Solving

by

Phyllis Koton

S.B, Massachusetts Institute of Technology (1980)
S.M., Massachusetts Institute of Technology (1983)

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Philosophy

at the

Massachusetts Institute of Technology

May, 1988

Signature of Author___
Department of Electrical Engineering and Computer Science
May 5, 1988

Certified by_
Peter Szolovits
Associate Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by___
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Using Experience in Learning and Problem Solving

by

Phyllis Koton

Submitted to the Department of Electrical Engineering and Computer Science
on May 7, 1988
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

The problem-solving performance of most people improves with experience. The performance of most expert systems does not. People solve unfamiliar problems slowly, but recognize and quickly solve problems that are similar to those they have solved before. People also remember problems that they have solved, thereby improving their performance on similar problems in the future. The thesis describes a system, CASEY, that uses case-based reasoning to recall and remember problems it has seen before, and uses a causal model of its domain to justify re-using previous solutions and to solve unfamiliar problems.

CASEY overcomes some of the major weaknesses of case-based reasoning through its use of a causal model of the domain. First, the model identifies the important features for matching, and this is done individually for each case. Second, CASEY can prove that a retrieved solution is applicable to the new case by analyzing its differences from the new case in the context of the model. CASEY overcomes the speed limitation of model-based reasoning by remembering a previous similar case and making small changes to its solution. It overcomes the inability of associational reasoning to deal with unanticipated problems by recognizing when it has not seen a similar problem before, and using model-based reasoning in those circumstances.

The techniques developed for CASEY are shown to result in solutions identical to those derived by a model-based expert system for the same domain, but with an increase of several orders of magnitude in efficiency. Furthermore, the methods used by the system are domain-independent and should be applicable in other domains with models of a similar form.

Thesis Supervisor: Peter Szolovits

Title: Associate Professor, Electrical Engineering and Computer Science

Acknowledgements

I would like to thank the following people who helped make this thesis possible:

My thesis supervisor, Peter Szolovits, for seven years of support and good advice, for encouraging me to keep going when I was ready to give up, and for giving me the intellectual freedom to let my ideas develop and grow.

Ramesh Patil, who as teacher, reader, and friend, taught me about artificial intelligence, medicine, engineering, and life in general.

William Long, whose knowledge, good nature, and patience inspired me, and whose Heart Failure program is a challenging and impressive body of work that I was extremely fortunate to have as a resource for my own research.

Patrick Winston, who did exactly what he promised, and whose comments during my thesis defense made it quite a pleasant experience.

Janet Kolodner, to whom I owe an immense intellectual debt, for reading this thesis although under no official obligation to do so, and for supporting me in so many ways.

Robert Jayes, who provided the test cases for my program.

The members of the Clinical Decision Making Group, past and present, especially Tom Russ, Elisha Sacks, Mike Wellman, Alex Yeh, Robert Granville, and Isaac Kohane, for many years of camaraderie, peer review, and good ideas, and Nomi Harris for making our office such a fun place.

Paul Resnick, for some very illuminating discussions.

My friends David Goddeau and Kent Pitman for all-around good advice since the time we were undergraduates.

And of course, my family: my daughter Jaclyn, my husband Ben (a merci-

less editor), and my parents, without whose love, support, time, and sacrifices I could never have finished this thesis.

The work reported herein has been supported in part by National Institutes of Health grants R01 LM 04493 from the National Library of Medicine and R01 HL 33041 from the National Heart, Lung, and Blood Institute.

Contents

Chapter 1

Introduction

The problem-solving performance of most people improves with experience. The performance of most expert systems does not. People solve unfamiliar problems slowly, but recognize and quickly solve problems that are similar to those they have solved before. People remember problems that they have solved, thereby improving their performance on similar problems in the future. People also learn from their mistakes. Research in artificial intelligence has resulted in techniques that exhibit some of these capabilities. Associational reasoning solves common problems quickly. Model-based reasoning[1] can be used to solve unfamiliar problems, but it does so slowly. Memory-based reasoning [22] techniques can be used to remember previously solved problems and to learn from experience. However, no current system demonstrates all three capabilities. A reasoning system that (1) used associational reasoning for efficiency, (2) used model-based reasoning for robustness, and (3) learned from experience, could combine the advantages of each technique while complementing their individual limitations. Such a method would represent a substantial enhancement of current technology. This thesis presents the theory, implementation, and evaluation of such a system, CASEY.

1.1 Background

Much of the recent research in artificial intelligence has been directed towards the development of high-performance, domain-specific problem solving systems, called *expert systems* or *knowledge-based systems.* Such systems can be

[1] by which I mean reasoning from a causal model of some domain.

classified according to the type of reasoning used by the program.[2] The vast majority of current expert systems rely on *associational* reasoning (associating data with solutions via heuristics, empirical associations, or "rules of thumb"). The alternative approach, which solves problems by reasoning about a model of the behavior of objects in the domain, is known as *model-based reasoning*.[3] Each approach has its advantages and disadvantages, but neither approach allows expert systems to learn from experience. Although work in machine learning has developed techniques that allow computer programs to learn, barring few exceptions (e.g., AQ11 [31]) these techniques have not been applied to expert systems. Furthermore, this work has concentrated on the development of rule sets through training examples, after which learning ceases.

People seem to use both associational and model-based reasoning. For familiar problems, we use associational reasoning, taking advantage of the speed of this approach. When confronted with unfamiliar or difficult problems, people can refer to a more detailed knowledge base, much like the type used by model-based systems. The human ability to exploit both types of reasoning requires us to (1) recognize a new problem as being of a type we have encountered previously, and to (2) constantly update our knowledge; that is, to learn from experience. Current knowledge-based systems rely on knowledge painstakingly compiled from human experts, a process that is time-consuming

[2]This dichotomy had previously been identified as "shallow vs. deep" *knowledge*. However, the difference is in the method of *reasoning*, since the distinction between deep and shallow knowledge is relative [25], and deep knowledge can be employed with techniques traditionally considered shallow [15].

[3]Also known as "reasoning from first principles" [9].

and labor-intensive. When faced with the same problem twice in succession, they work just as hard to solve the problem the second time. The development of a technique that integrates associational and model-based reasoning with the ability to learn from experience could result in improved system performance.

1.2 Associational vs. model-based reasoning

Associational reasoning reduces long chains of inferences in the underlying "deep" knowledge to shorter, often uncertain, links between data and solutions. This approach has the advantage of efficiency, because the alternative of following all of the intermediate links and choosing among alternate paths in the problem space can be slow and is often unnecessary. However, programs using associational reasoning have their limitations. Because such programs solve problems by matching the current situation against a set of predetermined situations, the knowledge base must *anticipate* situations that may arise. If the program is presented with an unanticipated, peripheral, or difficult problem, it may be unable to solve it [8] or worse, appear to solve it but yield a solution that is incorrect [24]. Also, associational knowledge typically must contain many implicit assumptions. For a complicated domain, it might be infeasible or impossible to explicitly enumerate the exact conditions under which the knowledge is applicable. Such systems, therefore, cannot ensure that their knowledge will be applied correctly.

Models provide a different kind of knowledge for reasoning in many domains. Knowledge about the domain that might be excluded from an associational reasoning system is often explicitly represented in the model. Models are typically combined with a general reasoning method, such as simulation or search, affording the model-based system more flexibility than an associational system for the same domain [9], [24], [44]. However, the more explicit knowledge and more general problem solving method creates longer inference chains. For this reason, model-based systems are slower, more complicated, and less widely employed than associational systems. Also, if the relationships in the model are uncertain, long inference chains may generate too much uncertainty to draw conclusions. Associational reasoning allows the relationships to be summarized at a manageable level of uncertainty.

There have been a few previous attempts to combine associational reasoning with model-based reasoning. ABEL [34], a program for diagnosing acid-base and electrolyte disturbances, maintained a description of a patient's illness at five levels of detail. The least-detailed level represented associational knowledge and the more-detailed levels were used for model-based reasoning. However, rather than *choosing* when to solve a problem using associational reasoning and when to use model-based reasoning, ABEL always reasoned about the patient at every level of detail. GORDIUS [45] combined associational reasoning and reasoning from a causal model for hypothesis generation in the geology domain. It also was incapable of deciding when to use each type of knowledge. It always used its associational rules to generate hypotheses, and always used its causal model to test proposed hypotheses.

1.3 Using past problem-solving experience

The ability to identify similar problems, recall previous problems, and store newly-solved problems could enhance a knowledge-based system's performance in several ways. Common problems could be solved more efficiently because the system could recognize that it already knew how to solve them and apply previously derived solutions. By remembering problems after it solved them, the system could continually increase the collection of problems that it knows how to solve. The system could also modify its knowledge by allowing the user to override the program's solution, and remembering the solution that the user preferred.

There have been several machine learning techniques developed that allow identification and recall of similar problems, for example case-based reasoning [23], memory-based reasoning [47], and derivational analogy [7]. These paradigms all rely on a memory of previously solved cases. Case-based reasoning and derivational analogy have the same basic framework when presented with a new problem. The programs recall a previous solution, adapt it to the current problem, and remember the new problem and its solution. Memory-based reasoning is used to remember a similar previous problem, but does no adaptation. These paradigms are fundamentally associational: they associate features of a problem with a previously-derived solution to that problem. However, neither case-based reasoning nor memory-based reasoning have been used with a strong causal model, and so their adaptations of previous solutions are basically *ad hoc*. Derivational analogy goes to the other extreme: it is so careful about justifying its use of each step in a previous solution that it loses the efficiency advantage of associational reasoning. Winston's work on analogy [50, 51, 52] uses the causal explanation of a previous situation to produce a solution for a new problem. However, this work does not address the issues of remembering, determining the applicability of, and choosing among previous similar problems.

Case-based reasoning was the most applicable to CASEY's goals of combining associational reasoning, model-based reasoning, and learning from experience. By their ability to match the features of a new problem against a memory of previously-solved problem, case-based reasoning systems achieve the efficiency of associational reasoning. If no previous case is recalled, it could serve as a signal that the problem is unfamiliar to the program and that model-based reasoning should be used. By their ability to remember new problems and their solutions, case-based reasoning systems continually increase their collection of easily solved problems. Most importantly, as several similar cases are solved, most programs that use case-based reasoning (e.g., citeKolo, [46], [48]) make and remember generalizations about the problems that they have solved and the solutions to these problems. These generalizations represent ***new associational knowledge which links the common features of a group of problems with a solution to that type of problem.***

Until now, case-based reasoning has been applied only to domains without a strong causal model (e.g., SHRINK [21] in psychiatry, MEDIATOR [46] in dispute mediation, PERSUADER [48] in labor negotiations, JUDGE [5] and HYPO [4] in legal reasoning, PLEXUS [2] in real-world planning, SWALE [16] in newspaper story explanation). The lack of an explicit causal model gives case-based reasoning programs a problem commonly seen in other associational reasoning systems: they cannot ensure that their knowledge will be applied correctly. There is one underlying reason for this: without an explicit causal model, case-based reasoning programs depend exclusively on coincidence in selecting similar previous problems[4] and in making generalizations. A second problem, also seen in associational reasoning systems, is that when an adequate match is not found, case-based reasoners are unable to fall back on model-based reasoning and must still use the best match available to arrive at a solution. A consequence of these two limitations is that a retrieved solution sometimes leads a case-based reasoner down the wrong path. A previous case-based reasoning program which did use a causal model was CHEF [13], a planning program in the domain of cooking. CHEF's causal model was extremely simple. Moreover, its causal reasoning consisted solely of chaining rules backward from an observed failure to a cause. This approach could not scale up to a reasonably sized domain. Furthermore, his causal model was not used to derive a solution *de novo.*

Integrating associational, model-based, and case-based reasoning results in a program which has the strengths of each approach while compensating for their weaknesses. The model-based reasoning component solves complicated and unfamilar problems, and releases the case-based component from its dependence on coincidence. The case-based reasoning component uses associational knowledge to recognize problems that the system already knows how to solve, and allows the constant creation of new associational knowledge by the program. The combination is synergistic.

[4]memory-based reasoning programs also have this drawback.

1.4 The domain of medical decision making

As a complex real-world domain, medical decision making is particularly well-suited as a testbed for combining associational reasoning, model-based reasoning, and learning techniques. Medical decision making involves an experiential component as well as reasoning from causal models. Physicians start with a large basic and clinical science knowledge base. Then, the accumulation of cases seen over a physician's career improves his day-to-day problem-solving ability. Making generalizations about previous patients lets a physician make predictions about future similar patients; remembering how an unusual past case was resolved can be helpful the next time a similar case is seen. However, when a good physician confronts an unfamiliar problem he refers to his knowledge of pathophysiology – his model.

Medical reasoning is more challenging than some other diagnosis domains that typically deal with "single faults" and have an underlying model that is small and well-characterized (e.g. digital circuit diagnosis). The models used in the medical domain are often large and complex. They are incomplete and therefore uncertain. Medical problems can include multiple interacting diseases with partially overlapping symptoms, which are problematic for many diagnosis programs.

For these reasons, the ideas developed for CASEY were tested in the domain of managing patients with heart failure. The techniques do not depend on any specific domain information and therefore should be applicable to other domains with similarly designed models.

1.5 A simple example

The input to CASEY is a description of a patient. CASEY produces its solutions using a memory of cases that it has already solved and a causal model of the cardiovascular system. CASEY's output is a causal explanation of the patient's symptoms. The causal explanation relates items in the patients description to states in the model. This section gives a simple example of CASEY's operation.

A new patient, Uri, is presented to the program. Uri is a 67-year-old male with dyspnea (shortness of breath) on exertion and a history of anginal chest pain. His blood pressure is 135/80, his heart rate is 87, his respiration rate is 14, and his temperature is 98.4. His chest x-ray reveals aortic valve calcification. The rest of his physical examination is normal.

The best match CASEY finds for Uri is a patient named Sarah. She was a 72-year-old woman with a history of angina, complaining of unstable anginal chest pain. Her blood pressure was 138/81, her heart rate was 76, her respiration rate was 14, and her temperature was 98.4. The rest of her physical examination was normal.

The causal explanation for Sarah's findings retrieved from the memory is shown in Figure 1.1.[5] It indicates that her chest pain was caused by a fixed coronary obstruction. She was suffering from both exertional angina (which

[5]In this and all subsequent causal explanation diagrams, items in upper case indicate states in the model of the cardiovascular system. Items in bold face are diagnosis states. Items in lower case are inputs to the program. An arrow from item A to item B indicates that A causes B. A lack of connection between items indicates that they are not causally related.

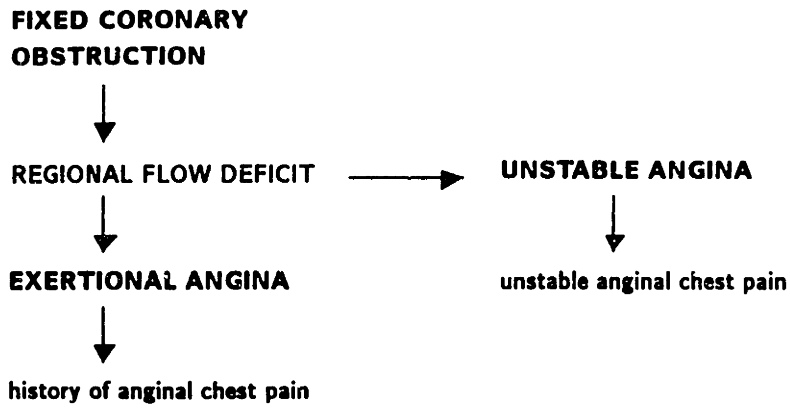


Figure 1.1: **Causal explanation for Sarah.**

explained her history of angina), and unstable angina (which explained her unstable anginal chest pain).

CASEY's next task is to determine whether the solution for Sarah can be adapted to fit Uri. The differences between the patients, shown in Table 1.5, might make the solution unsuitable. One of Sarah's symptoms that was used as evidence in the solution (unstable angina) is absent from Uri's case. Similarly, Uri exhibits symptoms that are absent from Sarah's case and which must be explained. Using information in its causal model and a set of principles for reasoning about causal explanations, CASEY makes the following judgements about the differences between Sarah and Uri:

1. No state in the model of the cardiovascular system uses the sex or age of the patient in any way, so these differences are insignificant.

2. Dyspnea is a significant symptom. CASEY knows this because the model contains the information that when a patient has dyspnea on exertion,

Feature:	Sarah	Uri
Sex	female	male
Age	72	67
Dyspnea	none	on exertion
Chest pain	unstable angina	none
Blood pressure	138/81	135/80
Heart rate	76	87
Chest x-ray	normal	aortic-valve calcification

Table 1.1: **Differences between Sarah and Uri.**

it can be explained by the model 70% of the time.

3. Uri does not have any evidence for unstable angina. This part of the diagnosis does not fit Uri.

4. The difference between the two patient's blood pressures is insignificant.

5. Uri's heart rate is slightly high, while Sarah's is normal. However, a slightly high heart rate does not strongly suggest any disease, so it can be ignored.

6. Aortic valve calcification has only one cause: aortic valve disease. Aortic valve disease must be part of the solution for Uri.

CASEY can repair Sarah's solution to fit Uri by

1. adding dyspnea on exertion as an unexplained feature,

2. removing the diagnosis of unstable angina,

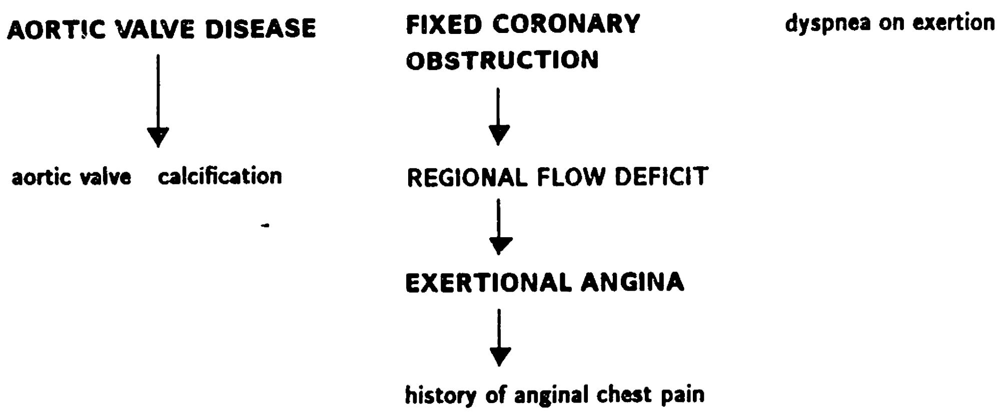


Figure 1.2: **Causal explanation for Uri.**

3. adding the diagnosis aortic valve disease to account for the aortic valve calcification.

The results of these repairs are shown in Figure 1.2 This is identical to the causal explanation for Uri produced by the Heart Failure program. CASEY's explanation, however, is derived without running that program, but by adapting the solution of the past case. This method is significantly more efficient.

Chapter 2

Design and operation

2.1 Overview of the memory system

CASEY remembers cases it has seen by storing them in a self-organizing memory system [17]. A self-organizing memory system records and organizes experiences or *cases.* The memory system also creates *generalizations*, which are structures that hold knowledge describing a group of similar cases.[1] A generalization is created from the *similarities* between the cases that it organizes. Individual cases that are stored in a particular generalization structure are indexed by the features that *distinguish* them from the other cases in the same generalization structure. As a new case is integrated into a generalization, it "collides" with the cases in the generalization that share its differences. This is termed *reminding* [42]. Two cases are said to be *similar* if they are integrated into the same generalization and share a set of differences with the generalization.

The implementation of the memory structure is based on the memory described by Kolodner [17]. Following Kolodner's scheme, the memory structure is represented as a discrimination net in which each node is either an individual case or a generalization structure (called a GEN). Each pointer to a subnode is labeled (indexed) by a feature of the subnode that differentiates it

[1] A note on terminology: Kolodner [17] used the terms Memory Organization Packet (MOPs), features, and norms to describe the structures in a self-organizing memory. The same structures can be thought of as frames, slots, and typical values; or concepts, roles, and prototypes. A MOP is a specialization of a frame that, in addition to holding general (i.e. prototypical) information (that which is true of a typical episode organized by this MOP), also contains a hierarchical structure that indexes all the episodes organized by this MOP. Kolodner later [19] began referring to MOPs as "generalized episodes."

from the parent node. Indexing requires two levels (see Figure 2.1). The first level indicates the *category* of the index (e.g., syncope/near-syncope). The second level indicates the *values* that the feature takes on in the subnodes (e.g., syncope/near-syncope on- exertion; syncope/near-syncope at-rest).

The set of indices defines a set of paths through the memory structure. At each point in the path, one of three conditions obtains. If exactly one case is stored at this point, the stored case and the new case are compared, their similarities placed in a new generalization, and they are indexed beneath the generalization by their differences from each other. Also, the stored case is returned (the program is "reminded" of it). If there is a generalization at the point, the new case is indexed in the existing generalization. If there is no further information that distinguishes the new case from the other cases stored in the GEN, the common features of the GEN are returned.[2] If no other case is stored at this point, the new case is simply installed there, and the common features of the GEN directly above this point are returned.

2.2 Overview of the Heart Failure Program

CASEY is designed around an existing model-based expert system (the Heart Failure program [30]) that diagnoses and suggests therapy for patients with heart failure. The building blocks of the Heart Failure model are *measures*, *measure values*, and *states*. Measures correspond to observable features, such as heart rate, or laboratory results. Measure values are the input values of

[2]In medicine, this would be an instance of a case being a "classic presentation" of some disease.

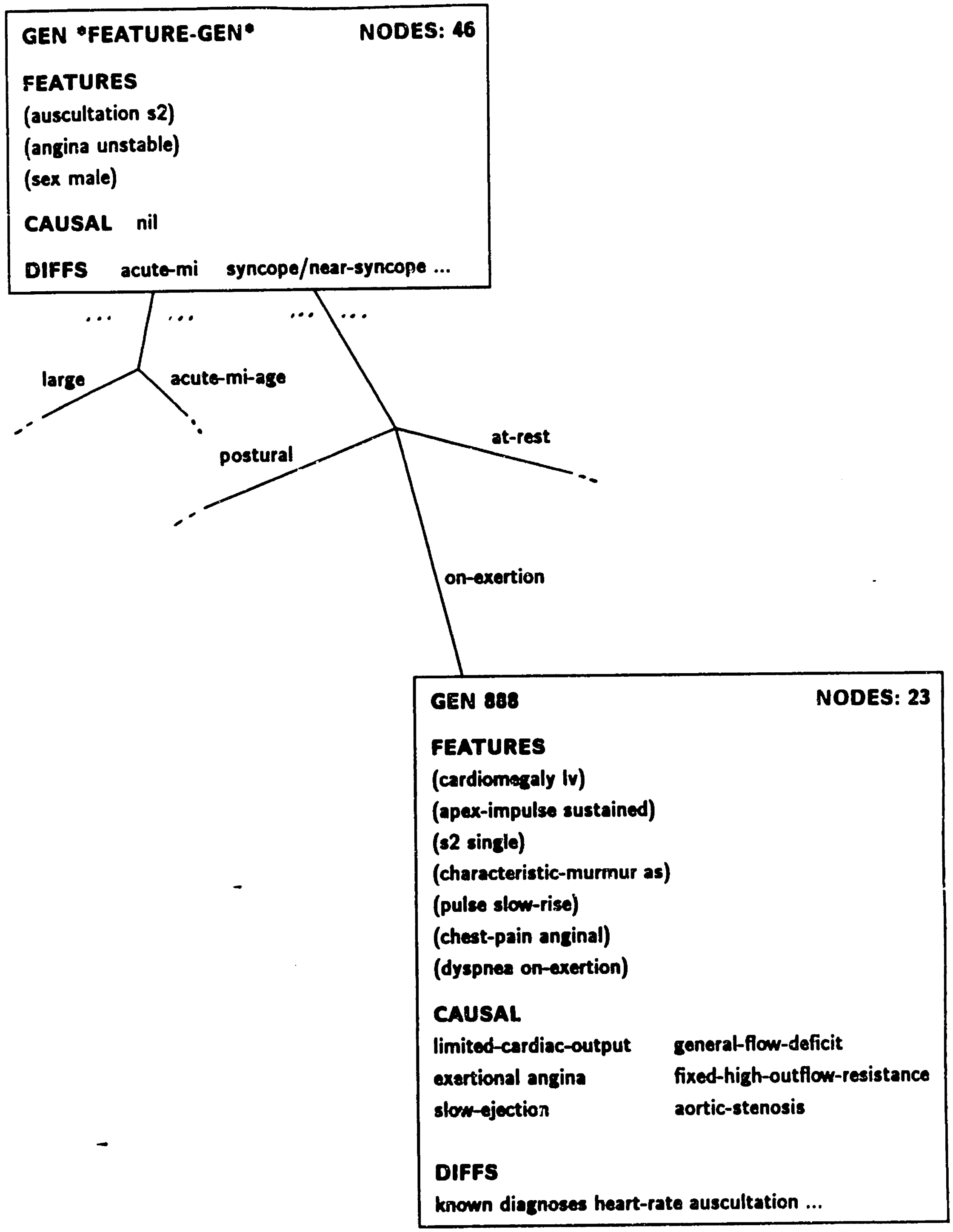


Figure 2.1: A fragment of the memory structure.

```
(defnode  mitral-stenosis
goal      diagnosis
causes    (primary (.003 (if (female sex) .01 .001))
          P+ (mitral-valve-disease :prob .1)
          D- (mitral-valve-replacement))
measure   ((characteristic-murmur (prob ms .5))
          (murmur (prob diastolic-rumble .5))
          (history-findings (prob hemoptysis .1))
          (cxr (prob kerley-b-lines .2))
          (EKG (prob (or first-degree-block wenckebach) .1))
          (s1 (prob loud .75))
          (auscultation (prob lv-s3 (p- 1.0)))
          (auscultation (prob opening-snap .7))
          (valvular-disease (prob MS 1.0))))
```

Figure 2.2: **Information about mitral stenosis.**

the measures, for example, "68" for the patient's heart rate, and are entered by the user. The combination of a measure and a measure value is referred to as a *finding*. States can represent three types of information: specific qualitative assessments of physiological parameters, for example HIGH LEFT ATRIAL PRESSURE; the presence of diseases ("diagnosis" states), for example PERICARDITIS; and therapies given to the patient, for example NITROGLYCERIN. Some states are distinguished as "goal states". These are states that can be treated. The Heart Failure program's information about the state MITRAL STENOSIS is shown in Figure 2.2.[3] The model recognizes two kinds of relationships. It can indicate that one state causes another with a given probability. It can also indicate that a state is associated with a particular finding with

[3]`goal diagnosis` indicates that this is a diagnosis state. `P+` indicates an uncertain cause; `D-` indicates a definite correction. The *measure* slots indicate the probability with which a patient with mitral stenosis will have the given finding.

a given probability. There are over 400 findings and about 140 states defined in the model. The model is represented as a causal inference network. States in the model are shown as nodes. They are connected by links indicating the direction of causality, whether the influence is positive or negative, and probabilities associated with the link.

The Heart Failure program takes as its input a list of findings that describe the patient. A patient description typically consists of about 40 findings. The description for a new patient presented to the system, Larry, is shown in Figure 2.3. From the input, the Heart Failure program produces a solution consisting of a causal explanation, a diagnosis, and therapy suggestions for the patient. The causal explanation describes the relationship between physiological states in the model and observable features of the patient. The diagnosis and the therapy suggestions are derived from states in the causal explanation.

The causal explanation consists of a set of findings, states, and directed links (Figure 2.4). A link between two states, or a state and a finding, indicates that one causes the other. Only abnormal findings are explained, but the program may not explain all abnormal findings. If a diagnosis state is established in the causal explanation, the name of the state is added to the patient's diagnosis. If a goal state is established, the therapy associated with that state is added to the list of therapy suggestions for the patient.

The prototypical concept of a causal model in artificial intelligence is one that contains descriptions of a set of primitive objects and a set of operations that exist in some domain. In order to derive the overall behavior of the system, programs which use this kind of model (e.g., [12], [24], [49], [13], [45], etc.) compute the effects of applying the operations to the objects until some end-

```
(DEFPATIENT  "Larry"
             HISTORY
             (age .  65)
             (sex male)
             (dyspnea on-exertion)
             (orthopnea absent)
             (chest-pain anginal)
             (anginal within-hours unstable)
             (syncope/Near-syncope on-exertion)
             (palpitations none)
             (nausea/Vomiting absent)
             (cough absent)
             (diaphoresis absent)
             (hemoptysis absent)
             (fatigue absent)
             (therapies none)
             VITAL-SIGNS
             (blood-pressure 138 80)
             (heart-rate .  90)
             (arrhythmia-monitoring normal)
             (resp .  20)
             (temp .  98.4)
             PHYSICAL-EXAM
             (appearance nad)
             (mental-status conscious)
             (jugular-pulse normal)
             (pulse slow-rise)
             (apex-impulse normal)
             (parasternal-impulse normal)
             (chest clear-to-auscultation-and-percussion)
             (abdomen normal-exam)
             (extremities normal-exam)
             LABORATORY-FINDINGS
             (ekg lvh normal-sinus)
             (cxr calcification)
             (calcification mitral aortic-valve))
```

Figure 2.3: **Patient description for Larry**

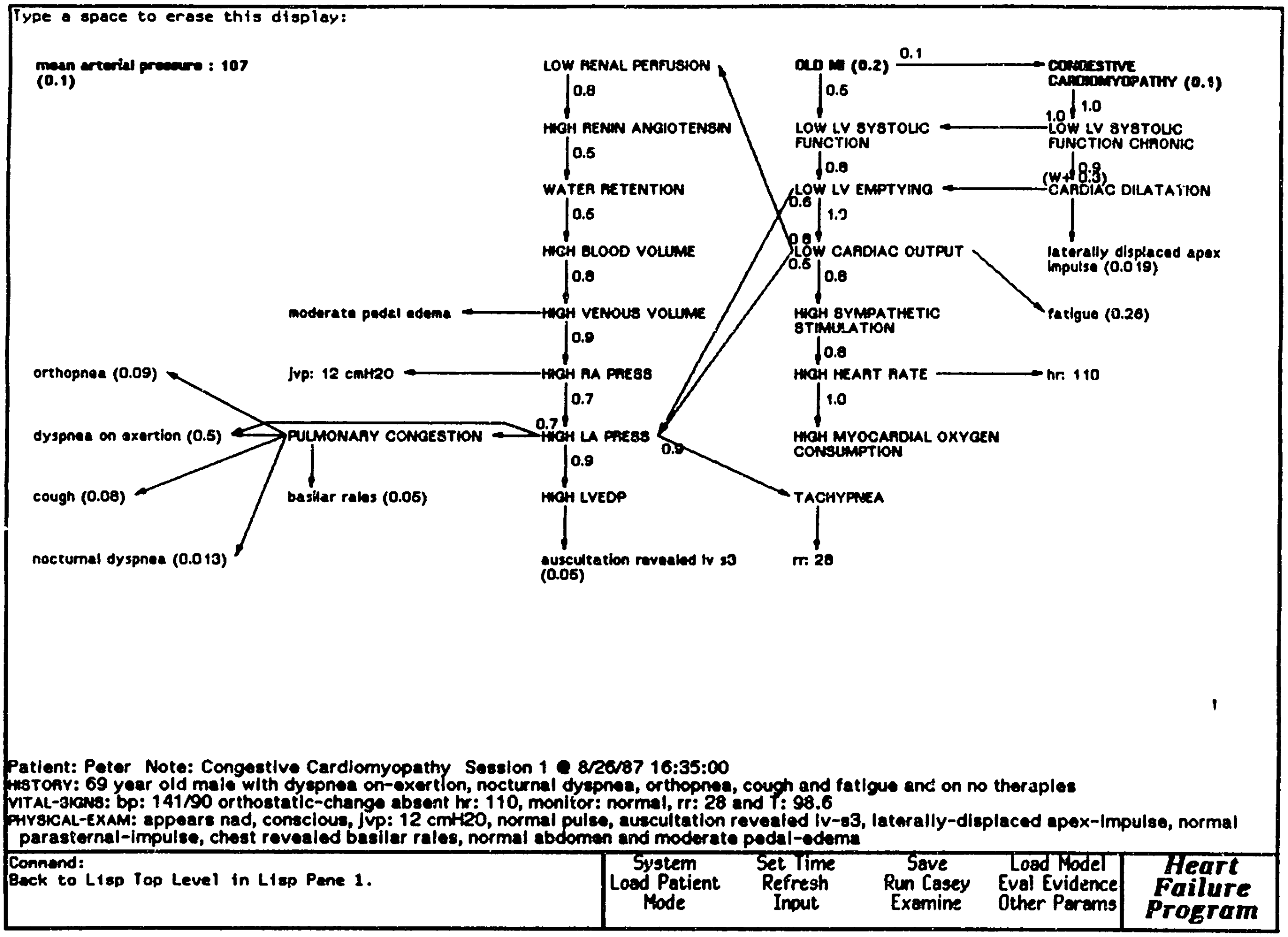


Figure 2.4: A causal explanation produced by the Heart Failure Program.

state or goal is achieved. This computation often takes the form of a simulation or search. The trace from initial state to end-state of the effects of operations on the objects in the system is called a *causal explanation* of the observed end-state. When the effects of applying an operation cannot be determined (as when computing the combined effects of two opposing influences of unknown magnitudes) such systems usually create multiple "possible worlds," one for each uncertain conclusion. For simple systems this method can be useful.

For some other domains, and in particular the cardiovascular domain in which the Heart Failure program operates, the cost of simulation is prohibitive due to the presence of approximately 270 feedback loops in the portion of the domain that the model covers. Furthermore, the cost of maintaining multiple possible worlds is also high in this particular domain. Much of the data needed for simulation can only be obtained invasively,[4] so it is not usually available. This results in an explosion of possible worlds [29]. The Heart Failure program therefore uses a different approach. When the information about states, their causes, and their effects is loaded into the Heart Failure program, the program *precomputes* the trace of the system under various conditions. The diagnostic task, then, is to work backwards from features in the patient description through the trace of potential causes and effects, to find the states which ultimately caused the symptoms. The paths from ultimate (or *primary*) cause to observed features is the causal explanation.

The causal explanation is derived through a complicated process which involves causal, probabilistic, and heuristic reasoning. The Heart Failure pro-

[4]that is, by inserting measurement devices into or otherwise invading the patient's body.

gram propagates evidence backward from the findings to the states that cause them. Some findings have definite causes; those states are established immediately. For each remaining unexplained finding, the system examines every pathway through the model from every diagnosis that could cause the finding. The process of producing an explanation is complicated by the presence of the 270 feedback loops in the model. It is further complicated because the links between findings and the states that cause them are frequently uncertain, so several possible explanations for the patient's findings must be considered simultaneously. The system allows for multiple diseases, and attempts to find a set of diagnoses that "cover" the findings. Each of these covering sets is evaluated and the most probable is selected.

The Heart Failure program was designed to deal with complex clinical situations. Its model has evolved painstakingly over several person-years of effort. Like other model-based programs, it is capable of solving difficult and unusual cases. However, like other model-based programs, its reasoning is extremely expensive computationally. For this reason, the Heart Failure program was an excellent testbed for enhancement through the use of experience.

2.3 Overview of CASEY

CASEY attempts to produce the same causal explanation, diagnosis, and therapy suggestions for a new patient (the case that CASEY is currently trying to solve) as the Heart Failure program. It does so by integrating model-based reasoning, associational reasoning and case-based reasoning, in a five-step process:

- *Retrieval.* CASEY finds a case similar to the new patient in its case memory. This is called the *retrieved case.*

- *Justification.* CASEY evaluates the significance of any differences between the new case and the retrieved case using information in the Heart Failure model. If significant differences are found, the match is invalidated. If all differences between the new case and the retrieved case are judged insignificant or if the solution can be repaired to account for them, the match is said to be *justified.* The *precedent case* is a retrieved case that has been justified and from which solution transfer will occur. The *precedent solution* is the solution associated with the precedent case.

- *Adaptation.* If none of the differences invalidate the match, CASEY adapts a copy of the precedent solution (called the *transferred solution*) to fit the new case. If all matches are ruled out, or if no similar previous case is found, CASEY uses the Heart Failure program to produce a solution for the case *de novo.*

- *Storage.* The new case and its solution are stored in CASEY's memory for use in future problem solving.[5]

- *Feature evaluation.* Those features that were causally important in the solution of this problem are noted in the memory.

The model-based reasoning component of CASEY employs the model of the cardiovascular system developed for the Heart Failure program. Other

[5]The user has the option of rejecting CASEY's solution, in which case Heart Failure program is used to produce a causal explanation, which is then stored in memory.

programs which integrate associational reasoning with causal models (e.g. CHEF and GORDIUS) use their causal model to *simulate* a proposed solution. The complexity of the Heart Failure program's model precludes simulation. CASEY therefore *analyzes* its proposed solution with respect to the causal model, by examining the relationships between evidence in the new case and states in the model.

Associational reasoning is used in CASEY through the association of descriptions of new patients with previously derived solutions for similar patients. This type of association is created with each new patient by the case-based reasoning component (see below). New associational knowledge is also constantly being created through generalizations.

The case-based reasoning component uses a self-organizing memory system [17] to store descriptions of every patient the program has seen, and generalizations derived from similarities between the patients. The patient description is comprised of *features*, such as signs and symptoms, test results, history and current therapy information, and *solution data*, such as the causal explanation for the patient, the diagnosis, therapy recommendation and outcome information.

Retrieving, adapting, and storing cases are standard procedures of a case-based reasoner. CASEY differs from previous case-based reasoning systems because it incorporates reasoning from its causal model in each of these steps.

- Most case-based reasoning systems use a fixed and often *a priori* ranking that indicates which features of a new case are important for matching against cases in the memory (e.g., [5], [13], [46]). It is not always possible to determine in advance which features are going to be important, and furthermore, the important features may vary from case to case. CASEY therefore matches a new case against cases in its memory using every feature in the patient description. Using knowledge of which features were important in determining the causal explanation of previous cases, CASEY then determines the important features of the new case, and gives these features greater weight for matching.

- During justification, model-based reasoning is used to judge the significance of differences between the new and previous cases. Because the match between a new problem and a previously solved problem usually is only partial, there may be differences between the two cases that preclude using even a modified version of a retrieved solution for a new problem. The justification step proves that a retrieved solution can be supported by the features of the new problem.

- Feature evaluation uses the causal explanation of the new case to determine its important features. These are then recorded as part of the case's representation in memory. Determining which features of the new problem were important to the solution helps the program make better matches in the future, because it allows the program to distinguish between extraneous and important features.

CASEY demonstrates that combining a memory of past cases with reasoning from a causal model can have significant advantages over either method used alone.

- CASEY combines the efficiency of associational reasoning with the improved problem-solving ability of model-based reasoning. It can recognize when a case is routine and when it is not. It efficiently solves routine cases by making small local changes to an existing solution. CASEY can recognize that it does not know how to solve a particular problem. When this occurs, it can solve the case by using the Heart Failure program.

- CASEY's performance improves with experience. It learns to solve more problems efficiently as it is given more problems to solve, because it remembers what it has done in the past. It can improve its knowledge by being corrected.

- CASEY can acquire new knowledge automatically by making generalizations about problems that it has solved. It automatically acquires new associational knowledge by making generalizations about each new case presented to it.

- CASEY's model-based reasoning component is enhanced by the ability of the case-based component to learn new associations and compile detailed reasoning structures into simple associations between features and solutions. This results in both improved performance speed and in improved accuracy of the program as new information is added.

- CASEY's case-based component is improved by the use of a causal model because the model can prove that a retrieved solution will be helpful for a new case. Also, the model can be used to identify important features for matching. This results in the elimination of a major limitation of previous case-based reasoning system, the need to fix the important features for matching.

2.4 Matching and Retrieval

2.4.1 Determining the relative importance of features

When presented with a new problem, CASEY searches its memory for a similar case. It compares a new case against cases in its memory using *all* the features in the patient description. However, all features are not equally important in matching a new case to a previous case. Furthermore, the important features for matching may *vary* from case to case. For example, the cardiac rhythm might be important and the heart rate unimportant for one case, whereas for another case, the opposite may be true. Therefore, unlike previous case-based reasoning programs, that use a fixed, and often *a priori*, measure of importance, CASEY's similarity metric allows the important features for matching to be determined for each retrieved case individually. CASEY performs this determination using information in the Heart Failure model. CASEY then compares the important features of the retrieved case with the features of the new case to determine similarity. Thus, although CASEY *retrieves* cases from memory on the basis of all features, it *matches* cases based on features known to be important. For CASEY, *important features* are defined as those that played a role in the causal explanation of previous similar cases.[6]

[6] A similar reluctance to fix a set of important features for matching is seen in HYPO [4], a case-based reasoning program for the domain of law. However, CASEY's causal model allows it to easily identify the important aspects of each precedent case. HYPO has no such model, and therefore must retrieve every precedent that partially matches the new case. It then ranks the precedents according to the number of *dimensions* [4] in common, and examines them in that order.

CASEY's justifier does not require that the new case be identical to a previous case in order to use the latter's solution. In real-world domains, several different pieces of evidence may have equivalent implications. For example, LV strain on EKG and LV enlargement on chest x-ray are both evidence for the same state, LV HYPERTROPHY, even though they represent different features in a patient description. CASEY can repair a causal explanation that includes the state LV HYPERTROPHY to fit a new patient whose description includes evidence of LV HYPERTROPHY, say from an EKG, even if the evidence in the previous case came from a different source, such as a chest x-ray. For matching, therefore, it is sufficient to have features in both cases that are evidence for the same states in the model. CASEY generalizes features in the new case to refer to the states for which they are evidence.[7] These generalized features are called *evidence-states*, because they are states for which there is evidence in the patient. Later, at the time of storage, features of the new case that were used in that patient's causal explanation are generalized to refer to the states which they supported *generalized causal features*.[8] For example, LV HYPERTROPHY ON EKG supporting the state LV HYPERTROPHY becomes EVIDENCE-OF LV HYPERTROPHY.

[7]Thus, CASEY incorporates a form of explanation-based generalization [33], [11], because CASEY generalizes the evidence to the level that retains the same causality. This is discussed further in section 2.7.

[8]The difference between evidence-states and generalized causal features is exactly that evidence-states are states which *might* be in the causal explanation, whereas generalized causal features refer to states that *are* in the patient's causal explanation.

2.4.2 Choosing the Best Match

An input case may have similarities with many previous cases. Most case-based reasoning systems use some sort of similarity metric to determine how similar two cases are, and to choose the "best" match from among the similar cases. A good similarity metric gives a high value for cases that are similar and a low value for cases that are dissimilar. CASEY typically recalls between one and four cases similar to a new case, and places them in a list ordered according to a novel similarity metric. The score for each retrieved case is calculated using the evidence-states of the new case, the generalized causal features of the retrieved case, and the total number of features that the new case and the retrieved case have in common.

CASEY's task is to produce a causal explanation that links evidence and states in the model by finding previous cases that are similar to the new one and would thus have a causal explanation similar to the new case's causal explanation. The relationship between the evidence-states of the new case and the generalized causal features of the retrieved case is thus vital to identifying a good match. The generalized causal features of a past case essentially tell the matcher: "Here are the states for which I need evidence in order to generate this causal explanation." The evidence-states of a new case essentially tells the matcher, "These are the states for which I can provide evidence." A retrieved case that finds evidence for many of its generalized causal features in the new case will be a better precedent than a retrieved case in which few generalized causal features are matched by evidence-states in the new case.

CASEY's similarity metric thus orders matches according to the cardinality of the intersection of the evidence-states and the generalized causal features, minus the number of generalized causal variables that are not matched by evidence-states. The purpose of the latter adjustment is to avoid matching relatively simple cases with large, complicated cases whose explanations cover the simple case but also have many extra states that will have to be removed (Figure 2.5).[9] When two retrieved cases have the same score, the number of features in common is used to break the tie. The reason for this choice is that although the Heart Failure program ignores most normal values, there are many cases in which normal values are important in establishing or ruling out a diagnosis.

Similarity metrics that match cases on the basis of *generalizations* of causally-related features are superior in case-retrieval to those that match cases on the basis of the causally-related features themselves. Typically, many different features can provide evidence to support the existence of the same state, so many different combinations of features can give rise to the same causal explanation. A system that requires the same causally-related features for matching cannot retrieve a case whose causal explanation would be identical except for the particular features used as evidence for the states in the causal explanation.

[9]There is no point in considering the number of evidence-states which go unmatched by generalized causal features. This is because each feature in the patient description generates anywhere from zero to more than 10 evidence-states, so the number of unmatched evidence-states is unrelated to the quality of the match (except when no evidence-states are matched at all; this is detected separately). An interesting possibility would be to calculate how many of the new case's features had been covered by a generalized causal feature which matched some evidence-state in the set generated by that feature. This is left for future work.

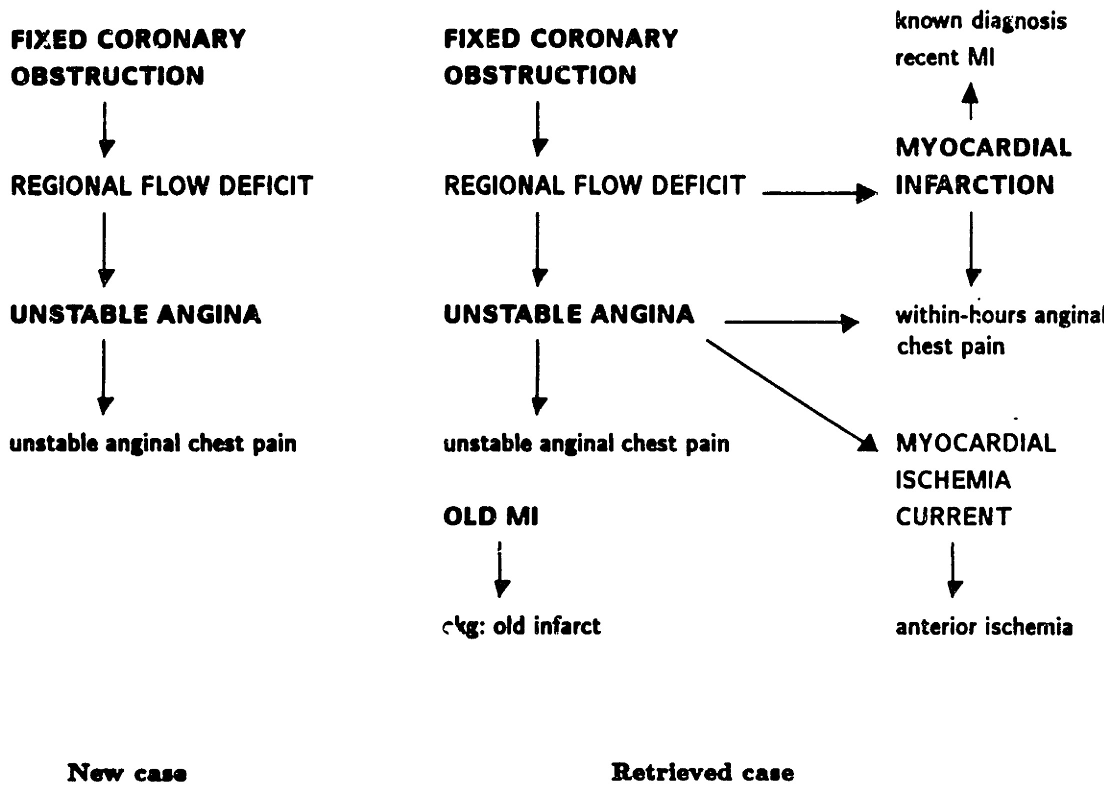


Figure 2.5: A match with many superfluous states.

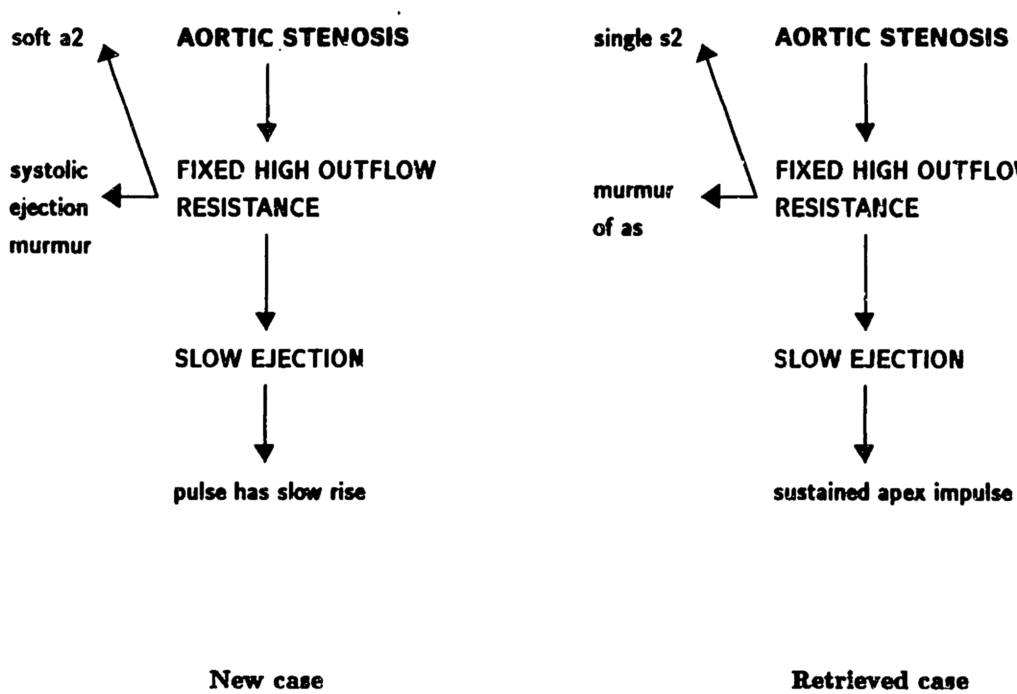


Figure 2.6: A good match with no features in common.

For example, Figure 2.6 shows the causal explanations for two patients who have no causally-related features in common. The two explanations, however, are identical, and therefore these cases represent a good match. The purpose of a matcher is to retrieve cases whose causal explanations will be useful for the new case. Generalizing the causally-related features increases a matcher's chances of finding relevant past cases, and this is the approach used in CASEY.

CASEY first examines the retrieved case with the highest rank. If this match is ruled out (see section 2.5) and there is another retrieved case with a close score (currently, within 10% of the highest score), that case is examined. This continues either until a match is accepted or there are no remaining high-scoring matches.

If the features of a new case are not evidence for any states that have been used to explain the findings of previous patients, CASEY can recognize that it does not know how to solve the case. This is analogous to a physician encountering a patient with a constellation of symptoms the physician has not seen before. Just as the physician would then consult his pathophysiology books, CASEY solves such a problem by invoking the Heart Failure program to find a solution for the new patient.

2.5 Justification

A key question that physicians as well as other problem solvers must answer is whether different constellations of findings still support the same solution. Likewise, CASEY determines whether different features in the patient description can still support the same solution by examining the relationship between evidence and physiological states in the Heart Failure model. The module in CASEY that performs this evaluation is called the *justifier* because it must justify using a retrieved case as a precedent for the new case. The justifier relies on a set of domain-independent heuristics for reasoning about evidence, termed *evidence principles.* The evidence principles reason about such concepts as alternate lines of evidence for states, additional supporting evidence for states, and inconsistent evidence. The first evidence principle is used to determine whether a state in the retrieved causal explanation is ruled out by evidence in the new case, the next four determine whether the difference in question is insignificant or repairable, and the last three handle features that have special values.

1. *Rule out.* A state must be eliminated from the transferred solution if there is some feature in the new case that is incompatible with that state. Incompatibility is defined as zero probability of a feature coexisting with some state in the retrieved solution. For example, a heart rate of 40 beats per minute is incompatible with the state HIGH HEART RATE. Ruling out a state does not necessarily mean that the match is ruled out (see below).

2. *Other evidence* is used when a feature present in the retrieved case is missing in the new case. This principle tries to determine if there is another feature of the new case that supports the same state that the missing feature supported.

   For example, if the feature `opening-snap` supported the state MITRAL-STENOSIS in the retrieved case, but was absent in the new case, CASEY would consult the causal model to find other findings that could be evidence for MITRAL STENOSIS, such as `loud S1` or `diastolic rumble`. CASEY would then search for these other findings among the features in the description of the new case (see Figure 2.7).

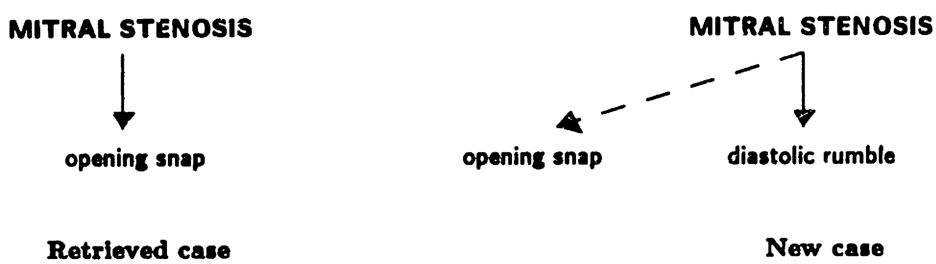


Figure 2.7: Using the evidence principle *other evidence.*

3. *Unrelated oldcase feature* is used when a feature is present only in the retrieved case. If the feature was not used in the causal explanation, its absence has no effect on any states in the explanation, so it can be ignored.

4. *Supports existing state* is used when a feature is present in the new case but not in the retrieved case. This principle determines whether it is possible to attribute the feature to some state in the retrieved causal explanation.

   For example, if the feature `ejection-click`, which is evidence for the states PULMONIC STENOSIS and AORTIC STENOSIS, appeared only in the new patient, CASEY would check for the presence of either of these two states in the retrieved causal explanation. If one or more of these states were present, CASEY would attribute the new feature to that state (see Figure 2.8).

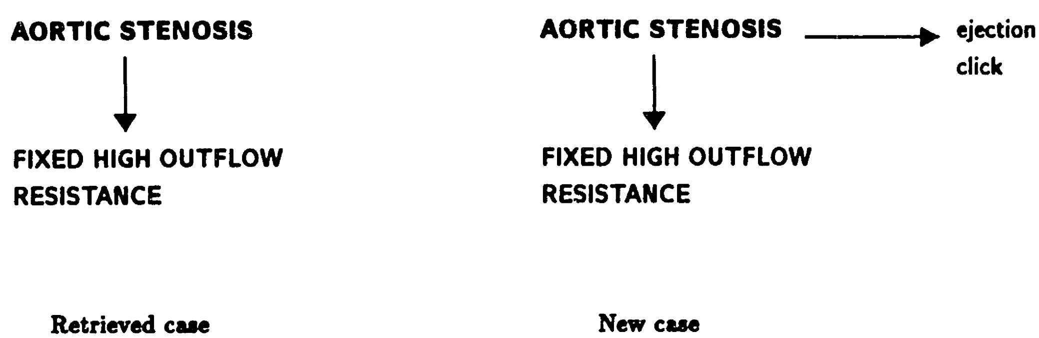


Figure 2.8: Using the evidence principle *supports existing state.*

5. *Unrelated newcase feature* is also used when a feature is present only in the new case. This principle identifies a feature that is abnormal, but does not provide evidence for any existing state and is not strongly suggestive of a new state. Such a feature is added it to the explanation as an "unexplained feature."

For example, the feature `single S2` is abnormal, so it cannot be ignored. It is evidence for the states FIXED-HIGH-OUTFLOW-RESISTANCE and COPD-OR-CHRONIC-BRONCHITIS. But `single S2` alone does not strongly suggest either of these states, so if neither of them are already present in the causal explanation, it is added to the causal explanation as an unexplained feature.[10]

6. *Normal.* Normal values are not explained by the Heart Failure program, so a normal value in the new case is not explained. (Note that if a model did reason using normal values, this rule could be changed).

7. *No information.* If there is no information given about a feature in one of the cases and it is known to have a normal value in the other case, then it is also assumed to have a normal value in the former case.

8. *Same qualitative region.* CASEY evaluates differences between features with numerical values by translating them into physiologically equivalent ranges. For example, a blood pressure of 180/100 becomes "high blood pressure." Features whose values fall into the same range are judged not to be significantly different. Information in the Heart Failure model is used to determine physiologically equivalent ranges.

[10]The information that CASEY uses to determine the advisability of ignoring a particular feature is called the *specificity* of the finding by the Heart Failure program. It indicates the percentage of time the finding is explained by the model. The same number is called the "import" of a finding in the Internist-1/QMR [32] system. CASEY can also determine specificity experientially by examining the role the feature played in similar past cases.

The use of the evidence principles is not guaranteed to result in the same solution as the Heart Failure program. This is because they do not reason about the relative likelihoods of findings. This is discussed in more detail in section 5.2. However, any solution they do produce is *guaranteed* to be a valid possible explanation for the patient's symptom complex.

The changes that CASEY proposes to the retrieved solution are small and local to the difference being considered, and therefore they are computationally inexpensive. However, CASEY evaluates each change in the context of the entire solution. This prevents it from being oblivious to unwanted interactions that might be created by its changes.

CASEY rejects a match either if a significant difference cannot be explained or if all the diagnosis states in the retrieved solution are ruled out. If all differences between the new case and the retrieved case are insignificant or repairable, then solutions are transferred from the precedent to the current case.

2.6 Adapting the solution

CASEY uses *repair strategies* to adapt a previous solution to a new case. There are three types of repair strategies corresponding to the three parts of the solution: causal explanation, diagnosis, and therapy.

2.6.1 Explanation Repair Strategies

Associated with each type of repairable difference detected by the evidence principles is an explanation repair strategy which modifies the precedent causal explanation to fit the new case. Repair strategies modify the transferred causal explanation by adding or removing nodes and links. CASEY makes seven types of repairs:

1. *Remove state.* This strategy can be invoked in two circumstances: either the state is known to be false, or all of the evidence that previously supported the state has been removed (the removed evidence could be either features missing in the new patient, or states ruled out during justification). In the first case, this strategy is invoked by the *rule out* evidence principle. In the second case, when all the evidence for a state is missing in the new case, or if the only cause of a state has been removed from the transferred causal explanation, CASEY removes that state from the explanation. CASEY also determines whether states caused by this state must now be removed.

2. *Remove evidence.* This repair strategy is invoked by the principles *other evidence* and *unrelated oldcase feature.* When a piece of evidence that

was used in the retrieved case is absent in the new case, this removes the feature and any links to it.

3. *Add evidence.* This repair strategy is invoked by the principles *other evidence* and *supports existing state.* It adds a piece of evidence to the causal explanation, and links it to those states for which it is evidence.

4. *Substitute evidence* is invoked by the *same qualitative value* principle. When two numerical values have the same qualitative value, this repair strategy replaces the old value with the new value as evidence for some state.

5. *Add state.* The only time CASEY adds a state to the causal explanation is when the feature it is attempting to explain has only one cause. This repair strategy is invoked by the principle *supports existing state*, because the fact that a feature has only one cause is discovered while CASEY is searching for existing states that cause this feature. When the evidence has only one possible cause, that state is added to the causal explanation. CASEY then tries to link it to existing states and features in the causal explanation (using *add link*).

6. *Add link* is invoked by the *add state* repair strategy, and is used to add a causal link between two states.

7. *Add measure* is invoked by *unrelated newcase feature.* This adds an abnormal feature which CASEY cannot link to the causal explanation.

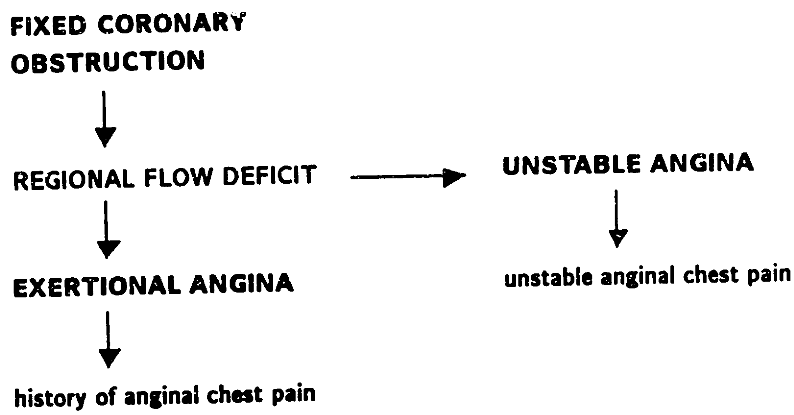


Figure 2.9: **Causal explanation with associated diagnosis of fixed coronary obstruction, unstable angina, and exertional angina.**

After explanation repair has been completed, CASEY can perform diagnosis and therapy repair.

2.6.2 Diagnosis and therapy repair

Because the diagnosis and therapy suggestions are deduced from a patient's causal explanation, diagnosis and therapy repair take place after causal explanation repair. The diagnosis for a patient is simply a list of the diagnosis states in the patient's causal explanation. For example, the causal explanation in Figure 2.9 indicates a diagnosis of fixed coronary obstruction, unstable angina, and exertional angina. Diagnosis repair strategies add and remove diseases from the transferred diagnosis. If a diagnosis state was removed from the transferred causal explanation during explanation repair, the corresponding diagnosis is removed from the patient's diagnosis list. If a diagnosis state was added to the causal explanation, that diagnosis is added to the patient's diagnosis list.

Therapy suggestions are derived from the goal states of the patient's causal explanation. They are indicated in the Heart Failure model as states whose presence *decreases* the effects of state they directly affect. For example, the causal explanation in figure 2.9 produces only one therapy suggestion, coronary artery bypass graft, which is associated with the state fixed coronary obstruction. The therapy repair strategies add a therapy suggestion if a treatable state is added to the causal explanation. They remove a therapy suggestion if the state that was asscociated with that therapy suggestion is removed from the causal explanation.

2.7 Storage and feature evaluation

CASEY stores each case it solves and its solution in its case memory for use in future problem-solving. New cases are stored in the memory indexed both by the input features that describe the case and the solutions (the causal explanation, diagnosis, and therapy suggestions) that were derived for the case. This is true whether the solution was produced by CASEY or by the Heart Failure program.

There are three structures in CASEY that are used to store generalizations: the FEATURE-GEN, the CAUSAL-GEN, and the THERAPY-GEN. In the FEATURE-GEN, cases are retrieved and stored by the features that describe them. Cases are stored in the FEATURE-GEN at the time a new case is presented to the system. In the CAUSAL-GEN, cases are retrieved using their evidence-states. They are stored in the CAUSAL-GEN using their generalized causal features after the causal explanation for the case has been determined. In the THERAPY-GEN, cases are retrieved and stored according to the therapy recommended for the patient.

An individual case is indexed in memory by the all features that describe it. In previous work on case-based reasoning, major effort was expended on selecting those features of the case which were to be used as indices for storing and retrieving the case. CASEY indexes a case by *every* feature that describes it. This approach has two advantages:

1. One can not always determine the usefulness of a feature in advance. My scheme allows useful features to be determined by experience. For each case it solves, CASEY increases the importance weight of the features

that were important in reaching the solution. Because random features should occur only rarely, less useful features fall into the background.

2. The indexing mechanism is very simple, because it always indexes a case by every feature, and does not have to decide which ones are significant or predictive.

CASEY makes generalizations about the cases it has solved by finding similarities between the new case and cases already in its memory [17]. This is known as *similarity-based generalization* [27]. Generalizing the patient descriptions allows CASEY to make predictions about patients who share features [19] by recognizing co-occurrences. In the FEATURE-GEN, CASEY generalizes *all* the features in the patient description, not just the causally-related features. Some features that describe a patient are not used for analysis by the Heart Failure model, and therefore will never be considered *important*. Some of these features may be related to (and therefore can predict) states in the model. For example, no state in the Heart Failure model uses the information on how a murmur changes with valsalva as evidence, although there is a known causal relation for why a systolic murmur associated with the disease IHSS increases upon valsalva maneuver. Normal values for findings are another example. The Heart Failure program ignores most normal findings, even though they can be used to rule out many states. By using similarity-based generalization to learn new associations between features and solutions, CASEY can augment the knowledge in the Heart Failure system. At the same time, making generalizations about groups of similar patients reduces the effect of noise (random, unimportant features in the patient description) on the performance of the program. This is because spurious features are likely to occur randomly, whereas important features will tend to recur with some regularity in cases presented to the program [28].

CASEY also generalizes the new case by creating a description of it using only its observable states (i.e. ignoring the specific evidence for those states, and ignoring internal states with no direct evidence). These are the generalized causal features introduced in section 2.4. The new case is indexed in the CAUSAL-GEN by its generalized causal features. This is an improvement over simply using the input features as indices for storage because it puts the emphasis on the states in the patient's causal explanation rather than on the specific evidence for those states. Since CASEY will accept any evidence for a state as a substitute for any other piece of evidence for that state, it makes sense to allow it to remember and match cases on the basis of classes of evidence.

Separating the generalized causal features and giving them priority in matching has the effect of determining the importance of features by experience. This is reasonable because the usefulness of a feature cannot always be determined in advance. This also allows the problem solver to adapt to changes in the types of problems it is presented over time. Giving extra weight to causally-related features is reasonable because causality often indicates which features are important in the case for matching [51], [43].

Re-evaluating the importance of features is of value if the types of problems presented to the program can change over time. For example, a program designed like CASEY for the domain of general medicine might reasonably be expected to form a generalization that represents a new cluster of simultaneous occurrence of lymphadenopathy, fever, malaise, and immunosuppression in young men (i.e. AIDS), based on its exerience, if it were presented with several such cases.

Chapter 3

Implementation

Figure 3.1 shows a block diagram of the program. The memory structure contains three organizing structures for the cases, the FEATURE-GEN, the CAUSAL-GEN, and the THERAPY-GEN, described in section 2.1. The memory organizer selects the indices from the input cases, organizes the indices to reflect their relative frequencies and importance, integrates cases into the memory structures, creates new generalized episodes, and modifies and refines the knowledge stored in the memory structure. The justifier produces justifications using information from the Heart Failure model, as described in section 2.5, and the adapter modifies past solutions, as described in section 2.6.

3.1 Interface with the Heart Failure program

CASEY is invoked via and takes its input from the Heart Failure program's input screen. The input is translated into CASEY's internal representation in order to search the case memory. If the search is not successful and the Heart Failure program must be run for the patient, the data is still available in the Heart Failure program's representation from the input screen. CASEY uses the Heart Failure program's representation for causal explanations, and can display its results using the Heart Failure program's graph-drawing utilities. CASEY also has routines that let it examine the Heart Failure model.

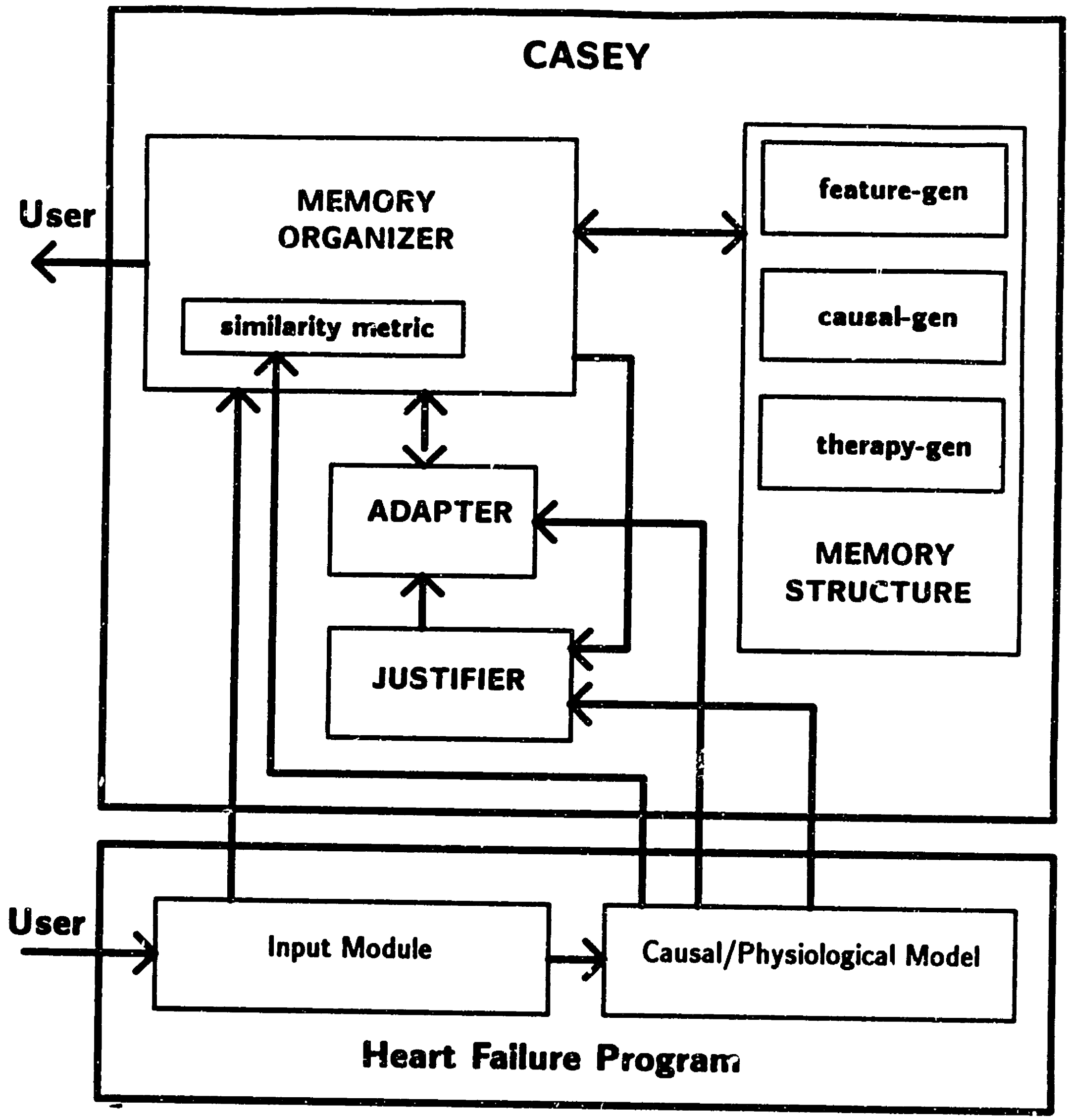


Figure 3.1: Module diagram of CASEY

3.2 Implementation of the memory nodes

3.2.1 Generalizations

GENS are data structures used to hold generalizations created by the program. Each GEN holds information about the two or more cases indexed by this GEN. The *features list* stores the features present in the description of at least 2/3 of the cases in this GEN.[1] Each element of the features list is a 3-tuple consisting of the name of the feature, the value of the feature, and the number of cases indexed in this GEN that share this feature. The *diffs list* holds the indices that are used to differentiate among the cases indexed in this GEN. The *causal list* holds a list of states that are common to the causal explanation of *all* the patients indexed in this GEN. Finally, the *node-count* records how many cases are indexed in this GEN. An example of a GEN created by CASEY is shown in Figure 3.2. This GEN organizes cases that included the feature syncope/near-syncope on exertion. CASEY saw 23 patients with this feature. The patients had other features in common, including anginal chest pain, dyspnea on exertion, and sustained apex impulse (normal values are not shown in the figure). Patients organized by this GEN all shared the causal explanation fragment listed under the heading "causal" in the figure. This generalization represents a substantial number of the cases solved by CASEY.

[1]The fraction of cases that a feature in the features list represents is determined by the system designer.

GEN 888 **NODES: 23**

FEATURES

(cardiomegaly lv)
(apex-impulse sustained)
(s2 single)
(characteristic-murmur as)
(pulse slow-rise)
(chest-pain anginal)
(dyspnea on-exertion)

CAUSAL

limited-cardiac-output	general-flow-deficit
exertional angina	fixed-high-outflow-resistance
slow-ejection	aortic-stenosis

DIFFS

known diagnoses heart-rate auscultation ...

Figure 3.2: A typical generalization structure (GEN)

3.2.2 Cases

The CASE data structure holds information about the individual cases presented to the program. Each CASE holds the following information:

1. The name of the patient.
2. A unique number that identifies this case (the *node-id*).
3. The date and time the case was entered.
4. The description of the patient. This is represented as a list of feature/value pairs.
5. The causal explanation derived for this patient. This is represented as a list of nodes and links.
6. The generalized causal features for the case. (Before a causal explanation has been derived for the case, this slot holds the evidence-states for the patient.
7. The patient's diagnosis.
8. Any therapy suggestions made for this patient.
9. The source of the solution for this patient (either the Heart Failure program or another case). If this patient's causal explanation was transferred from another case, the precedent case and any substitutions made in adapting the precedent solution to the current case are recorded.
10. Any follow-up information available for this patient.

PT-NAME:	Natalie
NODE-ID:	node-75
SESSION-AT:	10/30/87 8:55:00
DATA:	age 62, sex female, dyspnea on-exertion, orthopnea absent, chest-pain anginal, anginal unstable, syncope/near-syncope none, palpitations none, cough absent, diaphoresis absent, hemoptysis absent, nausea/vomiting absent, fatigue absent, therapies none, heart-rate 86, resp 14, temp 98.3, mean-arterial-pressure 103, appearance anxious, mental-status conscious, jugular-pulse normal, pulse slow-rise, parasternal-impulse normal, auscultation murmur, murmur systolic-ejection-murmur, auscultation s2, s2 soft-a2, apex-impulse laterally-displaced, cxr cardiomegaly, cardiomegaly lv, cxr calcification, mitral calcification, aortic-valve calcification, ekg sinus-rhythm, ekg lv-strain, arrhythmia-monitoring sinus-rhythm extremities normal-exam, abdomen normal-exam, chest clear-to-auscultation-and-percussion
CAUSAL-EXPLANATION:	limited-cardiac-output, slow-ejection, unstable-angina general-flow-deficit, lv-systolic-function, cardiac-dilatation, lv-systolic-function-chronic, lv-hypertrophy, lv-press-chronic fixed-high-outflow-resistance, aortic-stenosis, anxiety, aortic-valve-disease, mitral-valve-disease, (dyspnea on-exertion), (unstable anginal chest pain), (appearance anxious), (arterial-pressure 103), (heart-rate 86), (pulse slow rise), (s2 soft-a2), (ekg lv-strain), (cardiomegaly lv), (calcification mitral), (calcification aortic-valve), (apex-impulse laterally-displaced),

Figure 3.3: **An example of the CASE data structure.**

GEN-CAUSALS:	(present aortic-valve-disease), (present mitral-valve-disease), (present cardiac-dilatation), (present lv-hypertrophy), (present fixed-high-outflow-resistance), (present slow-ejection), (present anxiety), (present unstable-angina), (present limited-cardiac-output)
DIAGNOSIS:	unstable-angina, aortic-stenosis, aortic-valve-disease, mitral-valve-disease
THERAPY:	(aortic-valve-replacement aortic-valve-disease)
TRANSFERRED-FROM:	node-64 (same-qualitative-region (mean-arterial-pressure:103 mean-arterial-pressure:104) rule: (high blood-pressure)) (definite-cause (calcification mitral) mitral-valve-disease) (new-state mitral-valve-disease) (definite-cause (calcification aortic-valve) aortic-valve-disease) (new-state aortic-valve-disease) (causes aortic-valve-disease aortic-stenosis) (other-evidence (pulse slow-rise) slow-ejection) (other-evidence (s2 soft-a2) fixed-high-outflow-resistance) (other-evidence (apex-impulse laterally-displaced) cardiac-dilatation) (other-evidence (cardiomegaly lv) cardiac-dilatation) (supports-existing-state (cardiomegaly lv) lv-hypertrophy) (no-evidence high-sympathetic-stimulation)
OUTCOME:	nil

Figure 3.4: **More of the CASE data structure.**

An example of the CASE data structure is shown in Figures 3.3 and 3.4.

The patient name, date, and time together serve to uniquely identify the session with the patient. The follow-up information[2] is useful for making predictions about patients similar to this one. For example, if the patient did not respond to the therapy recommended by the program, this information could be used by the physician when considering the therapy for a future patient.

3.3 Complexity of the memory scheme

As more cases are added to the memory, concerns might be raised about the size of the case memory and the increase in retrieval time. The memory retrieval scheme used by CASEY never examines all the nodes in the memory. It only follows those paths specified by features in the new patient's description. The time to follow a path in memory is proportional to the depth of the search tree, which in turn is dependent on (at most) the number of features in the longest patient description. It is independent of the number of cases stored in the memory.

A small experiment was performed to determine how retrieval time changed as the number of cases in the memory increased. Four GENs were built using increasing numbers of cases. Then the time to retrieve matches for two cases was measured, one from the FEATURE-GEN and one from the CAUSAL-GEN. The results, given in Table 3.1, indicate that the time (in seconds) to

[2]none is shown in the example.

	Number of cases in memory			
	10	20	30	44
feature-gen	7.06	7.00	7.40	7.58
causal-gen	.02	.11	.14	.09

Table 3.1: **Results of a timing experiment.**

retrieve a case from memory was indeed almost constant as the number of cases increased.

The growth in size of the case memory is highly dependent on the nature of the cases stored in it. If the program were presented with one thousand identical cases, the case memory would consist of exactly one GEN whose features would be the features in their description, with no diffs and thus no individual cases stored in the memory. If the memory were presented with one thousand cases that had no features in common, it would consist of a single GEN whose features list was empty, and a diffs list that held every feature in each of the cases' description. At the end of each diff would be a single case. In ordinary use, the nature of the cases presented to the system will most likely fall between these two extremes.

When the memory is in its early stages of use, there are many features that it has never encountered before. A new feature is entered as a diff in the top-level GEN and increases the breadth of the memory structure. Subsequent cases that have this feature in their description will create generalizations below the diff, and thus increase the depth of the memory structure.

When CASEY is presented with a case whose features are identical to the features of a GEN, there is no way to distinguish the case from the GEN and thus there is no need to remember the case, since it is already completely described by the generalization. As more and more cases are seen by the program, GENs representing common types of problems seen by the program are formed and refined. The cases from which those GENs were derived are no longer accessible. When a case is no longer distinguishable by any feature from a generalization, it is discarded. The only cases that are explicitly stored are exceptions to these "prototypical" problem types. (Cases that are not stored because they are identical to a generalization are still used to increment the importance weights in the memory. This ensures that CASEY keeps its importance weights current.) Creating generalizations reduces the depth of the tree because a case is indexed into the memory structure beneath a GEN only by those features that are different from the features in the generalization. After the memory has seen a variety of cases, therefore, it grows more slowly, and may even compact itself.

3.4 Constructing a similarity metric

Two similarity metrics were implemented for CASEY, although only one is used. One used a combination of usage counts and causal importance of features, the other, presented in section 2.4.2, used generalized causal features.

To implement the first metric, each index is given two slots for maintaining usage information. The first slot, *usecount*, is incremented each time the feature is seen in a case. The second, *priority*, is incremented every time a feature is found to be used in a causal explanation. The ratio *priority/usecount* basically determines the importance ordering of the indices, although it is adjusted for frequency (see below). If the ratio is low compared to that of other indices, it indicates that a feature is common without being causally important so its usefulness is low. This scheme allows rare but causally important features to be considered more useful than simple frequency would indicate (since they are rare, their usecount is low, so their ratio is high). The weighting scheme is somewhat more sophisticated than a simple ratio. For example, a feature whose ratio = 1 but whose usecount = 100 is considered more important than a feature whose ratio and usecount are both 1. Similarly, the system ranks a feature whose ratio = 0 and whose usecount = 100 lower than one whose ratio = 0 but whose usecount is only 3. The result of this similarity metric is that important features are recognized, while spurious features are downplayed.

Although I had originally intended to use the above metric, an analysis of the best match for several cases determined that generalized causal features were most important in determining the best match for a case (the metric is described in section 2.4.2). This is because a decision was made to always attempt to reproduce the Heart Failure program's solution (rather than allowing CASEY to potentially find a better solution). Generalized causal features group features that are evidence for the same state, so using generalized casual features groups cases that have evidence for the same states. Therefore the second metric is the one that is used in the current implementation.

The first metric would be more useful than the second if the Heart Failure program were allowed to be overridden. In that case, CASEY would be calculating new "evoking strengths" of features for diagnoses, whereas currently CASEY keeps the Heart Failure program's probability weights.

3.5 Implementation of the justifier

Before the system can attempt to justify a match between two cases, it must first identify the differences between them, and then decide which ones are significant. Identifying any differences is simple, because the memory organizer indexes the two cases by their differences in the GEN that is created when the similar case is found. CASEY can identify the following types of differences: features missing in the new case, extra features in the new case, and features that have different values. The justifier then evaluates each difference using the evidence principles. Whenever a difference is judged insignificant or a repair can be made to account for the difference, the justification for the change is recorded in a list of justifications. Some features of the new case may remain unexplained after this step. The justifier examines all the causes in the Heart Failure model for each unexplained feature. If the feature has a definite cause, or only one cause, that cause is added to the causal explanation. Otherwise, the feature remains unexplained. Next, the justifier tries to find support for states in the causal explanation that have no support, either because all the evidence for the state is missing in the new case, or because the cause of that state was removed. Again, if the state has a definite cause, or only one cause, that cause is added to the causal explanation. The justifier also examines the causal explanation for any evidence that can be used to support the state. If no support for the state is found, it is removed from the causal explanation. Finally, the justifier checks for the two failure states. If all diagnosis states have been removed from the causal explanation, or if some feature in the new case remains unexplained, the match fails. Otherwise, the match is accepted and the list of justifications is returned.

3.6 Implementation of the repair strategies

The implementation of the explanation repair strategies is quite simple: each evidence principle generates a list containing the objects in the old causal explanation that must be changed or added, and a tag indicating the change that must be made to the explanation. For example, `(add (anginal experiencing) unstable-angina)` or `(substitute mean-arterial-pressure 102 103)`. The repair strategies are called according to the change that must be made, and the causal explanation is incrementally modified. This process is not very expensive because all changes to the causal explanation are local to the state named in the input string. The repair strategies are independent of the particular implementation of the domain model. In order to use these on a different model, only the lowest-level routines (the ones that actually add states and links to the explanation) need be changed.

Diagnoses and therapy suggestions are both determined by the presence of distinguished states in the causal explanation. Thus, diagnosis and therapy repairs are both linked to the explanation repair strategies *add-state* and *remove-state*. As described in section 2.6.2, when a diagnosis state is added or removed, a diagnosis repair strategy modifies the diagnosis appropriately. When a treatable state is added or removed, a therapy repair strategy adds or removes a therapy suggestion .

Chapter 4

Results

4.1 A detailed example

A new patient, Natalie, is presented to the system. She is a 62-year- old female complaining of dyspnea on exertion and unstable anginal chest pain. She appears anxious. Her blood pressure is 146/81 and her heart rate is 86 beats per minute. Auscultation reveals soft A2 and a systolic ejection murmur. She has a laterally displaced apex impulse. Her EKG shows LV strain, and her chest x-ray shows LV cardiomegaly and mitral and aortic valve calcification. The rest of her examination is normal. The exact input presented to CASEY is shown in Figure 4.1. renewcommand11.5 As a prelude to retrieval, CASEY generalizes all features in Natalie's description to determine the states for which there is evidence in this patient. These are Natalie's evidence-states. For example, the feature "LV cardiomegaly" is evidence for the states LV HYPERTROPHY and CARDIAC DILATATION. According to the model, Natalie has evidence for 66 states. In order to find a previous case similar to Natalie, CASEY searches the CAUSAL-GEN for cases that have the evidence-states in their causal explanation. It also searches for patients similar to Natalie in the FEATURE-GEN. This is the retrieval step. CASEY retrieves two cases that are similar to Natalie, Cal and Margaret. CASEY uses its similarity metric to rank the retieved cases. As shown in Table 4.1, all of Cal's seven generalized causal features are covered by Natalie's evidence-states. One of Margaret's eight generalized causal features, "evidence of high sympathetic stimulation," is not covered by Natalie's evidence-states. In terms of number of generalized causal features in common, these two cases rank equally. The number of total features in common is used to break the tie. Cal and Natalie

```
(defpatient  "Natalie"
             (age . 62)
             (sex female)
             (dyspnea on-exertion)
             (orthopnea absent)
             (chest-pain anginal)
             (anginal unstable)
             (syncope/near-syncope none)
             (palpitations none)
             (nausea/vomiting absent)
             (cough absent)
             (diaphoresis absent)
             (hemoptysis absent)
             (fatigue absent)
             (therapies none)
             (blood-pressure 146 81)
             (heart-rate . 86)
             (arrhythmia-monitoring normal)
             (resp . 14)
             (temp . 98.3)
             (appearance anxious)
             (mental-status conscious)
             (jugular-pulse normal)
             (pulse slow-rise)
             (auscultation s2 murmur)
             (s2 soft-a2)
             (murmur systolic-ejection-murmur)
             (apex-impulse laterally-displaced)
             (parasternal-impulse normal)
             (chest clear-to-auscultation-and-percussion)
             (abdomen normal-exam)
             (extremities normal-exam)
             (ekg lv-strain normal-sinus)
             (cxr calcification cardiomegaly)
             (calcification mitral aortic-valve)
             (cardiomegaly lv))
```

Figure 4.1: **Patient data for Natalie**

	Generalized Causal Features	Evidence States
Cal	Margaret	Natalie
mitral valve disease	—	yes
aortic valve disease	—	yes
unstable angina	unstable angina	yes
slow ejection	slow ejection	yes
limited cardiac output	limited cardiac output	yes
LV hypertrophy	LV hypertrophy	yes
fixed high outflow resist.	fixed high outflow resist.	yes
—	cardiac dilatation	yes
—	anxiety	yes
—	high sympathetic stim.	no
total: 7/7	total: 7/8	

Table 4.1: **Match analysis of Cal and Margaret for Natalie**

have 28 features in common, whereas Margaret and Natalie have 27 features in common. Therefore the match with Cal ranks higher than the match with Margaret, and CASEY first tries to justify the match with Cal.

Cal's causal explanation is shown in Figure 4.2. In the justification phase, CASEY determines that Cal's explanation can be modified to account for all of Natalie's findings except for the laterally displaced apex impulse. This finding is not accounted for by any state in Cal's explanation and it has no easily determined cause (as does Natalie's finding of anxious appearance, which has only one cause, anxiety). The justification thus fails, and CASEY next considers Margaret as a precedent.

Margaret's causal explanation is shown in Figure 4.3. The differences between patients Natalie and Margaret, which CASEY must explain by justifying the match, are shown in Table 4.2.

CASEY makes the following inferences about the differences between pa-

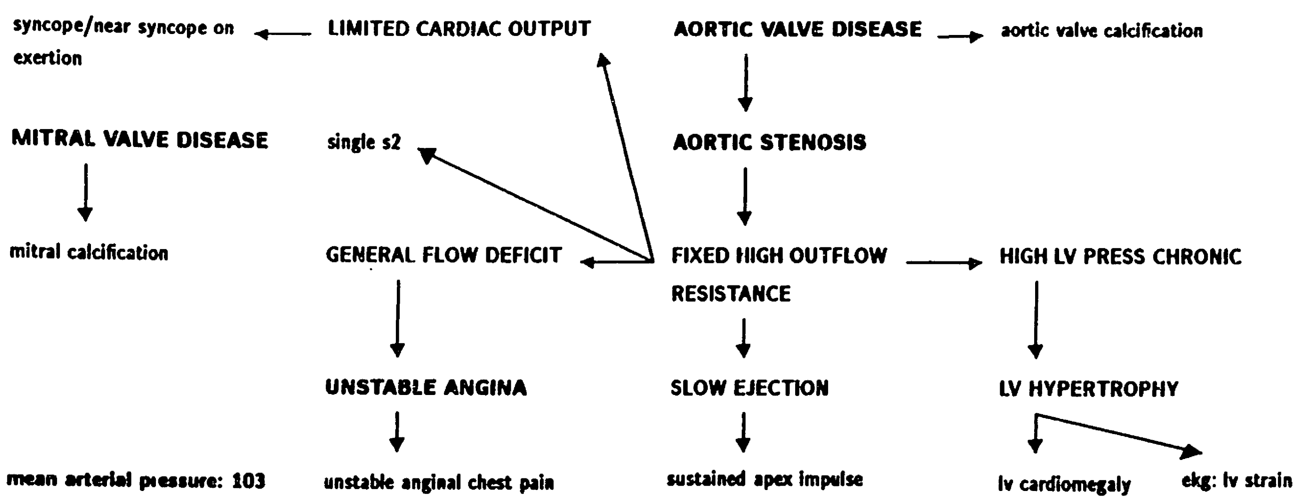


Figure 4.2: **Causal explanation for Cal.**

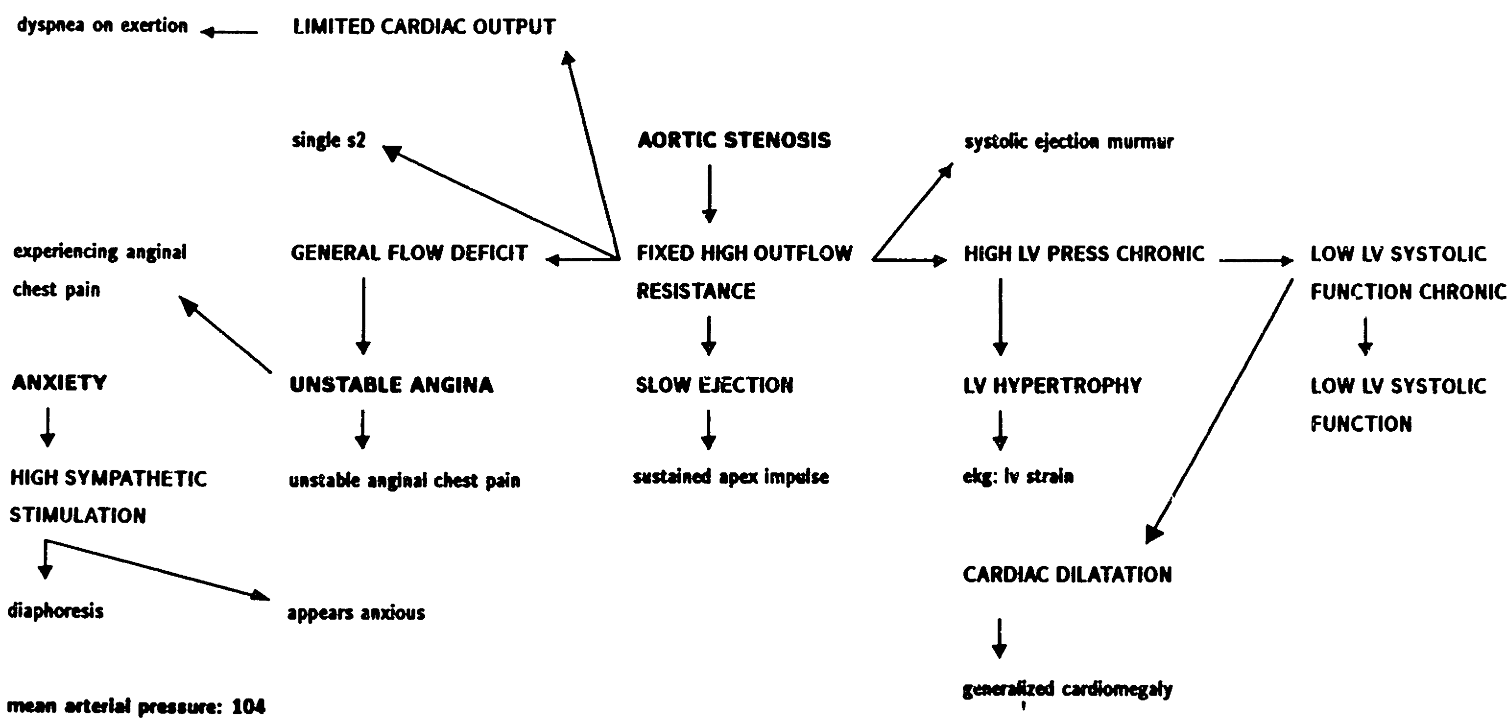


Figure 4.3: Causal explanation for Margaret.

Feature name	Value for Natalie	Value for Margaret
age	62	67
temperature	98.3	98.7
heart rate	86	90
blood pressure	146/81	148/unknown
apex impulse	laterally-displaced	sustained
parasternal impulse	normal	unknown
pulse	slow-rise	normal
s2	soft A2	single
chest x-ray	mitral and aortic calcification	none
	LV cardiomegaly	generalized cardiomegaly
angina	unstable	unstable experiencing
appearance	anxious	anxious diaphoretic

Table 4.2: **Differences between patients Natalie and Margaret.**

tients Natalie and Margaret:

- No rule in the Heart Failure model uses age as evidence, so Margaret's age is judged to be insignificant by the rule *unrelated oldcase feature*, and Natalie's age is judged to be insignificant by the rule *unrelated newcase feature*.

- Both patients' heart-rates are in the same qualitative region (moderately high heart rate) so the difference is considered insignificant.

- Both temperatures are in the "normal" qualitative region so the difference is considered insignificant.

- Both patient's blood pressures are in the "high" qualitative region so the difference is insignificant.

- Margaret's finding of sustained apex impulse supports the state SLOW EJECTION. Natalie does not have a sustained apex impulse, but she does have a slow rise pulse. This is *other evidence* for the state SLOW EJECTION.

- Natalie's finding of a laterally-displaced apex-impulse supports the existing state CARDIAC DILATATION.

- Natalie's parasternal impulse is normal and does not have to be explained.

- Single S2 and soft A2 both support the existing state FIXED HIGH OUTFLOW RESISTANCE.

- LV cardiomegaly in Natalie is evidence for the same states that generalized cardiomegaly supports in Margaret's causal explanation. LV cardiomegaly also supports the existing state LV HYPERTROPHY, so a link must be added between the finding and the state.

- Mitral valve calcification and aortic valve calcification on chest x-ray are both definite evidence for the states MITRAL VALVE DISEASE and AORTIC VALVE DISEASE, so theses states are added to the causal explanaiton.

- Natalie does not have the finding "experiencing unstable angina," but she has other evidence, namely "unstable anginal chest pain," to support the state UNSTABLE ANGINA,

- Natalie's finding of "appears anxious" supports the existing state ANXIETY. There is no longer any evidence for the state HIGH SYMPATHETIC

STIMULATION so it is removed.

All the differences between Margaret and Natalie are insignificant or repairable, so the match is said to be justified.

In order to adapt the explanation transferred from Margaret to fit the data for Natalie, the following repair strategies are invoked by the justifier:

```
(substitute-evidence mean-arterial-pressure 103 104)
(remove-evidence unstable-angina experiencing)
(add-state (mitral-valve-disease))
(add-evidence (calcification mitral) mitral-valve-disease)
(add-state (aortic-valve-disease))
(add-evidence (calcification aortic-valve) aortic-valve-disease)
(add-link aortic-valve-disease  aortic stenosis)
(remove-evidence (apex-impulse sustained) slow-ejection)
(add-evidence (pulse slow-rise) slow-ejection))
(add-evidence (s2 soft-a2) fixed-high-outflow-resistance)
(remove-evidence (s2 single) fixed-high-outflow-resistance)
(add-evidence (apex-impulse laterally-displaced) cardiac-dilatation)
(remove-evidence (cardiomegaly generalized) cardiac-dilatation)
(add-evidence (cardiomegaly lv) lv-hypertrophy)
(add-evidence (cardiomegaly lv) cardiac-dilatation)
(remove-evidence diaphoresis high-sympathetic-stimulation)
(remove-state diaphoresis)
```

The changes that must be made to Margaret's causal explanation to fit the details of Natalie's description are shown in graphically in Figure 4.4. The

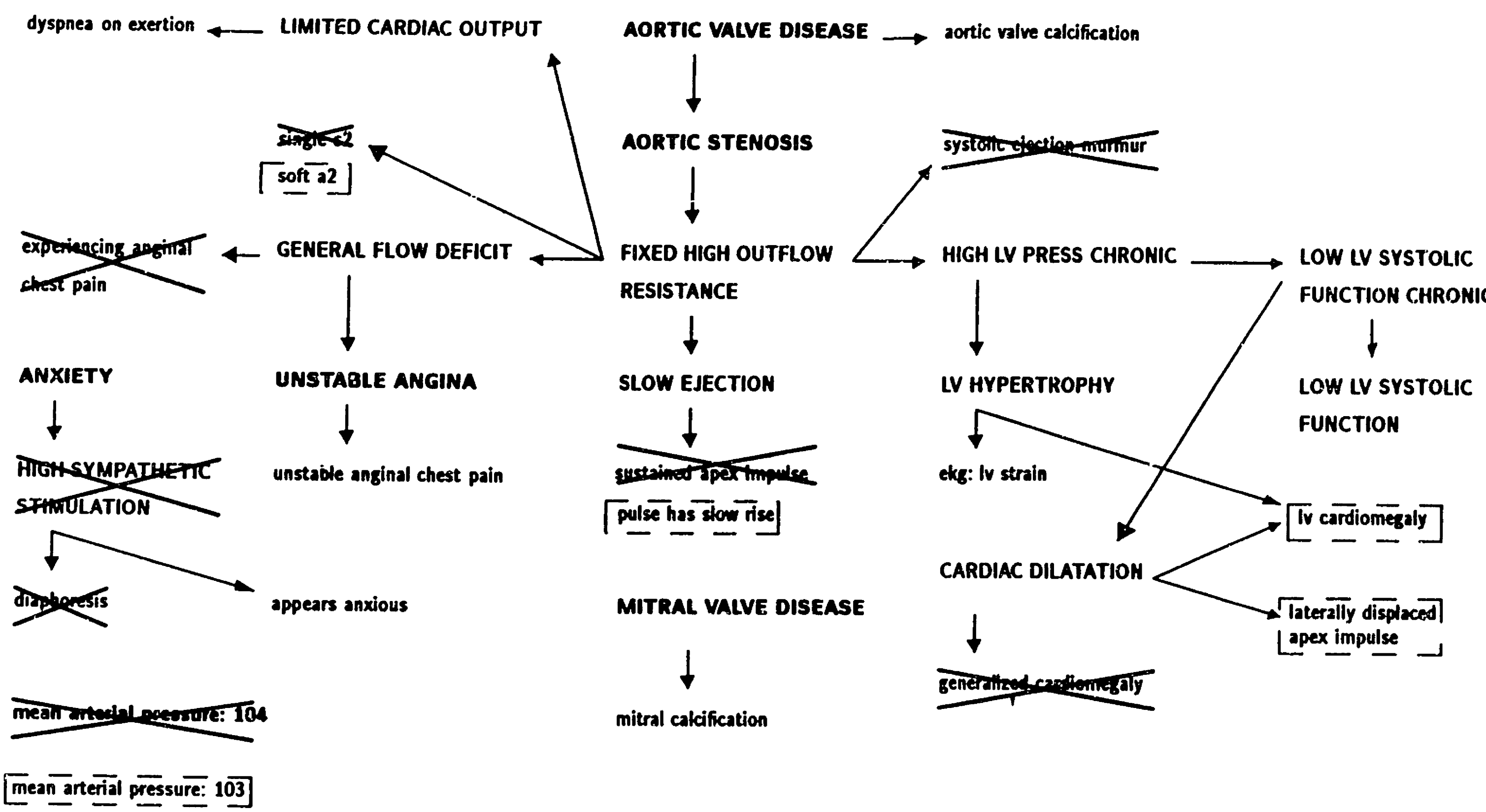


Figure 4.4: **Modifications to Margaret's causal explanation.**

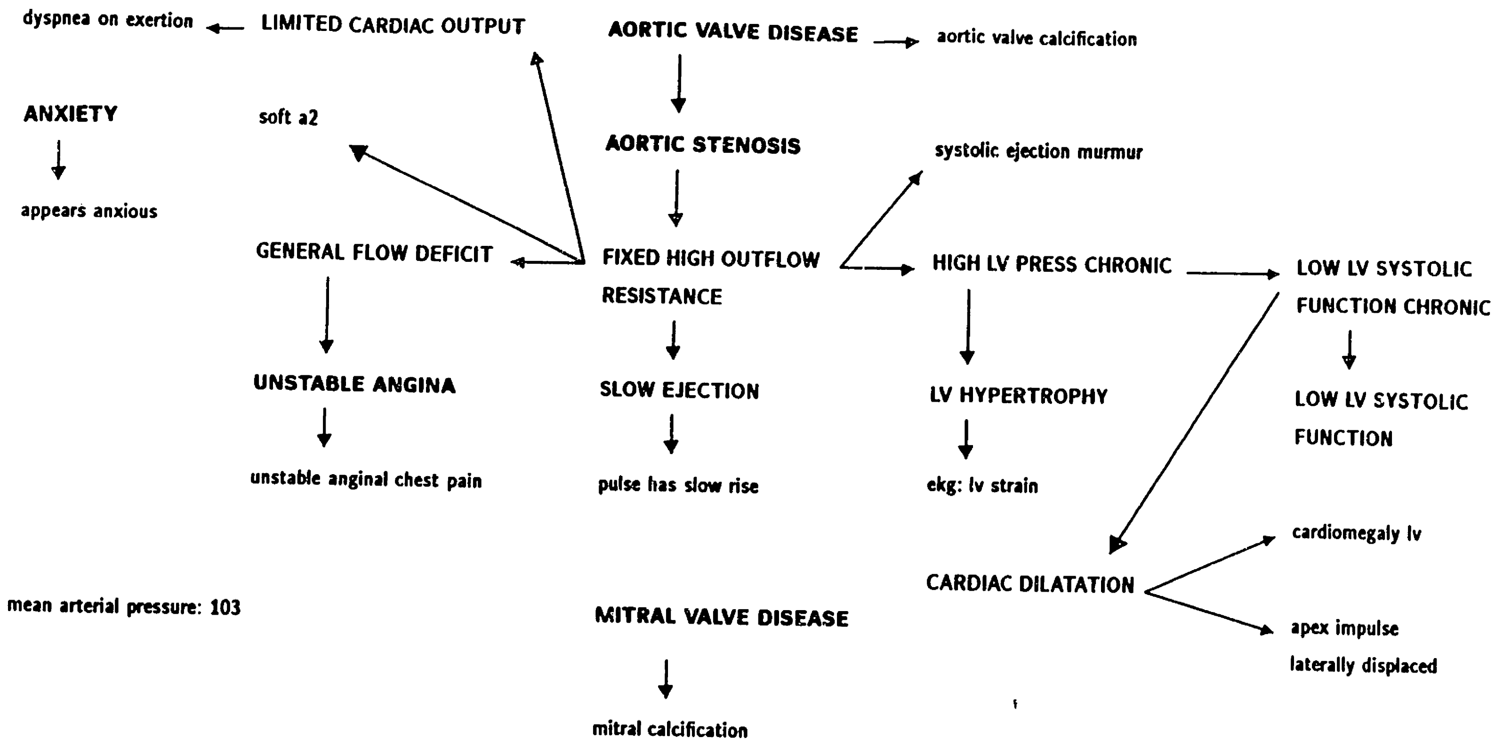


Figure 4.5: **Causal explanation for Natalie.**

causal explanation produced by CASEY for Natalie is shown in Figure 4.5. It is identical to the causal explanation produced *de novo* by the Heart Failure program.

CASEY then identifies the states in the causal explanation of the new case that are directly linked to findings (the generalized causal features), and indexes the new case in memory using these features. In the case of Natalie, these states are limited cardiac output, unstable angina, slow ejection, lv hypertrophy, aortic valve disease, mitral valve disease, cardiac dilitation, fixed high outflow resistance, and anxiety. Future cases that contain evidence to support these states will retrieve Natalie's case as a potential precedent.

In retrospect, Cal's causal explanation is quite close to the one finally derived for Natalie. However, CASEY can be much more certain about removing states for which there is no evidence than about adding states to explain a feature (except for those instances in which the feature has a definite cause). CASEY therefore takes a conservative approach and only adds a state when there is definite evidence for the state, or when the evidence only has one cause. The difficulty in adding states is that CASEY would be required to choose the "best" cause from among several likely causes of the feature. There are two ways in which this could be accomplished. First, CASEY could use heuristic methods to select what it thinks is the most likely cause. The advantage of this approach is that it would be fairly efficient. The problem with the heuristic approach is that CASEY could not be certain that the state it added was indeed the most likely. An alternative approach would be to run the Heart Failure program in a limited manner, only using it incrementally explain the unexplained features. This approach, while requiring more computation that the heuristic method, would have the advantage that the state selected by the Heart Failure program in fact would be the most likely cause of the feature, given the rest of the explanation.

4.2 Analysis of CASEY's performance

CASEY's performance was evaluated on two counts: *efficiency*, and *quality* of the solution. The program was tested on a set of 45 patients with symptoms of heart failure. Twenty of the cases represented patients with coronary artery disease or aortic stenosis. This set of test cases was specifically designed by a physician for testing CASEY. The remaining 25 cases were designed to test the range of the Heart Failure program, and described patients with various causes of heart failure.

Each case was first evaluated by the Heart Failure program, and that solution was saved. The 45 cases were then used to test CASEY in the following manner:

1. The test patient was removed from the test set.

2. The FEATURE-GEN and CAUSAL-GEN were built up using the other patients in the test set.

3. CASEY was invoked on the test patient.

The quality of CASEY's solution was evaluated by comparing its output to the Heart Failure program's output for the same patient. A solution was considered *successful* if it was identical to the Heart Failure program's solution. A solution was considered *satisfactory* if it was identical to the Heart Failure program's solution except for the features which CASEY could not explain. In these latter cases, CASEY had already performed most of the task of deriving the causal explanation, and the Heart Failure program could be used to incrementally account for the remaining features.

CASEY produced a solution identical to the Heart Failure program's solution in 14 out of the 45 test cases. It produced a satisfactory explanation for an additional 18 test cases. It gave up on six of the test cases, and produced an incorrect causal explanation for seven test cases. Of the twenty cases specifically chosen for testing CASEY, eleven were solved identically and nine were solved satisfactorily. None of theses were solved incorrectly or were unsolved. An examination of the test cases for which CASEY failed to reproduce even part of the Heart failure program's solution revealed that each one of these cases had a causal explanation that was completely different from any other patient in the memory. Even on these cases, CASEY could often produce part of the causal explanation, but could not account for the combination of features seen in the patient. CASEY failed to produce a solution in precisely those cases for which it had never seen a similar patient. Since the purpose of this thesis was to develop a system which improved with *experience*, it is neither surprising nor disappointing that the system failed to produce a solution for unfamiliar cases. When CASEY produced an incorrect causal explanation, it was often the result of attributing a mostly-correct causal explanation to an incorrect primary cause.[1] CASEY should have an additional evidence principle that uses the Heart Failure program to determine whether a particular primary cause is still the most likely primary cause given the new evidence.

CASEY's efficiency was evaluated by comparing the number of states (of the Heart Failure program) it examined to the number states examined by the Heart Failure program for the same patient. The number of states examined by

[1] the state at the beginning of the causal chain.

Patient	HF states	CASEY states
Adam	7K	126
Andrea	600	15
Bertha	7K	836
David	76K	674
Edith	13K	649
Francis	30K	486
Heywood	1700	123
Jethro	43K	366
Kalman	13K	1034
Karl	25K	952
Larry	7K	578
Natalie	13K	13
Thadeus	65K	2
Uri	18K	1100

Table 4.3: **Efficiency comparison for identically-solved cases.**

CASEY was calculated by counting all calls made to Heart Failure program procedures that access data structures representing states. The number of states examined by the Heart Failure program was approximated by counting the number of causal paths evaluated in reaching its solution, and multiplying that number by eleven, which is both the mode and mean of the number of states in a path. The results of the efficiency analysis are shown in Table 4.3 for the cases in which which CASEY's solution was identical to the Heart Failure program's solution. CASEY always examined fewer states than the Heart Failure program by *at least an order of magnitude, and often by two or three orders of magnitude.*

Cases that required relatively more effort by CASEY to solve did not necessarily correspond to cases that the Heart Failure program required a lot of effort to solve. Problems that can be solved quickly by the Heart Failure program have features which are specific to only one (or a small number) of states. Problems that require a lot of effort for the Heart Failure program are those with many symptoms that are evidence for a large number of states, which generate a large number of possible explanations that must be evaluated. By contrast, a simple case for CASEY is one in which there are few differences between the precedent and the new case. A difficult case for CASEY is one in which many differences between the precedent and the new case must be analyzed. A consequence of this difference is that as the number of cases solved by CASEY increases, it requires less effort to solve subsequent cases because it is more likely to find a close match. The Heart Failure program, conversely, cannot increase its efficiency except by re-implementation.

A complete analysis of CASEY's solution for each test case is given in the appendix.

Chapter 5

Discussion

5.1 Strengths of the method

CASEY demonstrates that combining a memory of past cases with reasoning from a causal model can have significant advantages over either method used alone.

- CASEY combines the efficiency of associational reasoning with the improved problem-solving ability of model-based reasoning. It can recognize when a case is routine and when it is not. It efficiently solves routine cases by making small local changes to an existing solution. CASEY can recognize that it does not know how to solve a particular problem. When this occurs, it can solve the case by using the Heart Failure program.

- CASEY's performance improves with experience. It learns to solve more problems efficiently as it is given more problems to solve, because it remembers what it has done in the past. It can improve its knowledge by being corrected.

- CASEY can acquire new knowledge automatically by making generalizations about problems that it has solved. It automatically acquires new associational knowledge by making generalizations about each new case presented to it.

- CASEY's model-based reasoning component is enhanced by the ability of the case-based component to learn new associations and compile detailed reasoning structures into simple associations between features and solutions. This results in both improved performance speed and in improved accuracy of the program as new information is added.

- CASEY's case-based component is improved by the use of a causal model because the model can prove that a retrieved solution will be helpful for a new case. Also, the model can be used to identify important features for matching. This results in the elimination of a major limitation of previous case-based reasoning system, the need to fix the important features for matching.

CASEY can produce the same solutions as the Heart Failure program but more efficiently. CASEY's ability to improve the efficiency of the Heart Failure program hinges on two characteristics of the latter program.

- The Heart Failure program is deterministic, so when presented with two patients whose descriptions are the same, it will produce the same causal explanation. If retrieval is faster than recomputation, and it almost always is, CASEY can save time by remembering a causal explanation rather than generating one.

- The Heart Failure program does not represent all the actual relationships between findings and states in the model.[1] CASEY can learn to identify which findings are important to the solution of the case and which can be ignored, thus developing "essential" descriptions of various diseases: these are the combinations of features for which the Heart Failure program will always produce the same solution. In essence, CASEY *memoizes*[2] the Heart Failure model. But because generalization allows

[1]The Heart Failure program is still under development, so its model is incomplete.

[2]"*Memoization* is a technique that enables a procedure to record ... values that have been previously computed. This technique can make a vast difference in the performance

it to ignore features unimportant to the solution, it memoizes the model using partial descriptions of cases. This is an improvement over the basic memoization scheme because it does not require identical matches.

CASEY learns new associations between features and solutions. As was noted in section 2.7, the Heart Failure model does not represent all the relationships that exist between features and states in the model. Sometimes the relationship between a feature and a state in the model is not known, or is on a more detailed level than the Heart Failure model uses. In this case, the Heart Failure model represents the probability of the feature being associated with a state without actually representing the causal mechanism involved. These are the same types of associations that CASEY can discover. For example, in section 2.7 we saw the example of the feature `(systolic-murmur increases with valsalva)`, which is not associated with any state in the model. In a patient with IHSS, a systolic murmur is heard which increases with valsalva. If a number of patients with this symptom were presented to the program, and subsequently found to have IHSS, CASEY would make a GEN representing patients with a systolic murmur that increases with valsalva and the disease IHSS. If enough patients with the feature and the disease were seen (currently, if $> 2/3$ of the patients with the feature had IHSS) it would use the information in the GEN for prediction. CASEY would predict that patients with a

---

of a program. A memoized procedure maintains a ... table in which values of previous calls are stored using as keys the arguments that produced the values. When the memoized procedure is asked to compute a value, it first checks the table to see if the value is already there and, if so, just returns that value. Otherwise, it computes the value in the ordinary way and stores this in the table." [1]

systolic murmur that increases with valsalva also have IHSS.

CASEY can in principle produce a causal explanation even for problems that the Heart Failure program cannot solve. A problem could lack sufficient information for the Heart Failure program to calculate a solution (i.e. not have enough necessary features in the patient description to deduce any solution), but still have enough information for CASEY to find a matching case (by matching using whatever causal features are present, as well as correlated but non-causal features).

Since determination of important features is based on information in the causal model, it is reasonable to ask why the Heart Failure model is not simply "compiled" to produce all this information in the form of associational rules relating important symptoms and physiological states. In fact, that is exactly what CASEY is doing, but it is compiling the knowledge incrementally, associating features of problems with solutions for the cases it has seen. Also, the Heart Failure program can generate solutions involving multiple diagnoses. To compile all of the Heart Failure program's knowledge taking into account multiple diagnoses would be computationally intractable. CASEY compiles the Heart Failure model for combinations that are observed to occur. The Heart Failure model provides the relative importance of features only for single diagnoses. Because CASEY also makes generalizations about patients who have multiple diagnoses, it can create associational knowledge relating features to solutions involving multiple diseases.

CASEY's causal reasoning ability lets it produce a complete causal analysis of the new case, not simply a reference to a previous solution. As noted in section 2.5, case-based reasoners with no method for evaluating the contribution of different pieces of evidence to a solution cannot guarantee that the retrieved solution is appropriate for the new case. Evaluating differences by use of a causal model improves the likelihood that the retrieved solution applies to the new case. When CASEY justifies the match between the old case and the new case, it demonstrates that although there are differences between the cases, the causal model still supports the retrieved solution.

5.2 Limitations

CASEY's current implementation has some limitations. Most problems presented to the system have a large number of "reasonable" explanations. CASEY does not use all the quantitative information available in the Heart Failure model that would allow it to distinguish between statistically more- and less-likely solutions.

For example, the program is parsimonious about adding additional states to the causal explanation. If a new feature could be attributed to two different physiological states, one of which is already included in the transferred explanation, CASEY will use the state that is already there rather than add a new state. It is possible that a feature has a higher probability of being caused by the state not already in the explanation. The model contains information that CASEY could use to discover this circumstance. CASEY works by modifying one particular solution, rather than by generating many solutions and comparing them, as the Heart Failure program does. This makes it more difficult for CASEY to evaluate the likelihood of its solution being correct compared to other possible explanations for the same data. The Heart Failure program, on the other hand, calculates the probabilities of all possible causal explanations that fit the data, and chooses the one with the highest probability. For certain applications (e.g. geological interpretation [44]), any explanation for the input features is acceptable. In the Heart Failure domain, the users require the most likely explanation. CASEY's justifier could be extended to recognize when the solution it is creating is not the most likely one, in which case it could reject the match. This would require invoking the Heart Failure program in a limited capacity to evaluate the probability of the change being made.

An objection that might be raised by the statistically-minded reader is that CASEY can make predictions based on too few cases. (This is only an issue for generalizations of non-causal features, since the causal features are generalized using the Heart Failure model and therefore are accurate according to the model.) A simple solution to this is to make the program wait until it has seen a desired minimum number of cases before being allowed to make predictions. On the other hand, CASEY's ability to make generalizations could be viewed as "learning quickly." If the program were suddenly placed in the middle of an epidemic, or confronted with a new disease, CASEY would soon realize that it was seeing many similar cases of the same thing, whereas a static program (such as Heart Failure) would be bound by its fixed prior probabilities. A similar complaint is that when it has seen a relatively small number of cases, CASEY can introduce incorrect biases in the importance weights of features, for obvious reasons. The probabilities used in the Heart Failure program are based on studies with larger numbers of patients than have been presented to CASEY. If CASEY were given a larger number of cases (for example, the same number as were used to develop the statistics used in the

Heart Failure program), it would overcome biases in its importance weights that were due to the small sample size of patients that it had encountered. Finally, some readers may be uncomfortable with the notion of reasoning from a single case, believing that this is "anecdotal". In fact, CASEY never reasons from a single case unless it has a perfect match. The retrieved solution is always evaluated for the new case in the context of the Heart Failure model, which represents information distilled from a large number of cases.

5.3 Learning

CASEY learns in several different ways. First, by remembering cases it has already solved, CASEY increases the collection of problems that it can recognize and quickly solve. However, this type of learning is not particularly useful because the knowledge acquired can only be applied to cases exactly like those already seen. CASEY can also make generalizations of the problems it knows how to solve. This is more useful because it allows the program to solve problems that is has not seen before. CASEY uses two generalization techniques, similarity-based generalization and explanation-based generalization.

5.3.1 Learning by generalization

In similarity-based generalization, a program acquires new information by comparing a number of examples and making a generalization defined by their similarities. The assumption in similarity-based generalization is that if a new example matches some features of a generalization, the other features of the generalization might hold also. There is no requirement for any specific domain knowledge in this paradigm. CASEY performs similarity-based generalization when it groups cases together on the basis of some similarity in features. The groups (GENs) allow CASEY to make predictions about certain aspects of new cases that match the common features of the GEN, even though the new cases' descriptions may not be completely specified. The system also learns new associations between features and states as a result of noting similarites between cases.

In explanation-based generalization, features of a single example are "explained" (analyzed) using detailed knowledge of the domain, then the details of the particular example are generalized so that the explanation of their relationship still holds, but unrelated features are ignored. The generalized description can then be used to analyze (presumably more efficiently) subsequent examples presented to the program. The specific example on which explanation-based generalization is performed is usually discarded. CASEY performs explanation-based generalization when it determines the generalized causal variables from a patient's causal explanation and uses these to describe a *class* of patients. For example, from the causal explanation of the patient Sarah (given in section 1.5), CASEY produces the general class of "patients with symptoms of exertional angina and unstable angina".

Similarity-based and explanation-based generalization are both useful in CASEY. Similarity-based generalization relies on coincidence to identify important features of a description, and is essential in domains without a strong causal model – the type of domain to which case-based reasoning has traditionally been applied. Generalizations based on coincidence are less likely to be true than explanation-based generalizations which have been created by reasoning about the underlying model of the domain. Because CASEY has a causal model at its disposition, it can do better than relying solely on similarity-based generalization, as other case-based reasoning programs must. CASEY, where possible, uses its causal model to identify and generalize causally-important features (using explanation-based generalization). However, for some features the model gives no information. Because the model is known to be incomplete, CASEY has no way of determining whether those features are irrelevant or whether they are important but simply missing from the model. Previous explanation-based generalization programs have either (1) been used in domains with small, complete models (e.g. chess, logic design, mathematical integration), (2) used incomplete models with the closed-world assumption, or (3) relied on input descriptions that were noise-free. The effect of all of these restrictions is to ensure that the generalizer is correct in deciding that a feature is irrelevant to the causal explanation. In CASEY's domain, as in most real-world problems, the data is noisy and the models are incomplete. The combination of explanation-based generalization to identify important aspects of the case where possible with similarity-based generalization to fill in gaps in the model and reduce the effects of noise is a logical choice.

5.3.2 Improving on the Heart Failure program

The assumption made during the implementation of CASEY is that there are two kinds of knowledge used by the program: associational knowledge (as, for instance, between a symptom set and a disease) which can be modified by experience, and basic science or "first principles" knowledge, which cannot. The Heart Failure program was assumed to embody first principles knowledge, and therefore was not to be overridden. In fact, the Heart Failure model does contain many basic principles of physiology, but it also contains uncertain knowledge (the probabilities), and its solution method includes heuristics. Thus, it is not always guaranteed to produce the correct solution. If the standard to which CASEY is held were changed instead to that of an expert user/physician, CASEY could learn to do better than the Heart Failure program. This would work in the following way: If the user did not like the answer produced by CASEY (or the Heart Failure program), she could enter her own solution, which would be stored. The next time a similar case was encountered, CASEY would remember the solution preferred by the user.

5.4 Indexing

Indexing refers to choosing the features of a case that will be used as pointers to it in the memory structure. Selecting the indices of a case for storage determines the ways a case will be stored in the memory structure. When retrieving past similar cases, the indices selected determine which aspects of the case will be matched against the cases in memory.

Previous case-based reasoning systems have emphasized constraining the number of indices (e.g. [17], [14], [13]). If the system designer decides *a priori* that certain indices will be discarded, the program will never have a chance to use those indices. The importance of an index may not be apparent in advance. That is why CASEY uses all the features of a case for indexing. Some may turn out not to be useful and are discounted, but none are eliminated *a priori*.

When case-based reasoning was first developed, memory constraints were such that total indexing resulted in unacceptably slow performance. Changes in technology have resulted in greatly increased memory capacity, while at the same time advances in parallel computing allow greatly reduced retrieval times with less memory usage [20]. Indexing by every feature is feasible from the technological point of view, as well as from the cognitive point of view.

5.5 Defaults and exceptions

The term *default reasoning* is commonly used in Artificial Intelligence to refer to the type of reasoning of the form "Unless there exists evidence to the contrary, assume ... holds." *Exceptions* refer to examples of a class which in some way violate the description of the class. For example, consider the question: "If Tweety is a bird, can Tweety fly?" In the absence of any other evidence, we answer this question using our default knowledge of birds and answer "yes." However, if we are told that Tweety is a penguin, we use our knowledge of exceptions to the rule that "birds fly" and answer "no."

CASEY's group-and-differentiate memory structure [17] provides a convenient representation for reasoning about defaults and exceptions. The memory structure automatically provides a representation for creating knowledge about defaults and exceptions. GENs hold information that is common to a substantial portion of[3] the cases that are indexed under that GEN. Any exceptions

[3] The exact fraction is implementation dependent. In CASEY, it is 2/3.

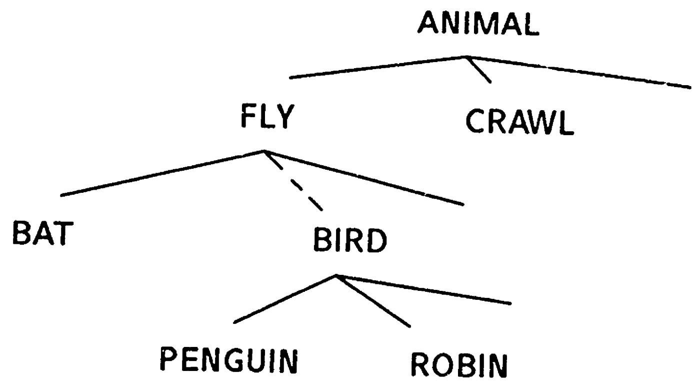


Figure 5.1: "Penguins don't fly" represented in a typical inheritance hierarchy.

that the memory knows about are explicitly represented by the diffs of the gen. The diffs distinguish those cases (or more specific generalizations) indexed within the GEN that do not fit some the default knowledge held in the GEN. Furthermore, diffs specifically identify the ways in which the exceptions differ from the GEN. In typical knowledge representation hierarchies, Reiter [38] observes that there is no way to establish inheritance from any node above an exception to any node below one. For example, in the hierarchy shown in Figure 5.1, there is no way to establish that a penguin is an animal. By contrast, the same information is easily represented in a group-and-differentiate memory structure, as shown in Figure 5.2. The exact way in which penguins differ from the default bird knowledge is identified. Thus there is no problem determining which aspects of the default bird knowledge that penguins do inherit. The diffs precisely indicate where the system knows that its default knowledge fails to hold. Since the generalizations are created in response to

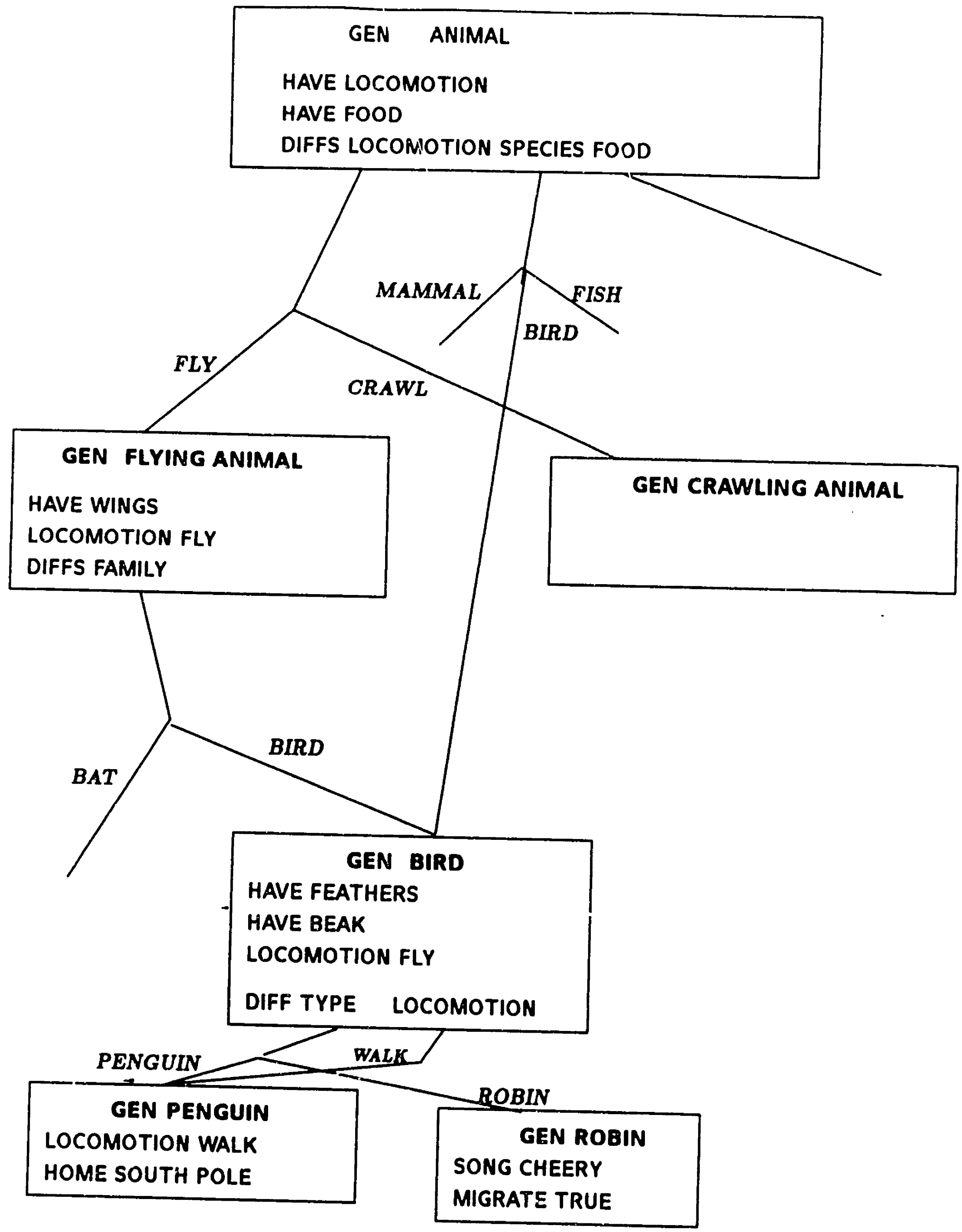


Figure 5.2: "Penguins don't fly" represented in a group-and differentiate memory structure.

examples presented to the program, they do not necessarily reflect reality. If the program were only presented with bird examples penguin, ostrich, emu, and dodo, it would conclude that the typical bird uses walking as its means of locomotion. This is a reasonable default given the information it has. The behavior of the memory system represents a form of commonsense default reasoning.

5.6 Relation to formal theories of diagnosis

Recently, a number of techniques have been developed which describe methods for diagnosing systems with multiple disorders or *faults* [37, 10, 39]. In each of these systems, competing diagnostic hypotheses are represented as sets of individual diagnoses. Each individual diagnosis identifies faulty components. When the individual diagnoses in a set are considered together, they explain all of the observed symptoms. In [10, 39], only the minimal sets which cover the observations (*minimal conflict sets*) are considered during diagnosis. A set is said to be minimal if no subset of the set can account for all the symptoms. A diagnosis in [10, 39] is defined as a *minimal cover set* for the findings. The requirements of a minimal cover set are that it accounts for all the observed symptoms, and no subset of it accounts for all the symptoms. All these techniques claim to be general. [10, 39] are demonstrated in the domain of electrical circuit troubleshooting, while [37] has been applied to medical diagnosis. As they are quite similar, we will only discuss one technique in general, that of Reiter [39].

CASEY's set of evidence-states for a given input feature is clearly equivalent to Reiter's *conflict set.* CASEY's set of generalized causal features appears to corresponds to Reiter's *minimal cover set.* However, they are different.[4] There are two characteristics of CASEY's problem domain, and in fact of many real-world domains, which directly rule out use of the the set cover technique. First, the generalized causal features need not account for every observed symptom. This is because some features are noise and need not be explained (covered) by the diagnosis. Second, the set cover technique assumes set additivity of observations [35]. It does not take into consideration the case in which features interact so as to mask each other. For example, a patient may display two or more simultaneous physiological processes, one acting to raise his blood pressure and a compensatory mechanism acting to lower it. A set may appear to be non-minimal precisely because the additional hypotheses are acting to explain findings whose net effect is to cancel each other. For these reasons, CASEY's generalized causal features need neither account for all observed symptoms, nor be minimal.

Thus the techniques developed for relatively straightforward domains are not adequate to reason about the noise and complexities of real-world problems.

[4]We will ignore for the purposes of this discussion the obvious difference, which is that CASEY has causal chains that extend backwards from the generalized causal features, and diagnosis states are often found higher up in the causal chain.

5.7 CBR vs. generate-and-test

In many traditional associational reasoners, an initial solution is generated by combining the results of many nearly-independent associational rules, which map features of the problem onto fragments of solutions. If parts of the problem interact, there is no coherence guaranteed in the hypothesis. It must be tested to ensure that the parts add up to a plausible whole. CASEY starts out by finding a complete solution to a similar problem. This solution is guaranteed to be coherent for the problem it originally solved. The problem CASEY must answer is, "is this solution still valid for a slightly different input?" By checking the relationship between the evidence in the old case and the evidence in the new state with respect to states in the model, CASEY can prove that the new evidence can be explained by the repaired solution. There is no need for a separate test step.

The test step in generate-and-test systems involves running the entire initial solution through a simulator based on the model of the domain to make sure that it results in the same state observed in the problem statement (the *goal* state). Of course, *verifying* through use of the model that a hypothesized solution results in the goal state is less work than *generating* that solution using the model. However, it still requires more work than is done in CASEY's justification step. Using CASEY's method, only parts of the solution which depend of features which differ in the old and new problem must be evaluated. If CASEY judges the differences insignificant or repairable, it means that the Heart Failure model allows the causal path to the findings described in the solution. This holds as long as the evidence principles make no assumptions not sanctioned by the CHF model (and in fact, they are more conservative than the model in this respect). Conversely, if CASEY judges a difference to be irrepairable, it means that within limits of the assumptions made by the evidence principles, the two problems are not equivalent, and the retrieved solution should not be used.

5.8 Generality of the Method

Although CASEY was designed for providing causal explanations, diagnoses, and therapy recommendations in the domain of heart failure, the memory structure and evidence principles used in the system do not depend in any way on the specific domain information in the Heart Failure model. The evidence principles, however, do depend on the *form* of the model, namely a causal inference network. The techniques in CASEY could be generalized to apply to models of the same form as the Heart Failure model.

5.8.1 Requirements

CASEY requires the following information for its reasoning:

- a set of features that can be used to describe some problem,
- a set of states that can explain these observations,
- specific information about features:
  - the set of possible values for this feature,
  - the set of states that use this feature as evidence,
- specific information about states:
  - the set of features that are evidence for this state,
  - features that rule this state out,
  - a list of states that cause this state,

– a list of states that this state causes.

Formally, in order to use CASEY, a model must provide the following information:

$\mathcal{S}$, a finite set of *states*.

$\mathcal{F}$, a finite set of *features* which can be evidence for the states in $\mathcal{S}$. $f \in \mathcal{F}$ is what up till now has been referred to as a feature-value pair.

$C \subseteq (\mathcal{S} \times \mathcal{F}) \cup (\mathcal{S} \times \mathcal{S})$. The relation C in the Heart Failure model is used to imply causality. In fact, it is not even necessary that the relation be causal. For CASEY's evidence principles it is sufficient that $(s, f) \in C$ is associational and s temporally preceed f (similarly for $(s_1, s_2) \in C$).

The problem presented to CASEY is then

$\mathcal{F}^+ \subseteq \mathcal{F}$, some subset of the features which has been observed.

There is other information which CASEY can use (and does) in its reasoning, in particular probabilistic knowledge of the relative likelihooods of different diagnoses (which it can get from the Heart Failure model), however these are not essential to its reasoning.

5.8.2 Other aspects of generalization

The criteria by which a match is ruled out can be made more or less stringent. For example, a match could be ruled out if any state in a retrieved case was ruled out. CASEY is very accommodating of differences between new and retrieved cases because the system currently does not have enough cases to allow very stringent similarity requirements. A more conservative criterion might be implemented if the system acquired a great many more cases. Having a more conservative criterion might increase the number of times that CASEY failed to find an acceptable match. Conversely, more cases would increase the chances of CASEY's finding a matching case, and matches would tend to be closer, meaning that CASEY would do less modification to a retrieved causal explanation. Also, with more cases CASEY would call the Heart Failure program less often, which would make the system more efficient.

The ability of the technique to produce a good solution depends in part on selecting a good precedent case. Much research has been done in this area (for example [18], [43], [46], [47], [16]). CASEY uses a novel matching algorithm specifically designed for reasoning about causal explanations.

5.8.3 Application to more complex models

An issue related to the generalizability of the method is the applicability of the techniques developed in CASEY to different models. Specifically, as models change, can CASEY adapt?

One way in which models might change is by becoming more precise. As models become more precise, features which were different but roughly equivalent may no longer be acceptably substituted. CASEY will have to refine its evidence principles to reflect the increased precision of the model.

Techniques are being developed that allow qualitative models to predict the long-term behavior of a system rather than the immediate behavior [26], [41]. CASEY could adapt to this type of model by acquiring new vocabulary to describe the long-term behaviors. This could be qualitative result (i.e. oscillation) or quantitative result. Such a system would attempt to predict long term behavior by recognizing certain features in the problem.

Whereas the Heart Failure program describes its domain model through a combination of associations and equations, many systems are described by models consisting completely of equations. If this were the basis of the model, CASEY would require additional techniques to match a new case to a precedent. The behavior of these equations is predicted from their initial conditions. After solving a case, CASEY would have to do an "analytical generalization" of the model, wherein it determined the intervals of initial variables for which the model's equations give the same result (much as CASEY currently does for numerically-valued features). The criterion for matching a case would not be identical initial conditions, but the features would have to be in the same interval. This analysis could become very complicated if the features of the model interact.

An example of a purely quantitative domain in which CASEY's techniques might be applied is in remembering the results of past linear programming (LP) optimization model executions. CASEY handles numerically-valued features by determining the ranges of those features which, according to the model, result in the same final state. In LP optimization, the ranges over which the solution holds for a particular model can be derived by parametric analysis (a different form of model-based reasoning). CASEY could generalize a particular model to describe the class of models which will result in the same solution. Of course, this would require an exact match between the new case and the class since evaluating the results of changes outside the range would require pivoting.

Models are increasing being called upon to handle temporal data (e.g.

"worsening high blood pressure," "pneumonia one month ago"). CASEY would most likely be able to handle this issue by using qualitative time regions such as "in the recent past", "immediately preceding admission", and "in the past", to qualify features in the patient description. The significance of the qualitative region may be disease-dependent (e.g. tuberculosis one year ago is seen differently than pneumonia one year ago). This technique was also used in PIP [36], another medical reasoning program. But for more sophisticated models of time, a further analysis will be required.

5.9 Future Work

There are many ways in which the work described in this thesis could be extended in the future. This section describes several of them.

Using book-knowledge. An important feature of CASEY's memory structure is that it can be used to store "book knowledge" as well as experiential knowledge. CASEY's memory does not have to start out totally empty, but could take advantage of pre-existing sources of medical knowledge to create prototype cases of diseases.[5] This would have several advantages.

1. It would save CASEY the effort of learning known causal explanations for common problems.

2. In the case of rare problems, the program would not have to wait until it had been presented with a case before gaining the needed knowledge.

3. CASEY would perform better faster because fewer cases would be needed to ensure coverage of the domain.

Book knowledge would be stored in GENs as the framework throughout which cases are distributed. The cases would represent unusual presentations and combinations of diseases. CASEY would still learn from cases that fit the classic descriptions, because those cases would strenghten the association between symptoms and the disease.

[5] Much like a medical student at the beginning of the third year, who has book knowledge but no clinical experience.

```
(defnode  constrictive-pericarditis
 goal     diagnosis
 causes   (primary .01
          P+ (pericardial-calcification :prob .5))
 measure  ((EKG (prob non-specific-t-changes .8))
          (EKG (prob low-qrs .5))
          (echocardiography (prob pericardial-thickening .7))
          (calcification (prob pericardial .3))
          (cat-scan (prob pericardial-thickening .7))
          (abdomen (prob (or ascites hepatosplenomegaly) .3))
          (jugular-pulse (prob inspiratory-increase .7))
          (known-diagnoses (prob constrictive-pericarditis .5))))
```

Figure 5.3: **The Heart Failure model's information about constrictive pericarditis.**

The book knowledge in CASEY would be stored in the same way as GENs derived from experience. A difference would be that the weights for the common features of the GEN could be set initially to reflect the probability of seeing the feature in the given state. Also, the weight of the path to each such GEN could be set initially to be the *a priori* probability of this state in the population.

Long[6] points out that all the book knowledge necessary to creat prototype cases is available in the Heart Failure model. It is possible to generate the prototype cases directly from the model, by building a compiler that reads the information in the Heart Failure model, extracts the relevant information, and creates GENs that hold that information. For example, the knowledge about the disease constictive pericarditis contained in the Heart Failure model is shown in Figure 5.3. A compiler could automatically produce the GEN

[6]personal communication

```
GEN CONSTRICTIVE-PERICARDITIS        NODES: 0

FEATURES
(ekg non-specific-t-changes)
(echocardiography pericardial-thickening)
(cat-scan pericardial-thickening)
(jugular-pulse inspiratory-increase)

CAUSAL

DIFFS
EKG  CALCIFICATION  ABDOMEN  KNOWN-DIAGNOSES
```

Figure 5.4: **A disease prototype created from information in the Heart Failure model.**

shown in Figure 5.4. Notice that only measures with a probability greater than 0.6 (which indicates the probability that patients with constrictive pericarditis have this finding) would be placed in the GEN as features, since these are meant to represent features present in the description of at least 2/3 of the cases stored in this GEN. The other measures could be stored as diffs; these represent ways in which cases could vary from the classic presentation.

Reasoning on multiple levels of detail. The process of abstracting from specific evidence for states to generalized causal features is recursive.[7] The generalized causal features can be be further generalized to be evidence for states at another level of description, as is shown in Figure 5.5. For example,

[7]William Long, personal communication.

`dyspnea on-exertion` is generalized to "evidence-of limited cardiac output," which in turn is evidence for left-sided heart failure, a pathophysiological state not represented in the model's current level of description. This is similar to the multiple-level patient-specific model representation pioneered in ABEL, but which has not been used extensively due to the difficulty of constructing multiple models. The techniques in CASEY provide a means by which these multiple-level models could be *generated* automatically.

In order to *reason* on multiple levels of description, CASEY would require additional indices, one for each generaliz[illegible]iption that could be used to describe each new level. Cases could be indexed at all levels of description in the same generalization structure (which is the equivalent of reasoning on multiple levels of detail simultaneously), or each level could be stored in a separate structure (in the same way as the separation between the FEATURE-GEN and the CAUSAL-GEN is currently maintained). This would give the problem solver more control over the level of detail being used at a given time.

Combining retrieved solutions. CASEY could not solve several test cases that involved multiple, non-interacting diagnoses because it had not seen a case with the particular combination of diagnoses before. For most of these test cases, CASEY found precedents which could account for different parts of the solution, but it had nc way of combining them. The ability to combine solutions from several precedents is a logical next step for the program.

Integrating multiple models. Generalizing the causal explanations produces partial explanations that explain features that the new case and retrieved

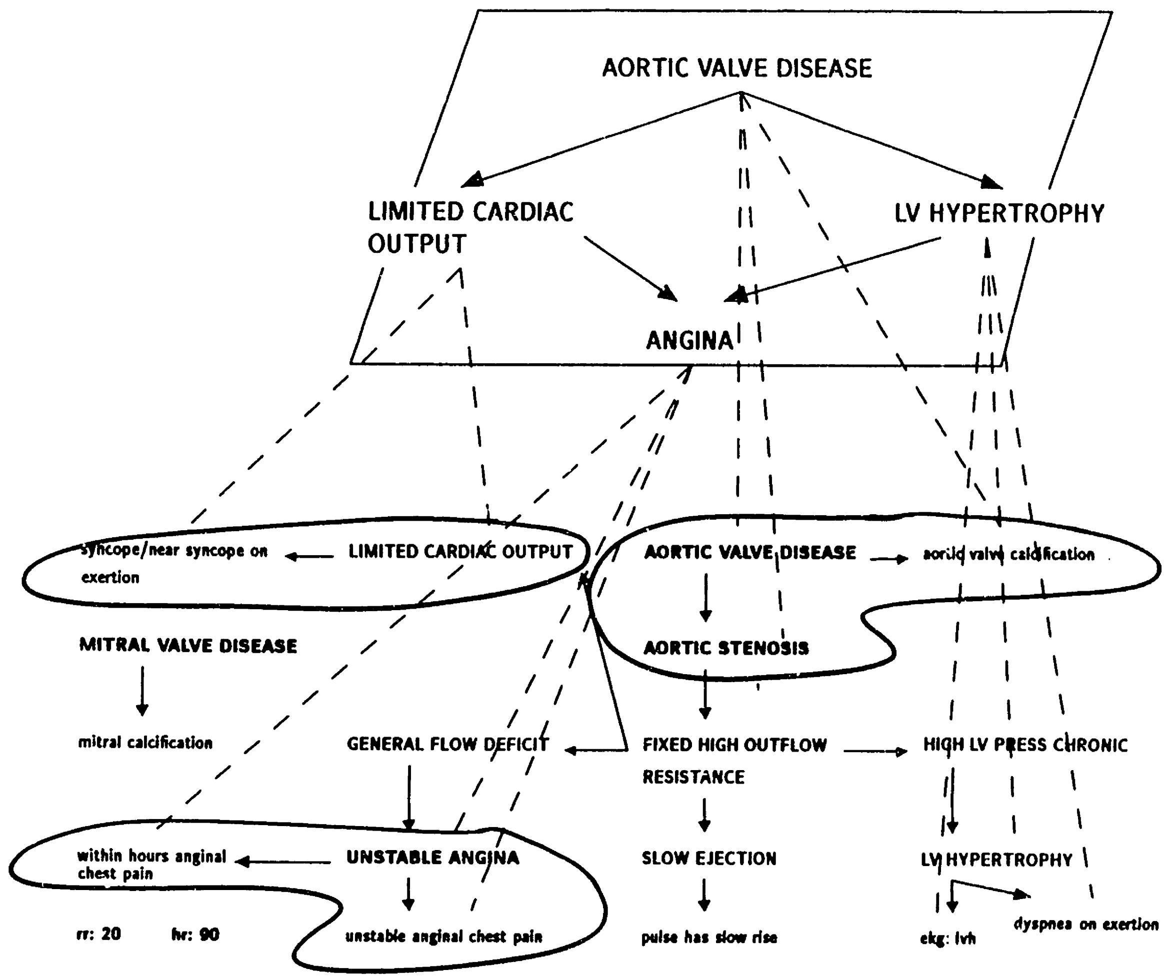


Figure 5.5: Using generalized causal features recursively to reason on multiple levels of detail.

case have in common. CASEY can thus produce a partial solution for a problem even in the absence of enough information to explain the whole problem. This suggests using CASEY to integrate several causal models developed for different domains, where no one model could account for all the features in the problem.

Other tasks. The evidence principles and similarity metric in CASEY have been optimized for the task of finding a case with the same causal explanation. If CASEY were applied to a different task, the same reasoning structure would hold, but it would need different details. For example, it would probably still be true that the important features for each case should be individually determined, but these features would not necessarily be the generalized causal features that are used in CASEY. Similarly, it would still be a good idea to justify solutions before transferring them, even if the task were different (e.g. the robot-planning task in NODDY [3], geometry problem-solving in [6]). CASEY's evidence principles were designed to justify transferring causal explanations; other tasks may require different types of justifications.

Learning evidence principles. A challenging extension of the system would be to present CASEY with two cases that are known to be similar, and have it develop the evidence principles itself, by examining similarities between the cases. This would probably require a combination of both similarity-based and explanation-based learning techniques.

Learning more from mistakes. When the user rejects a solution produced by CASEY, the program stores the preferred solution. Storing the solution preferred by they user is one way in which CASEY learns from its mistakes, because on a future similar patient it will return the solution that the user preferred. It does not, however, make use of the information to determine what mistake it made in arriving at the faulty solution. If the user provided the correct causal explanation, CASEY could find the differences between the correct solution and the incorrect one, and examine the model to determine what knowledge it had used to make a faulty substitution, and then remember that the substitution was not allowable. This would reduce the chances of CASEY making an incorrect substitution.

Improving the Heart Failure model. Occasionally the Heart Failure program gives incorrect answers (as judged by the physician user). A interesting use of CASEY would be to program it to identify the information in the Heart Failure model that might have led to the faulty conclusion. Again, this would require CASEY to examine the the two solutions and determine what knowledge in the Heart Failure program was responsible.

Comparing alternatives. CASEY currently retrieves a number of precedents for each new case, and only evaluates them one at a time, until it has found a satisfactory match. An alternative would be for CASEY to justify *all* the retrieved cases for a new case. Some will be ruled out and some will remain. Among the ones that remain, the causal explanations might not all be the same.

- CASEY could present the various causal explanations as alternatives (because there really is a variety of reasonable explanations for most patient descriptions.)
- Also, CASEY could find the overlap among the different causal explanations. The part of the causal explanation in the overlap is strongly indicated in the patient, because it is common to all the explanations.
- If there are different treatments depending on different causal explanations, for example systolic vs. diastolic failure, CASEY can suggest that these are the states that must be distinguished.
- If there is no difference in the treatment consequence of the different causal explanations, the user could feel confident in his recommendations even if he isn't sure that the causal explanation is exactly right.

Critiquing. A final suggestion is to use CASEY for critiquing. CASEY could be used to critique a user's diagnosis or therapy plan for a patient by commenting on the similarity (or dissimilarity) between that patient and a similar case recalled by the program. For example, the user could propose a therapy plan for a new patient. CASEY could find similar therapy plans by looking in the THERAPY-GEN. This structure stores patients according to the therapy recommended for them. CASEY could critique the user's plan by comparing the new patient to the retrieved patient, and using additional information in the model, if applicable. For example, CASEY might state, "This plan was used on patient X but the new patient has the following differences which make that plan inappropriate...". This use of CASEY would be similar to ROUNDSMAN [40], but would have the advantage of being able to examine differences between patients in the context of the Heart Failure model. This application of CASEY would be especially useful in physician training.

5.10 Conclusions

CASEY *integrates* associational reasoning, model-based reasoning, and learning techniques in a program which is efficient, can learn from its experiences, and solves commonly-seen problems quickly, while maintaining the ability to reason using a detailed knowledge of the domain when necessary. Furthermore, the methods used by the system are domain-independent and should be generally applicable in other domains with models of a similar form.

CASEY starts out with a model and an empty memory; it develops all its associational knowledge of the domain through experience. New associational knowledge can be derived from remembering past cases and by generalizing from many similar past examples. CASEY uses both similarity-based generalization and explanation-based generalization. Explanation-based generalization takes advantage of the causal information in the Heart Failure model. Similarity-based generalization is also useful because the Heart Failure model is incomplete and noisy.

CASEY overcomes some of the major weaknesses of case-based reasoning through its use of a causal model of the domain. First, the model identifies the important features for matching, and this is done individually for each case. Second, CASEY can prove that a retrieved solution is applicable to the new case by analyzing its differences from the new case in the context of the model.

CASEY overcomes the speed limitation of model-based reasoning by remembering a previous similar case and making small changes to its solution. It overcomes the inability of associational reasoning to deal with unanticipated problems by recognizing when it has not seen a similar problem before, and using model-based reasoning in those circumstances. Because of its "group and differentiate" representation of previous cases, CASEY knows exactly how the new problem and the retrieved problem differ. It can then evaluate the significance of these differences in the context of the model. Determining that a difference is significant based on the model is equivalent to an associational system determining that a piece of its knowledge does not apply. Other associational reasoning programs, however, do not have a way to identify that the problem to which the associational knowledge currently is being applied differs from the situation in which that knowledge was intended to apply.

The group-and-differentiate memory structure used by CASEY lets the program know where its associational knowledge might not apply. The model lets CASEY determine whether the flaw is indeed fatal, and if not, repair strategies let CASEY adapt a solution to the new case. If CASEY indeed cannot solve a problem itself, it can recognize this and call upon its model-based component to solve the problem.

Bibliography

[1] Harold Abelson and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, 1985.

[2] Richard Alterman. An adaptive planner. In *Proceedings of the National Conference on Artificial Intelligence*, 1986.

[3] Peter M. Andreae. Justified generalization: Acquiring procedures from examples. AI-TR 834, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, January 1985.

[4] Kevin D. Ashley and Edwina L. Rissland. Compare and contrast, a test of expertise. In *Proceedings of the National Conference on Artificial Intelligence.* American Association for Artificial Intelligence, 1987.

[5] William M. Bain. A case-based reasoning system for subjective assessment. In *Proceedings of the National Conference on Artificial Intelligence*, pages 523–527. American Association for Artificial Intelligence, 1986.

[6] Richard Brown. Use of analogy to achieve new expertise. Technical Report AI-TR-403, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1977.

[7] Jaime G. Carbonell. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. Technical Report CMU-CS-85-115, Carnegie-Mellon University, Department of Computer Science, 1985.

[8] Randall Davis. Expert systems: Where are we? and where do we go from here? *AI Magazine*, 3(2):3–22, 1982. Also appeared as MIT A.I. Memo No. 665, June 1982.

[9] Randall Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.

[10] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[11] G. DeJong and R. Mooney. Explanation-based learning: and alternative view. *Machine Learning*, 1(2):145–176, 1986.

[12] Kenneth D. Forbus. Qualitative process theory. AIM 664, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, February 1982.

[13] Kristian Hammond. *Case-based Planning: An Integrated Theory of Planning, Learning and Memory.* PhD thesis, Yale University, 1986.

[14] Larry Hunter. Indexing with knowledge goals: How to put knowing what you know to use. In *Proceedings of the Forth International Workshop in Machine Learning*, 1987.

[15] Peter D. Karp and David C. Wilkins. An analysis of the deep/shallow distinction for expert systems. Knowledge Systems Laboratory Working Paper KSL-86-32, Stanford University, April 1986.

[16] A. M. Kass, D. B. Leake, and C. C. Owens. Swale: A program that explains. In *Explanation Patterns: Understanding Mechanically and Creatively.* Lawrence Erlbaum Associates, Hillside, NJ, 1986.

[17] Janet L. Kolodner. Maintaining organization in a dymanic long-term memory. *Cognitive Science*, 7:243–280, 1983.

[18] Janet L. Kolodner. Reconstructive memory: A computer model. *Cognitive Science*, 7:281–328, 1983.

[19] Janet L. Kolodner. Experiential processes in natural problem solving. Technical Report GIT-ICS-85/22, School of Information and Computer Science, Georgia Institute of Technology, 1985.

[20] Janet L. Kolodner. Retrieving events from a case memory: A parallel implementation. Unpublished manuscript, 1988.

[21] Janet L. Kolodner and Robert M. Kolodner. Using experience in clinical problem solving: Introduction and framework. Technical Report GIT-ICS-85/21, School of Information and Computer Science, Georgia Institute of Technology, 1983.

[22] Janet L. Kolodner and Chris Riesbeck. *Experience, Memory, and Reasoning.* Lawrence Erlbaum Associates, Inc., Hillside, NJ, 1986.

[23] Janet L. Kolodner, Robert L. Simpson Jr., and Katia Sycara-Cyranski. A process model of case-based reasoning in problem solving. In *Proceedings of the National Conference on Artificial Intelligence*, pages 284–290. American Association for Artificial Intelligence, 1985.

[24] P. A. Koton. Empirical and model-based reasoning in expert systems. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 297–299, August 1985.

[25] Phyllis A. Koton. Heuristic classification: Does it have any heuristic value? Unpublished manuscript, 1987.

[26] Benjamin Kuipers. Abstraction by time-scale in qualitative simulation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 621–625. American Association for Artificial Intelligence, 1987.

[27] Michael Lebowitz. Concept learning in a rich input domain: Generalization-based memory. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning*, volume 2. Tioga Publishing Company, Palo Alto, CA, 1983.

[28] Michael Lebowitz. Not the path to perdition: The utility of similarity-based learning. In *AAAI.* American Association for Artificial Intelligence, May 1986.

[29] W. J. Long, S. Naimi, M. G. Criscitiello, and S. Kurzrok. Reasoning about therapy from a physiological model. In R. Salamon, B. Blum, and

M. Jørgensen, editors, *MEDINFO 86: Proceedings of the Fifth Conference on Medical Informatics*, pages 756–760, Washington, October 1986. North-Holland.

[30] William J. Long, Shapur Naimi, M. G. Criscitiello, and Robert Jayes. The development and use of a causal model for reasoning about heart failure. In *Symposium on Computer Applications in Medical Care*, pages 30–36. IEEE, November 1987.

[31] Ryszard S. Michalski and J. B. Larson. Selection of most representative training examples and incremental generation of vl1 hypotheses: The underlying methodology and the descriptions of programs ESEL and AQ11. Computer Science Dept. Report 867, University of Illinois, 1978.

[32] Randolph Miller, Fred E. Masarie, and J. D. Myers. Quick medical reference (QMR) for diagnostic assistance. *M.D. Computing*, 3:34–48, 1986.

[33] Tom M. Mitchell, R. M. Keller, and Smadar T. Kedar-Cabelli. Explanation-based generalization: a unifying view. *Machine Learning*, 1(1):47–80, 1986.

[34] Ramesh S. Patil. Causal representation of patient illness for electrolyte and acid-base diagnosis. TR 267, Massachussetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, October 1981.

[35] Ramesh S. Patil. Artificial intelligence techniques for diagnostic reasoning in medicine. Survey talk presented at AAAI87, 1987.

[36] Stephen G. Pauker and Peter Szolovits. Analyzing and simulating taking the history of the present illness: Context formation. In W. Schneider and A. L. Sagvall-Hein, editors, *Computational Linguistics in Medicine*, pages 109–118. North Holland, 1977.

[37] James A. Reggia, Dana S. Nau, and Pearl Y. Wang. Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, 19:437–460, 1983.

[38] Raymond Reiter. On reasoning by default. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 402–410. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1985.

[39] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–96, 1987.

[40] Glenn D. Rennels. *A Computational Model of Reasoning from the Clinical Literature*, volume 32 of *Lecture Notes in Medical Informatics*. Springer-Verlag, 1987.

[41] Elisha Peretz Sacks. *Automatic Qualitative Analysis of Ordinary Differential Equations Using Piecewise Linear Approximations*. PhD thesis, Massachussetts Institute of Technology, February 1988.

[42] Roger C. Schank. *Dynamic Memory: A Theory of Learning in People and Computers*. Cambridge University Press, Cambridge, 1982.

[43] Roger C. Schank, G. C. Collins, and Larry Hunter. Transcending inductive category formation in learning. *Behavioral and Brain Sciences*, 9:639–686, 1986.

[44] Reid Simmons and Randall Davis. Generate, test, and debug: Combining associational rules and causal models. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1071–1078, 1987.

[45] Reid Gordon Simmons. *Generate, Test, and Debug: A Paradigm for Solving Interpretation and Planning Problems*. PhD thesis, Massachussetts Institute of Technology, 1988.

[46] Robert L. Simpson. A computer model of case-based reasoning in problem solving: An investigation in the domain of dispute mediation. Technical Report GIT-ICS-85/18, Georgia Institute of Technology, 1985.

[47] Craig Stanfill and David Waltz. Towards memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[48] E. P. Sycara. Resolving adversarial conflicts: An approach integrating case-based and analytic methods. Technical Report GIT-ICS-87/26, Georgia Institute of Technology, 1987.

[49] Daniel S. Weld. Combining discrete and continuous process models. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 140–143, 1985.

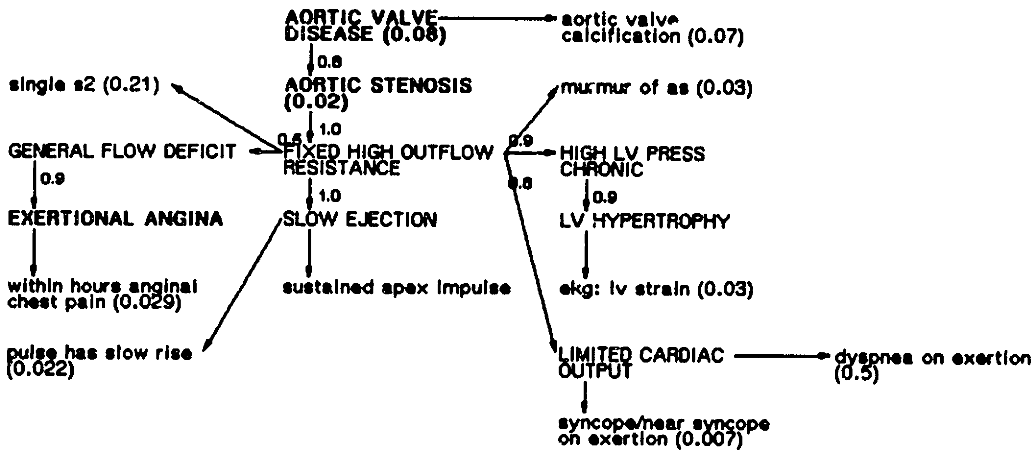
[50] Patrick H. Winston. Learning and reasoning by analogy: The details. AIM 520, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, 1980.

[51] Patrick H. Winston. Learning new principles from precedents and exercises: The details. AIM 632, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, 1981.

[52] Patrick H. Winston. Learning by augmenting rules and accumulating censors. AIM 678, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA, 02139, 1982.

Patient: ADAM

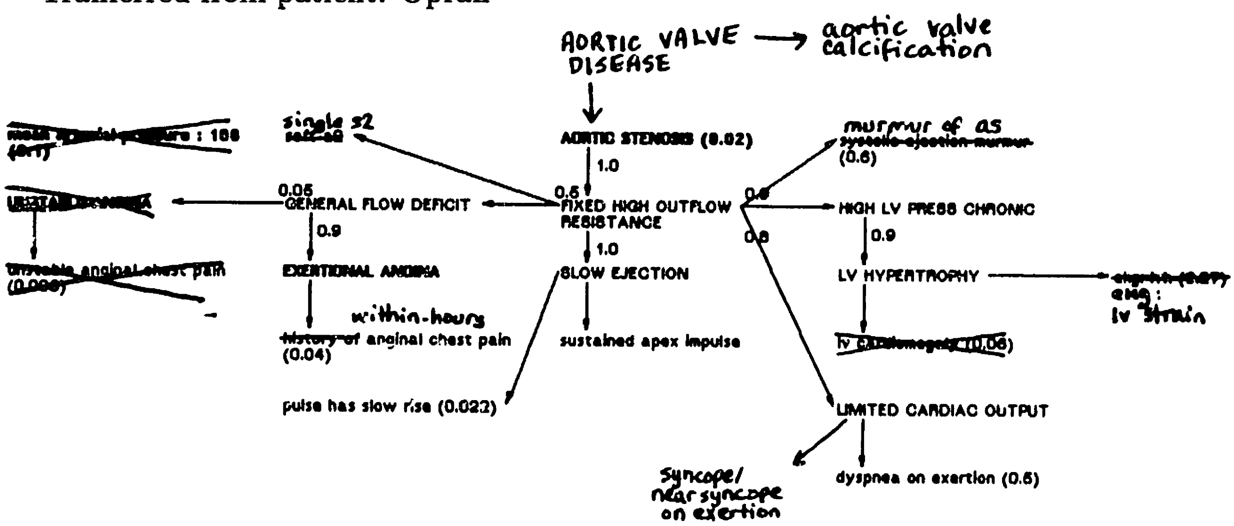
Heart Failure program's solution:



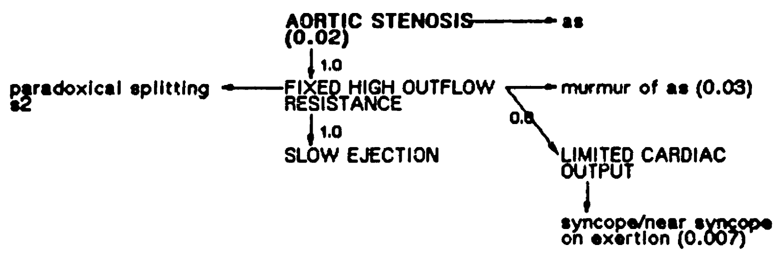
CASEY's result: identical.

Unexplained features: none.

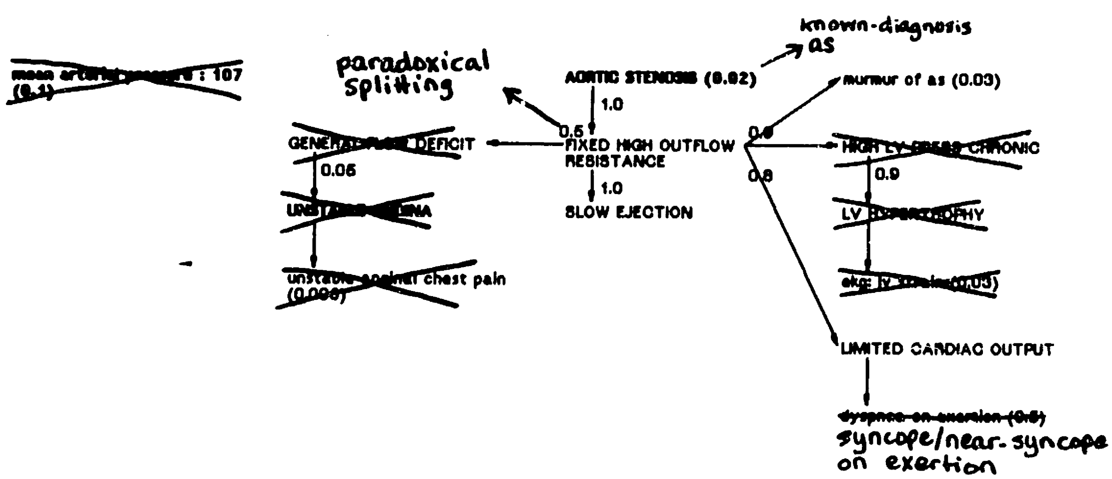
Tranferred from patient: Oprah



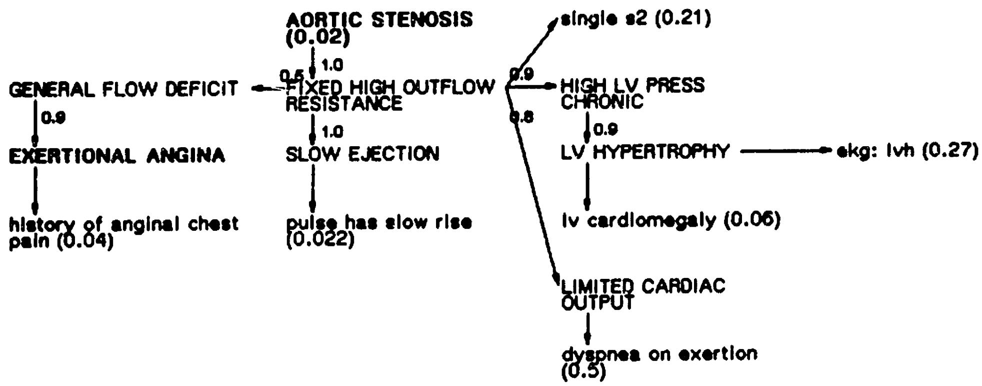
Patient: ANDREA
Heart Failure program's solution:



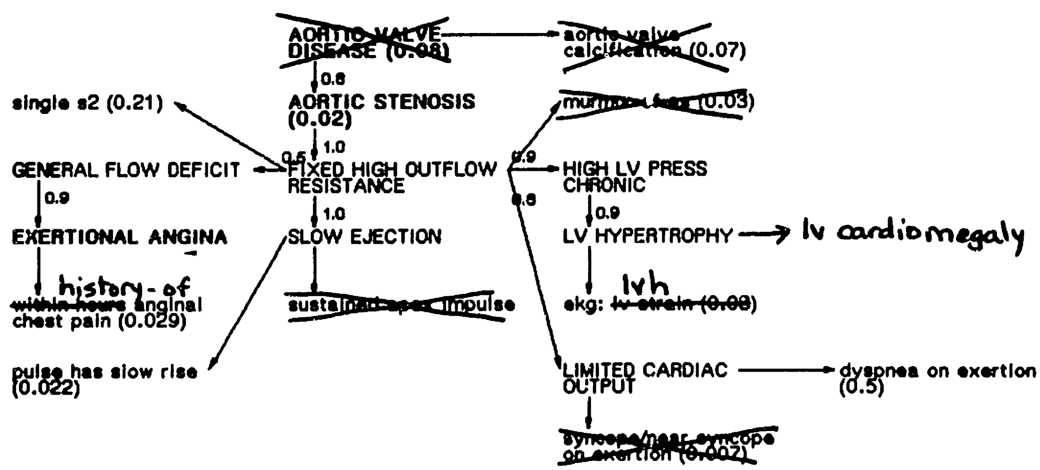
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: David



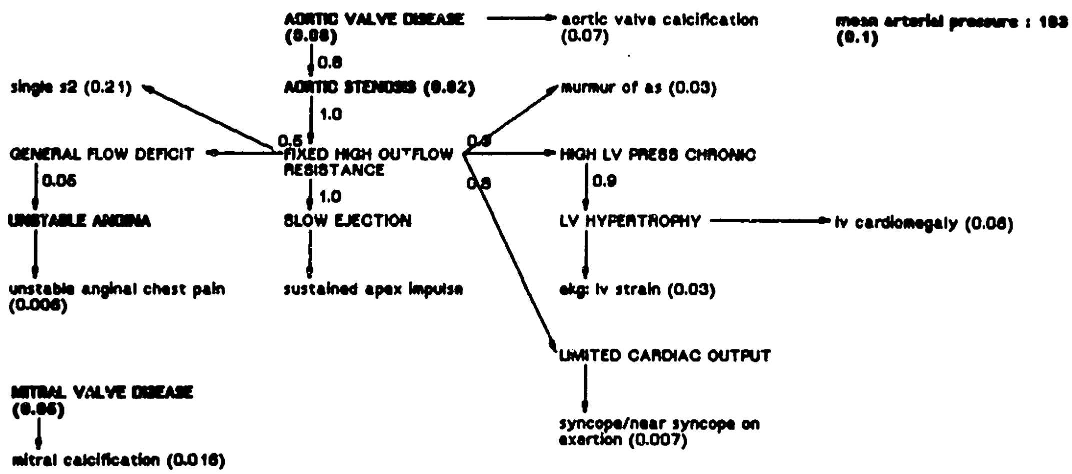
Patient: BERTHA
Heart Failure program's solution:



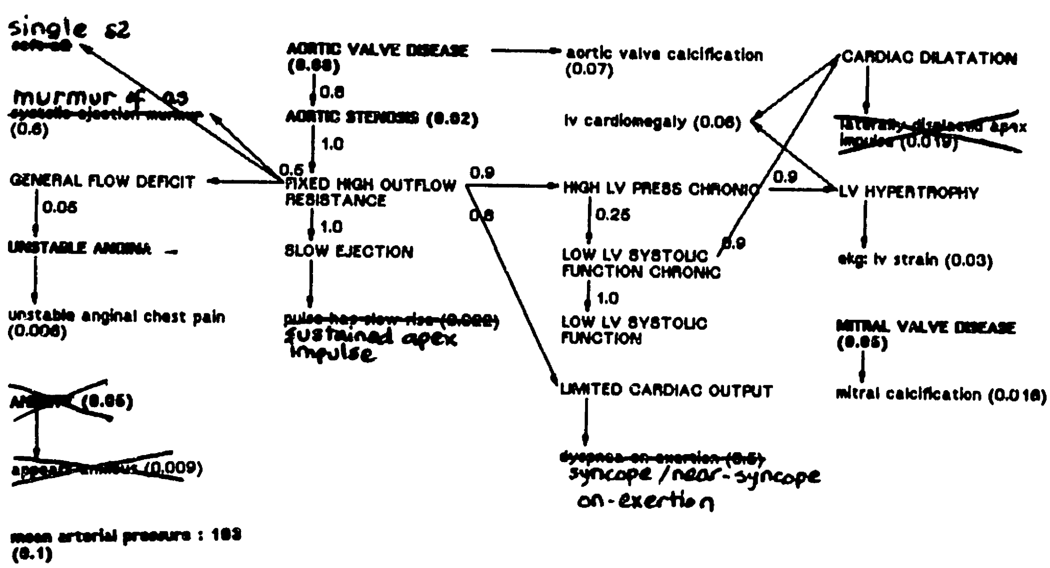
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Adam



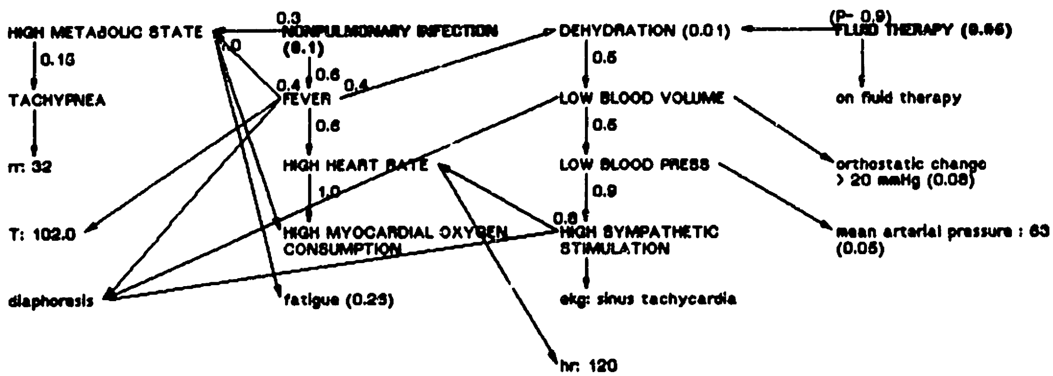
Patient: CAL
Heart Failure program's solution:



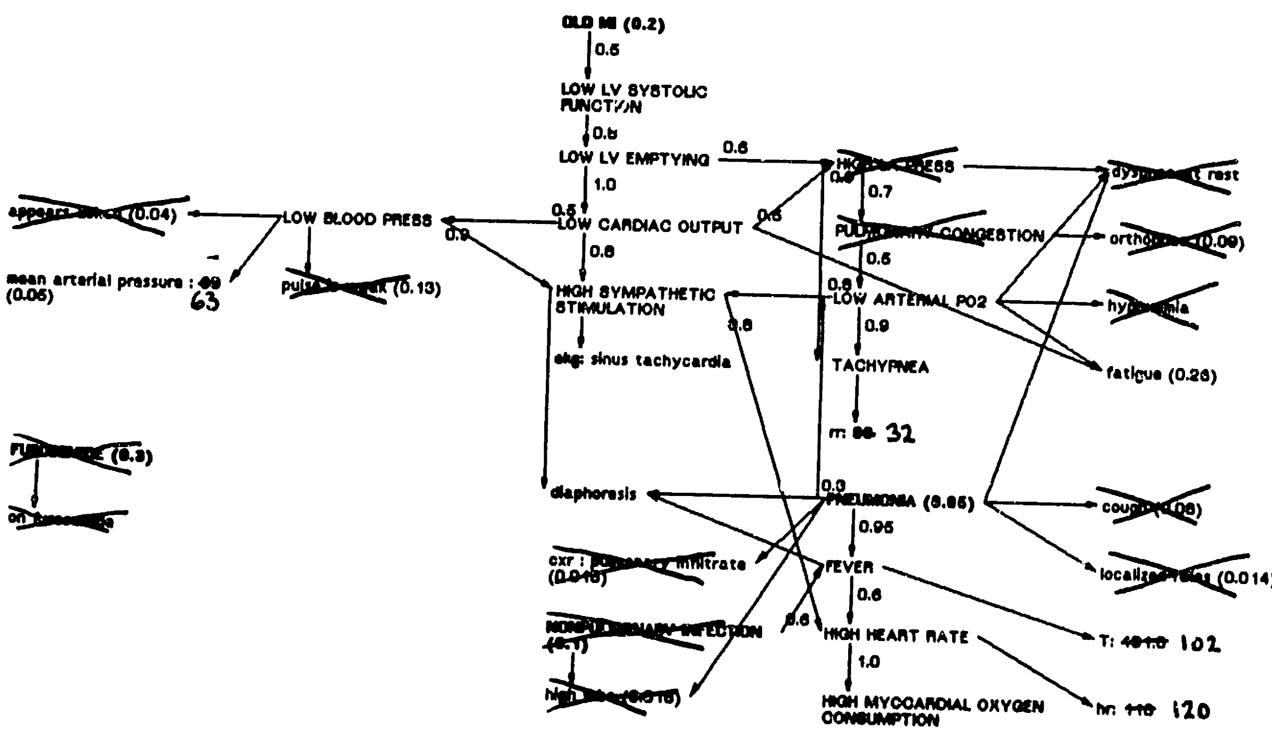
CASEY's result: satisfactory.
Unexplained features: none.
Tranferred from patient: Natalie
Note: Included one extra state, CARDIAC DILATATION.



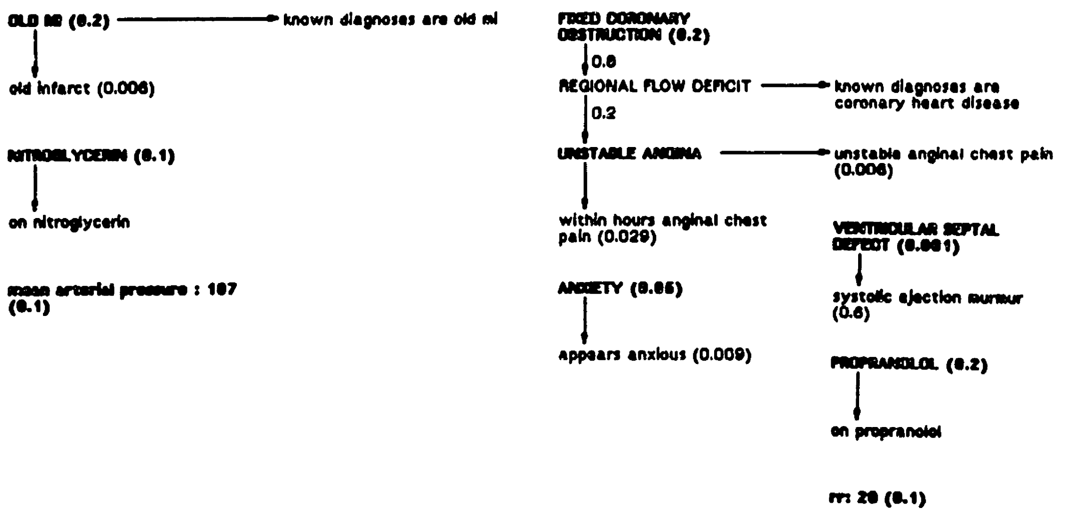
Patient: CHIP
Heart Failure program's solution:



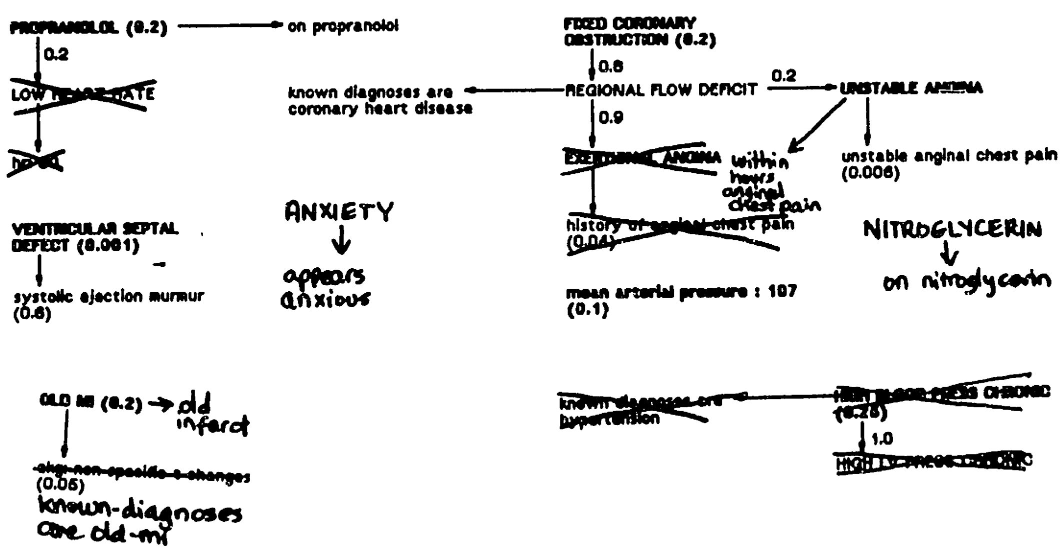
CASEY's result: wrong.
Tranferred from patient: Egbert
Note: Attributed patient's symptoms to pneumonia.



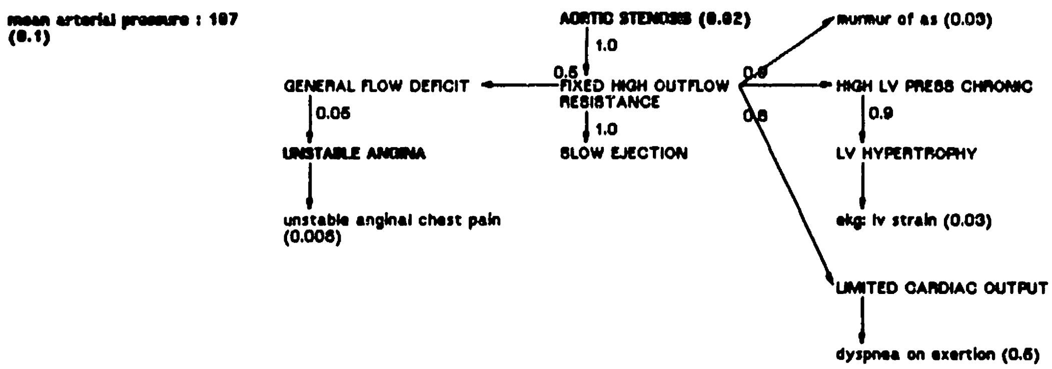
Patient: CODY
Heart Failure program's solution:



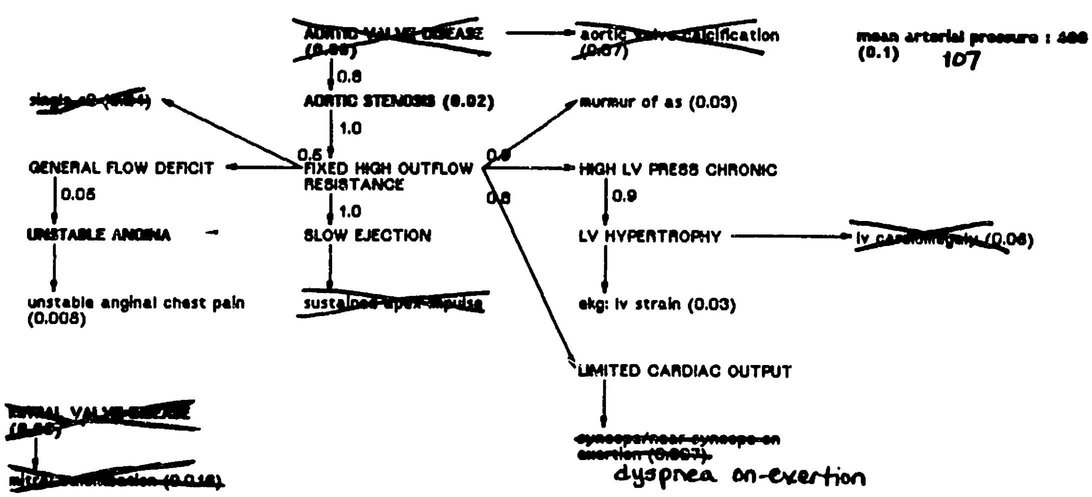
CASEY's result: satisfactory.
Unexplained features: systolic ejection murmur, mean arterial pressure: 107.
Tranferred from patient: Karl



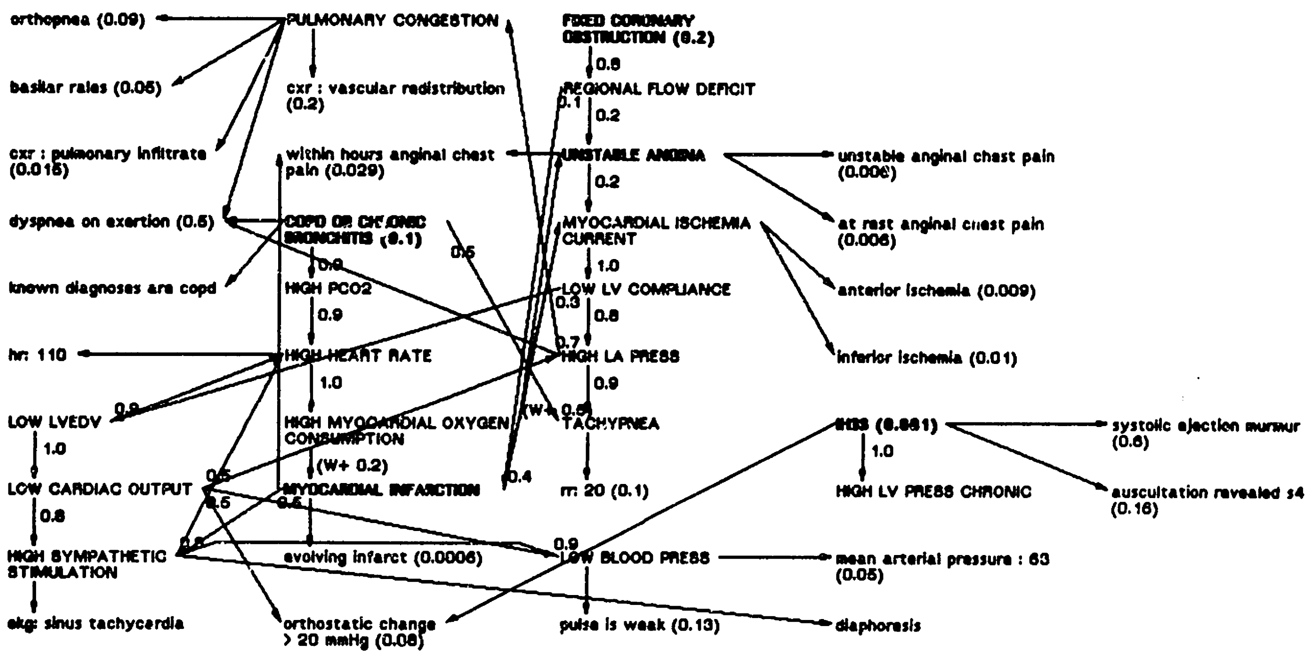
Patient: DAVID
Heart Failure program's solution:



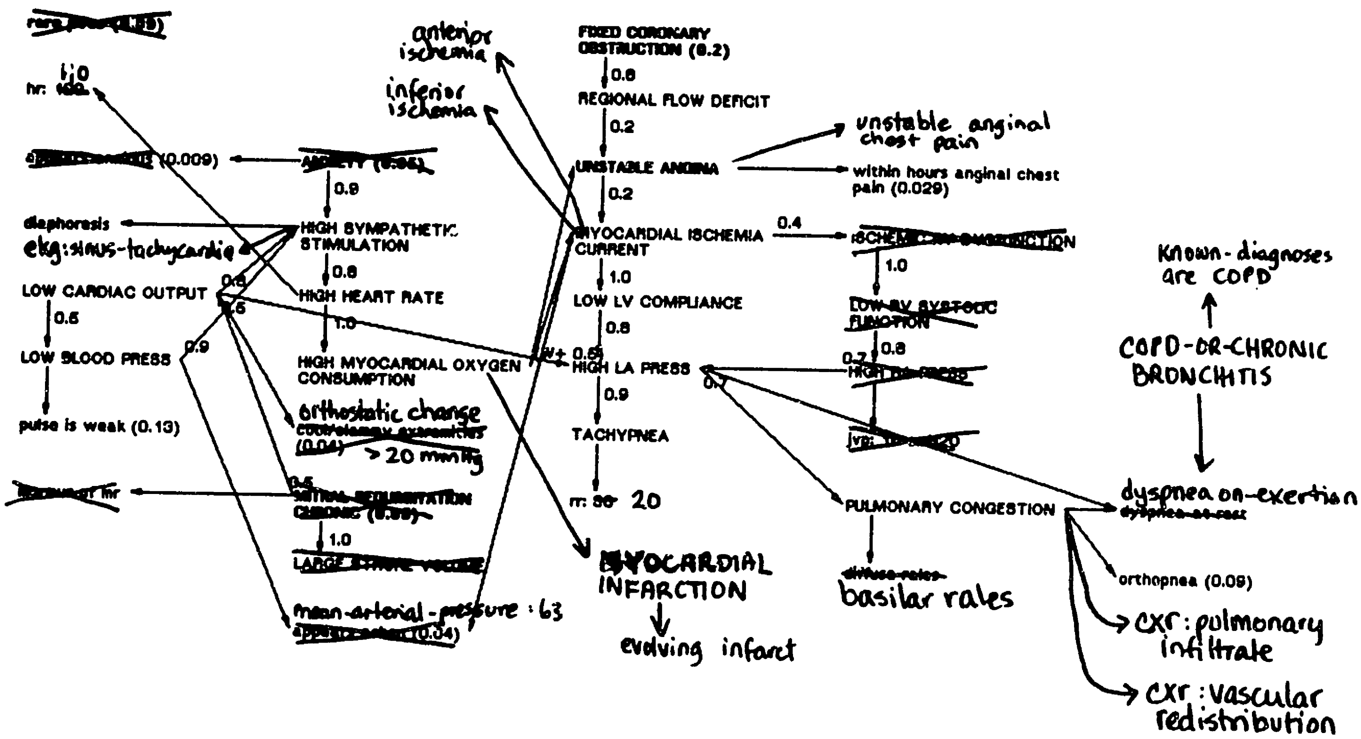
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Cal



Patient: DORIAN
Heart Failure program's solution:

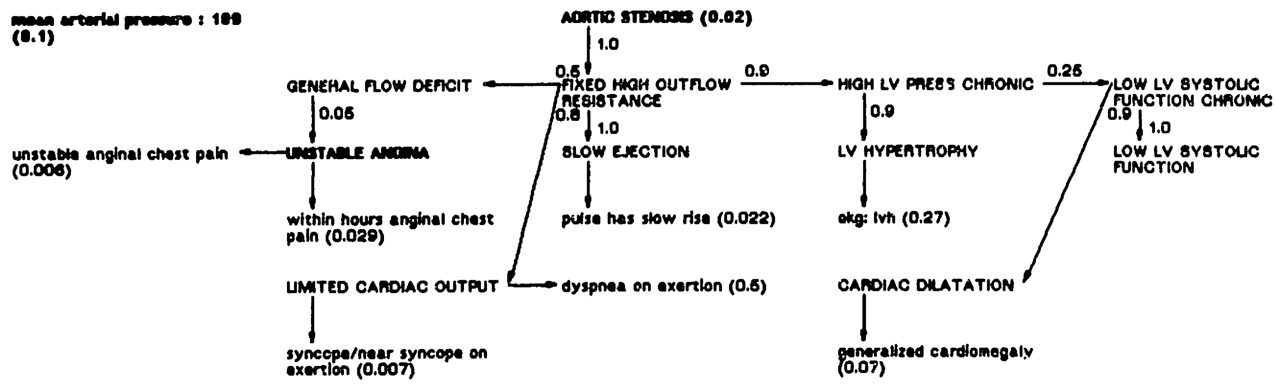


CASEY's result: satisfactory.
Unexplained features: systolic ejection murmur, s4.
Tranferred from patient: Sam

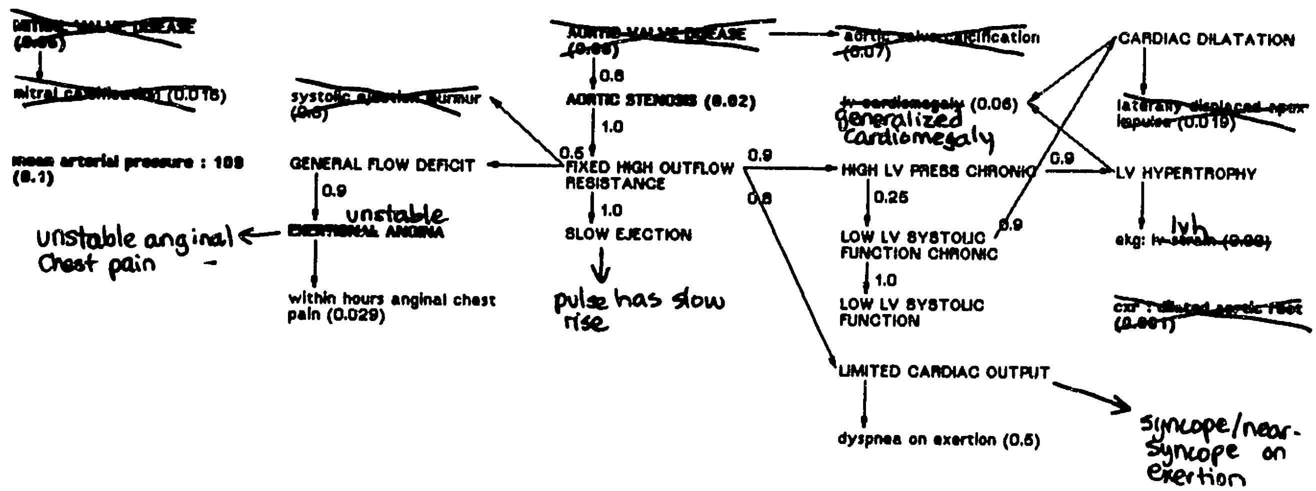


anterior ischemia
inferior ischemia
FIXED CORONARY OBSTRUCTION (0.2)
0.8
REGIONAL FLOW DEFICIT
0.2
UNSTABLE ANGINA
unstable anginal chest pain
within hours anginal chest pain (0.029)
0.2
MYOCARDIAL ISCHEMIA CURRENT
0.4
1.0
LOW LV COMPLIANCE
0.8
HIGH LA PRESS
0.9
TACHYPNEA
rr: 20
MYOCARDIAL INFARCTION
evolving infarct
hr: 110
(0.009)
0.9
diaphoresis
HIGH SYMPATHETIC STIMULATION
ekg: sinus-tachycardia
0.8
HIGH HEART RATE
LOW CARDIAC OUTPUT
0.5
LOW BLOOD PRESS
0.9
1.0
HIGH MYOCARDIAL OXYGEN CONSUMPTION
pulse is weak (0.13)
orthostatic change
>20 mmHg
0.5
1.0
mean-arterial-pressure: 63
PULMONARY CONGESTION
basilar rales
0.8
0.7
dyspnea on-exertion
orthopnea (0.09)
cxr: pulmonary infiltrate
cxr: vascular redistribution
Known-diagnoses are COPD
COPD-OR-CHRONIC BRONCHITIS

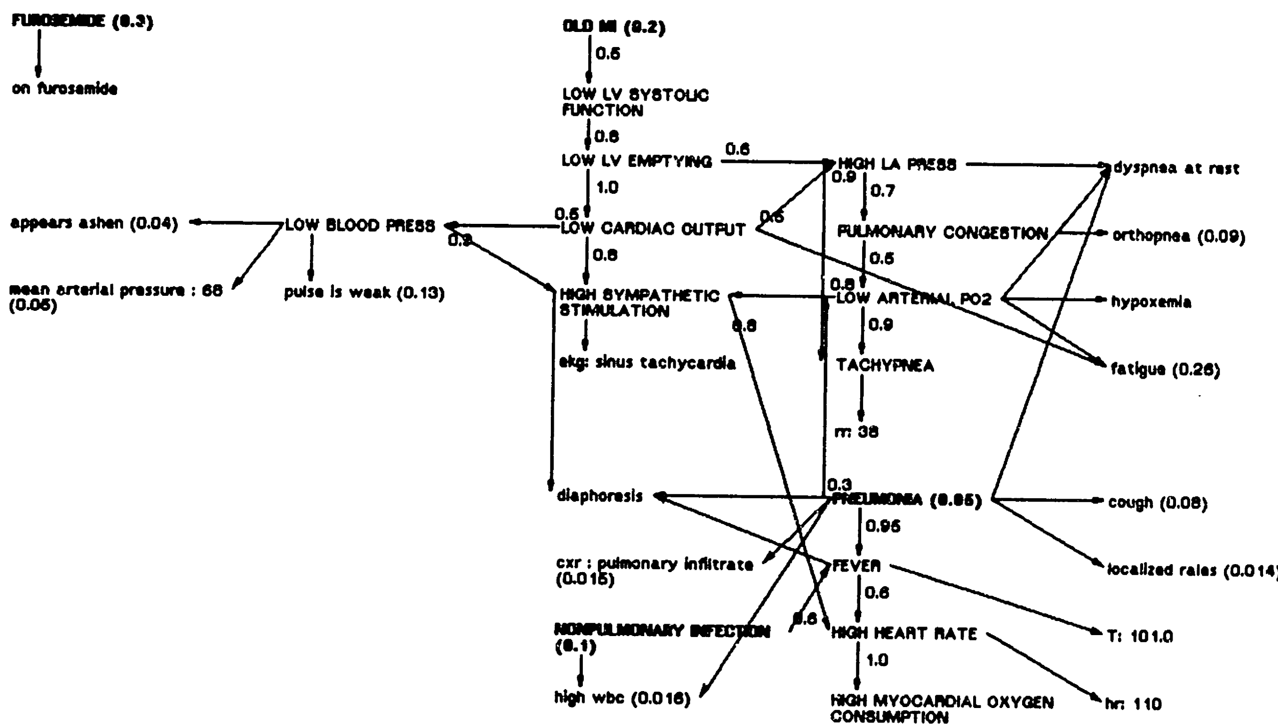
Patient: EDITH
Heart Failure program's solution:



CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Jethro

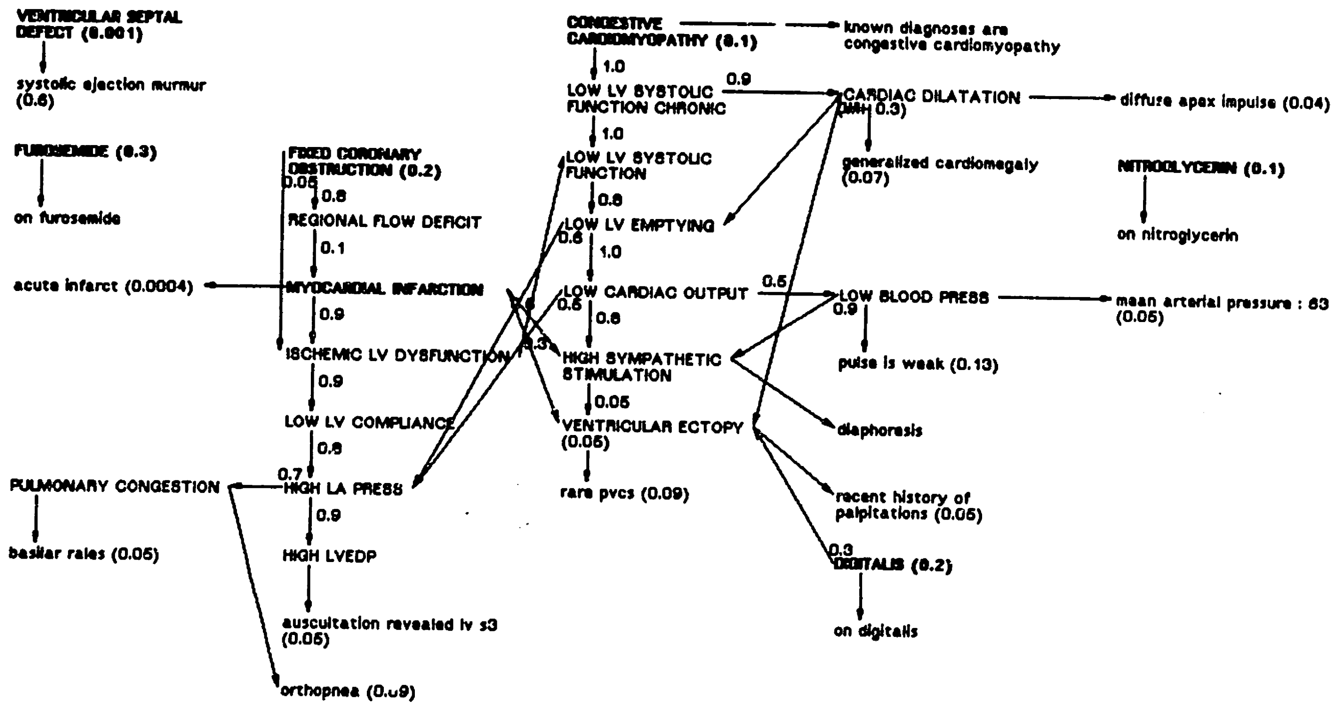


Patient: EGBERT
Heart Failure program's solution:

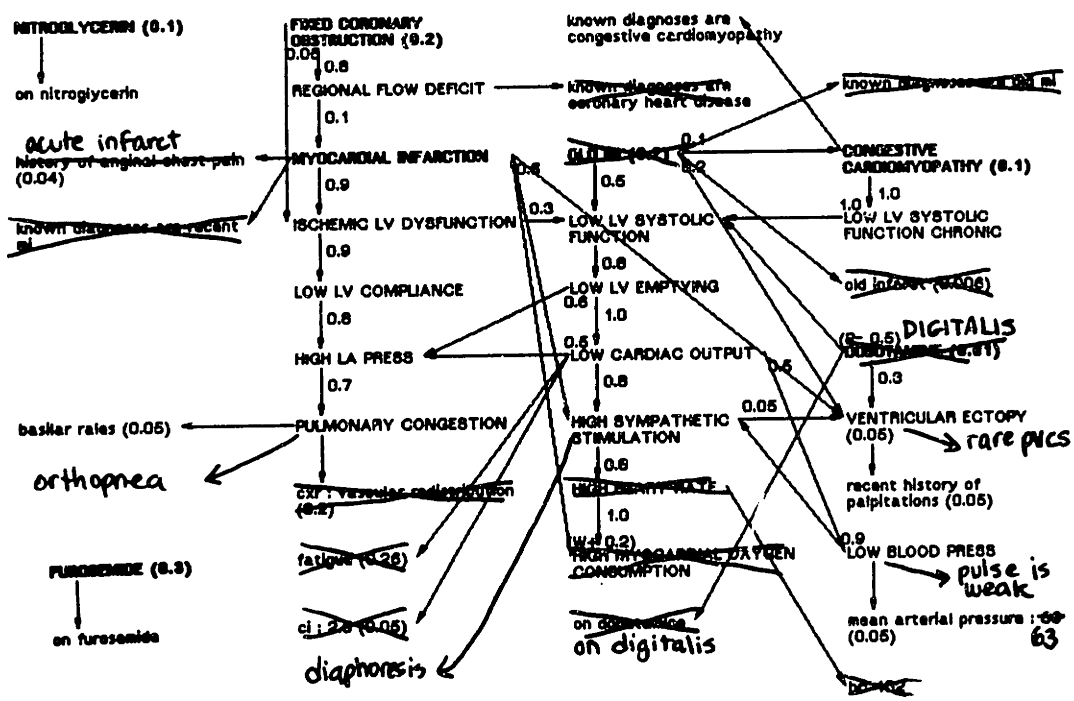


CASEY's result: gives up.
Note: CASEY is reminded of several cases with pneumonia, nonpulmonary infection, or old-mi, but it cannot account for all of this patient's findings with any precedent.

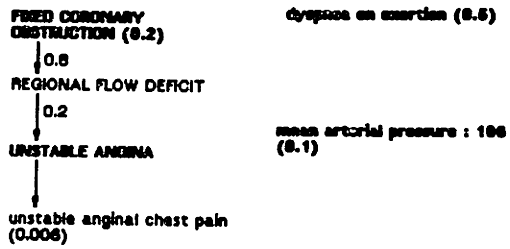
Patient: FARLEY
Heart Failure program's solution:



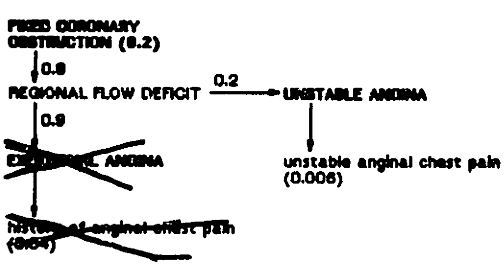
CASEY's result: satisfactory.
Unexplained features: systolic ejection murmur, lv-s3, diffuse apex impulse, generalized cardiomegaly.
Tranferred from patient: Mac


NITROGLYCERIN (0.1)
on nitroglycerin
FUROSEMIDE (0.3)
on furosemide
acute infarct
history of anginal chest pain (0.04)
known diagnoses are recent mi
basilar rales (0.05)
orthopnea
FIXED CORONARY OBSTRUCTION (0.2)
0.05
0.8
REGIONAL FLOW DEFICIT
0.1
MYOCARDIAL INFARCTION
0.9
ISCHEMIC LV DYSFUNCTION
0.9
LOW LV COMPLIANCE
0.8
HIGH LA PRESS
0.7
PULMONARY CONGESTION
cxr : vascular redistribution (0.2)
fatigue (0.26)
ci : 2.0 (0.05)
diaphoresis
known diagnoses are congestive cardiomyopathy
known diagnoses are coronary heart disease
OLD MI
0.1
0.2
0.5
0.3
LOW LV SYSTOLIC FUNCTION
0.8
LOW LV EMPTYING
0.6
1.0
0.5
LOW CARDIAC OUTPUT
0.8
HIGH SYMPATHETIC STIMULATION
0.8
HIGH HEART RATE
1.0
HIGH MYOCARDIAL OXYGEN CONSUMPTION
on digitalis
0.05
0.5
known diagnoses are old mi
CONGESTIVE CARDIOMYOPATHY (0.1)
1.0
LOW LV SYSTOLIC FUNCTION CHRONIC
old infarct (0.005)
DIGITALIS
0.3
VENTRICULAR ECTOPY (0.05)
rare pvcs
recent history of palpitations (0.05)
0.9
LOW BLOOD PRESS
pulse is weak
mean arterial pressure : 60 (0.05)
63
hr : 102

Patient: FRANCIS
Heart Failure program's solution:

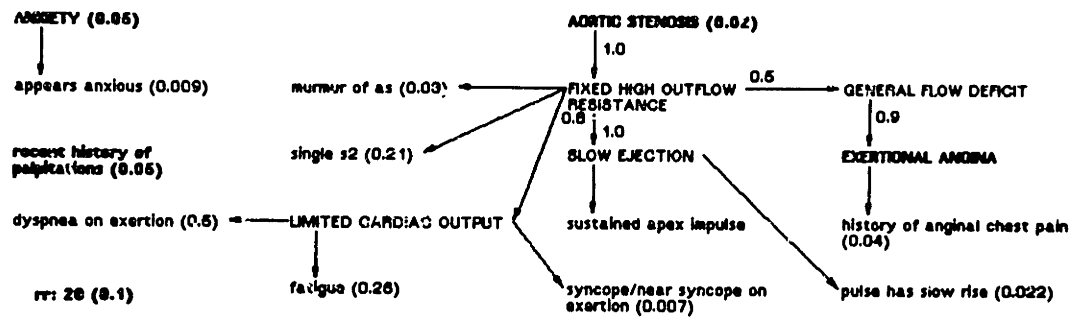


CASEY's result: identical.
Tranferred from patient: Sarah.

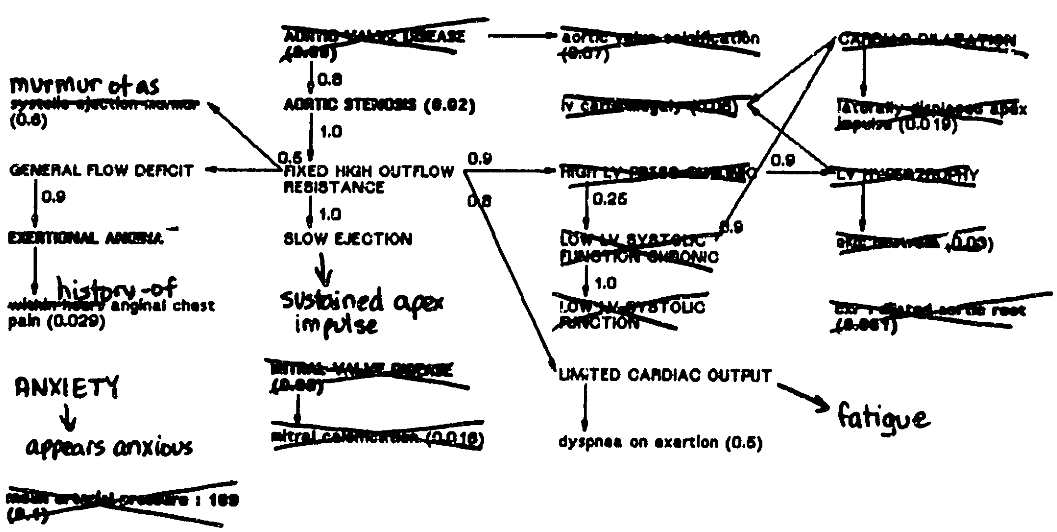


dyspnea on exertion
mean-arterial-pressure : 106

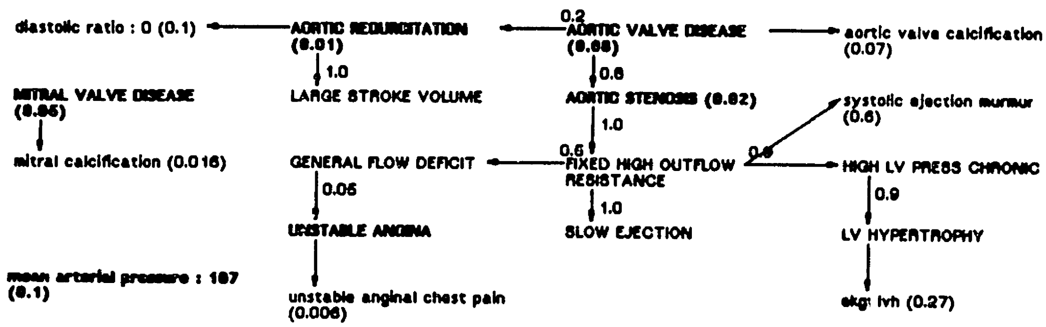
Patient: FRANK
Heart Failure program's solution:



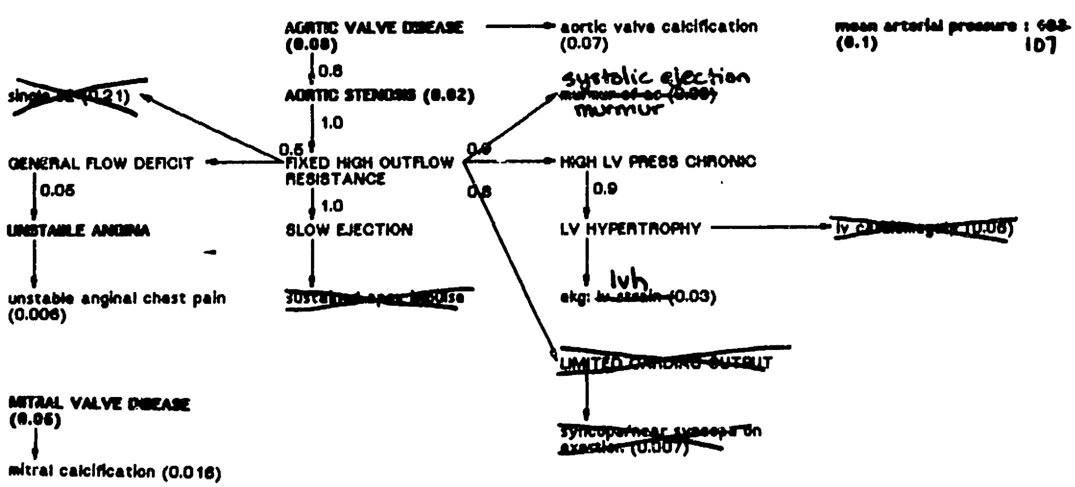
CASEY's result: identical.
Unexplained features: recent history of palpitations.
Tranferred from patient: Jethro



Patient: GERTRUDE
Heart Failure program's solution:

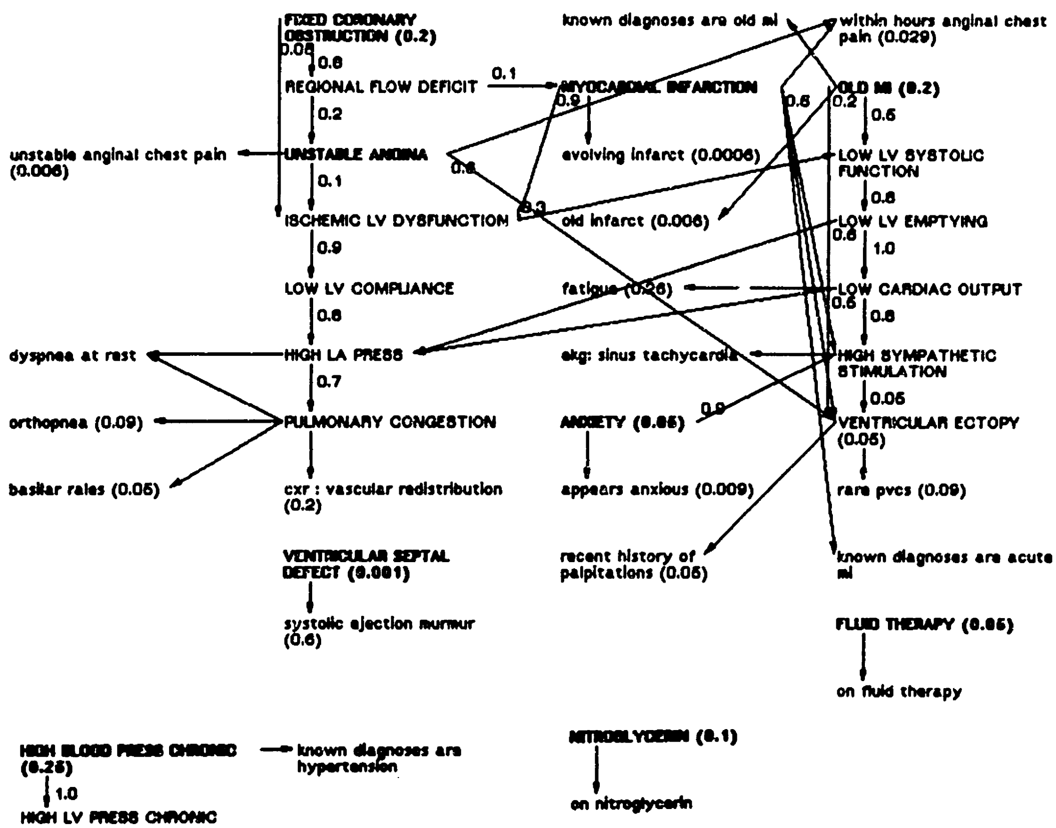


CASEY's result: satisfactory.
Unexplained features: diastolic ratio: 0, heart rate: 88.
Tranferred from patient: Cal



Patient: HAROLD

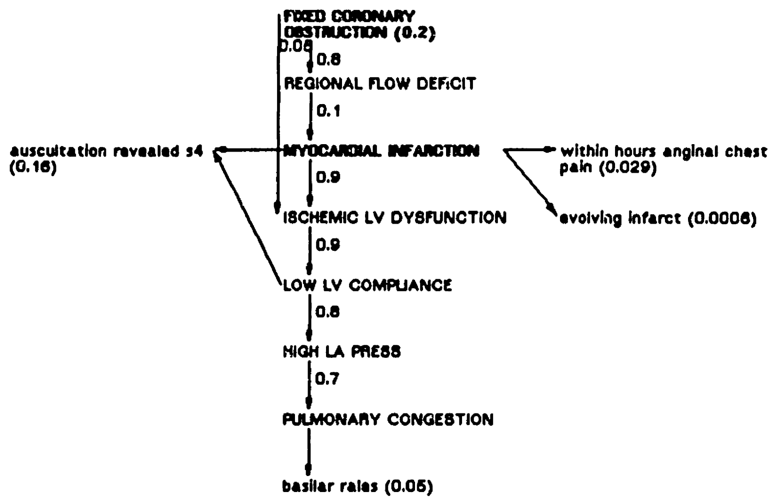
Heart Failure program's solution:



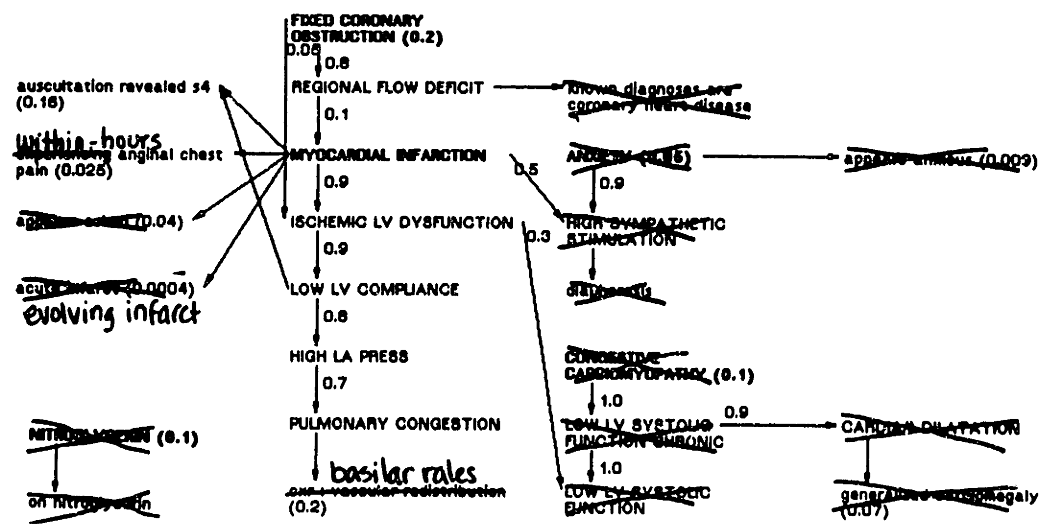
CASEY's result: gave up.

Note: CASEY recalled several precedents each of which could account for some of Harold's findings, but no one precedent covered all the diagnoses.

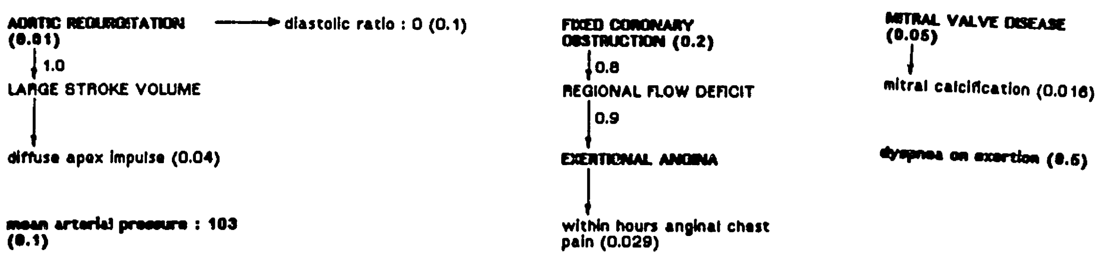
Patient: HEYWOOD
Heart Failure program's solution:



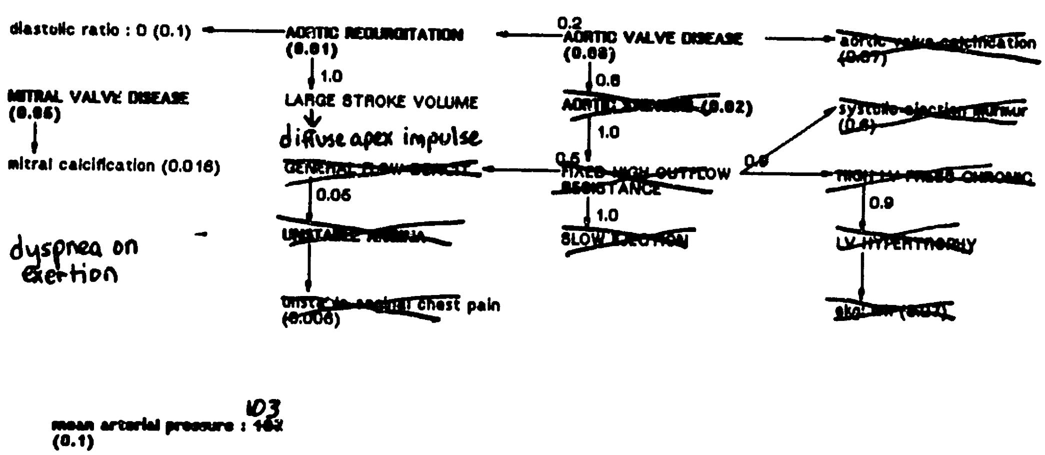
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Saladin



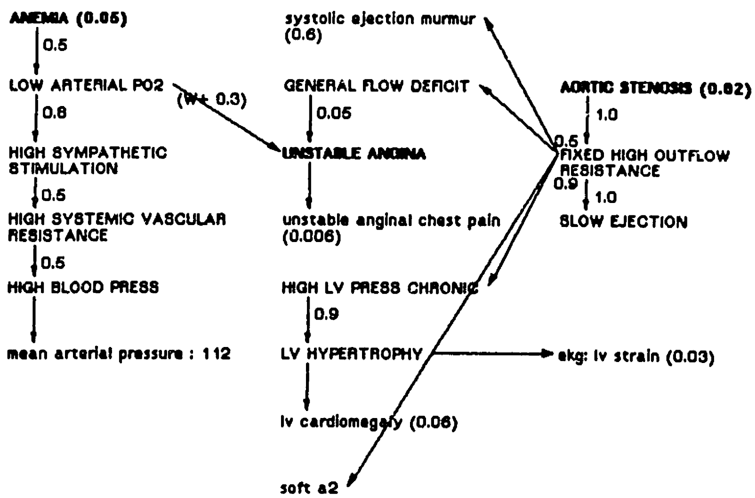
Patient: HORATIO
Heart Failure program's solution:



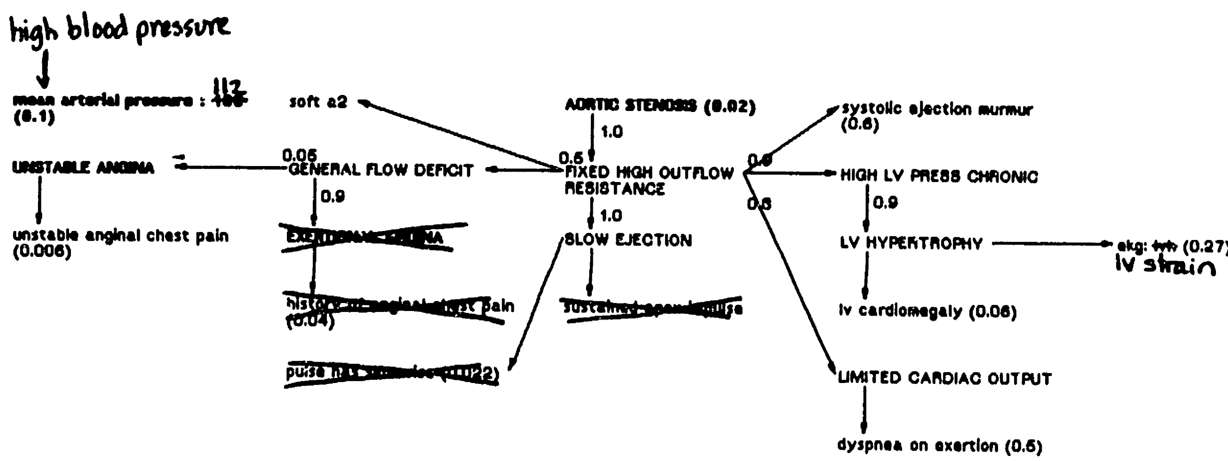
CASEY's result: satisfactory.
Unexplained features: anginal chest pain within hours.
Tranferred from patient: Gertrude



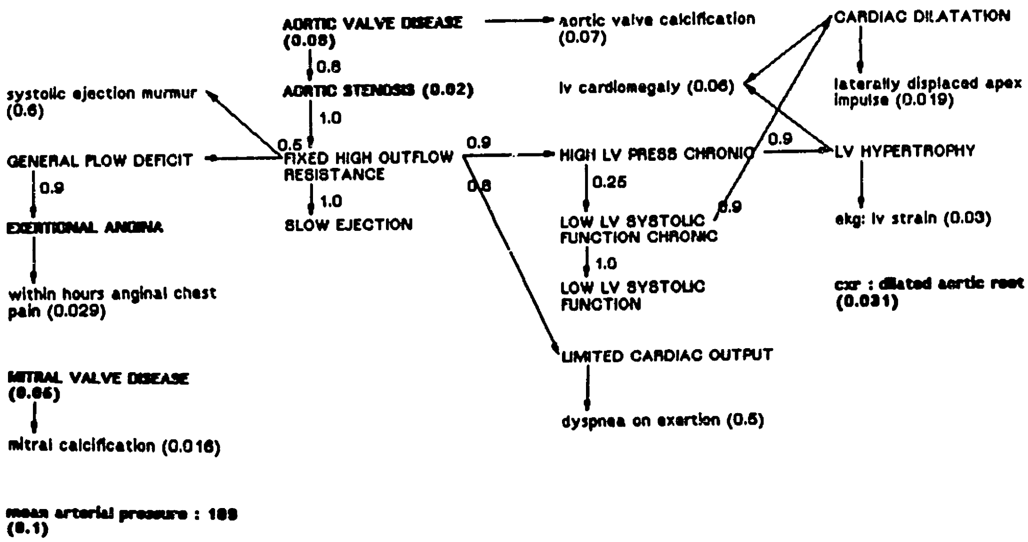
Patient: INDIRA
Heart Failure program's solution:



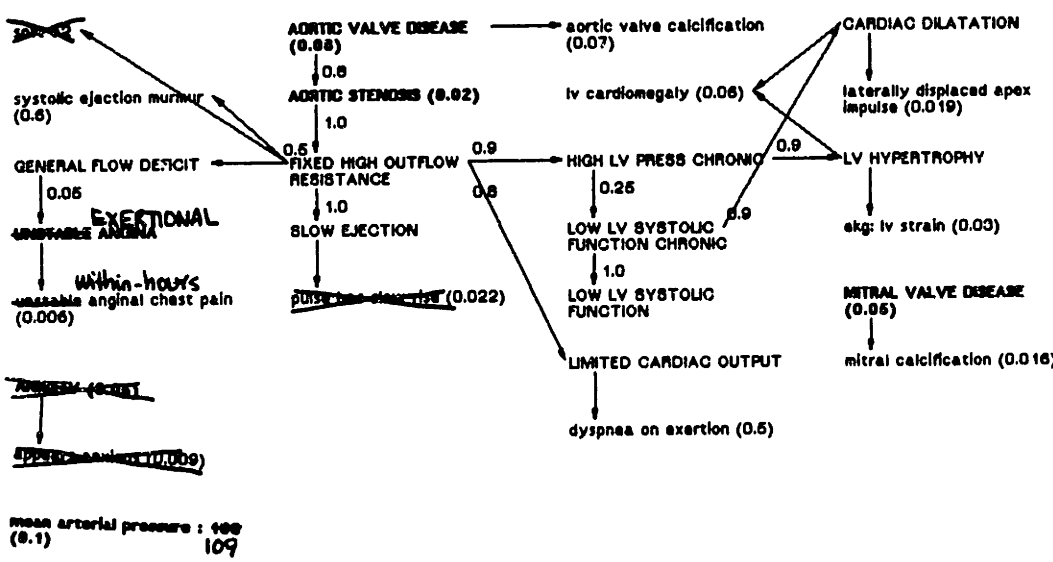
CASEY's result: satisfactory.
Unexplained features: high blood pressure.
Tranferred from patient: Oprah



Patient: JETHRO
Heart Failure program's solution:

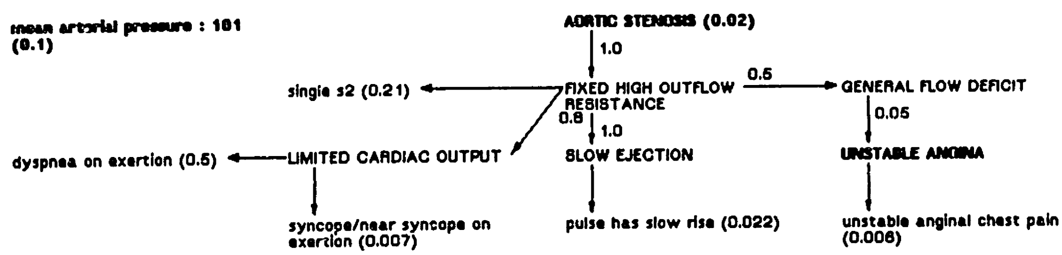


CASEY's result: identical.
Unexplained features: cxr: dilated aortic root.
Tranferred from patient: Natalie



Patient: KALMAN

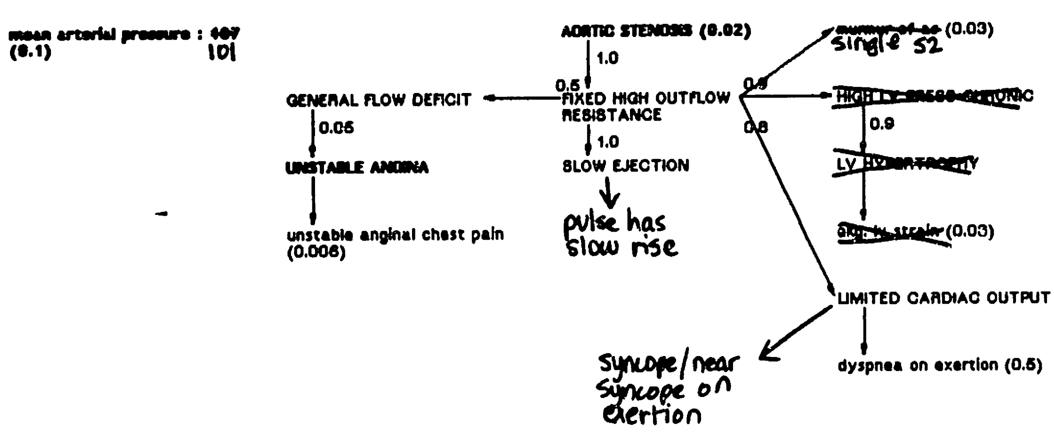
Heart Failure program's solution:



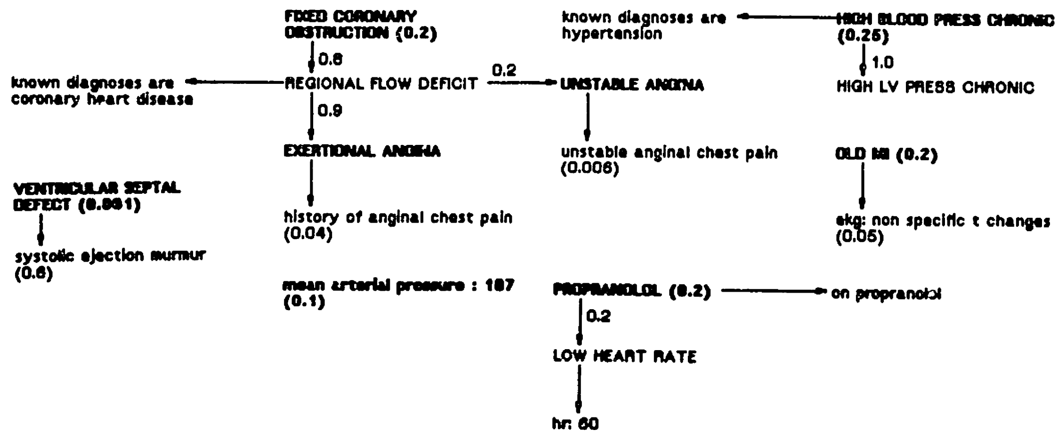
CASEY's result: identical.

Unexplained features: none.

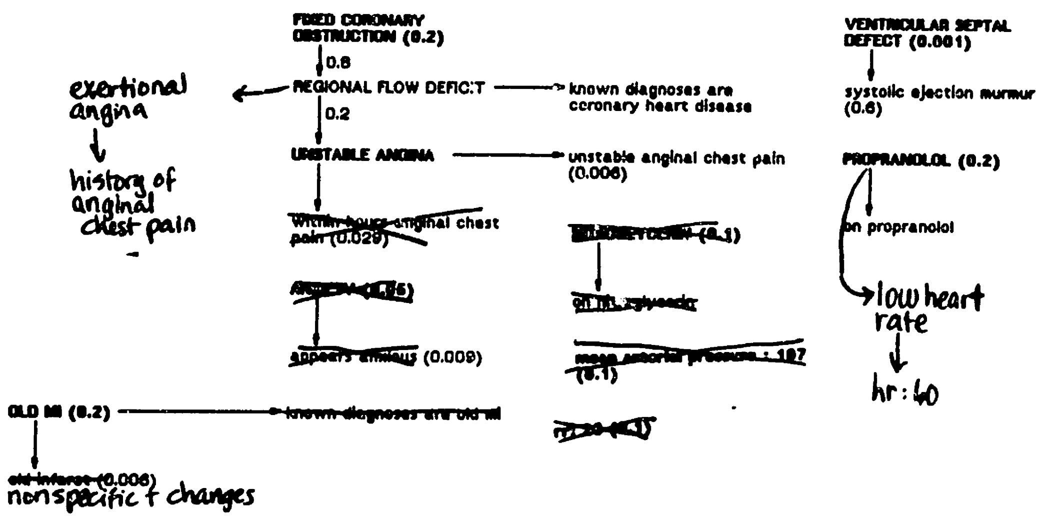
Tranferred from patient: David



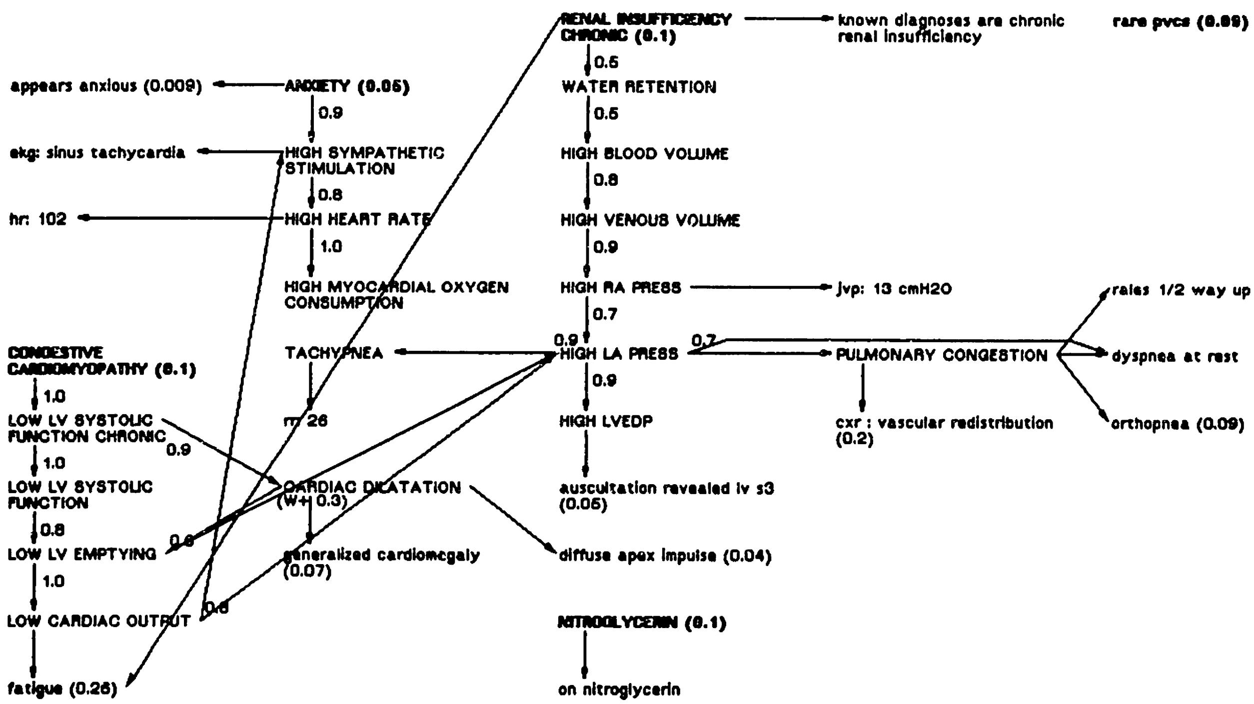
Patient: KARL
Heart Failure program's solution:



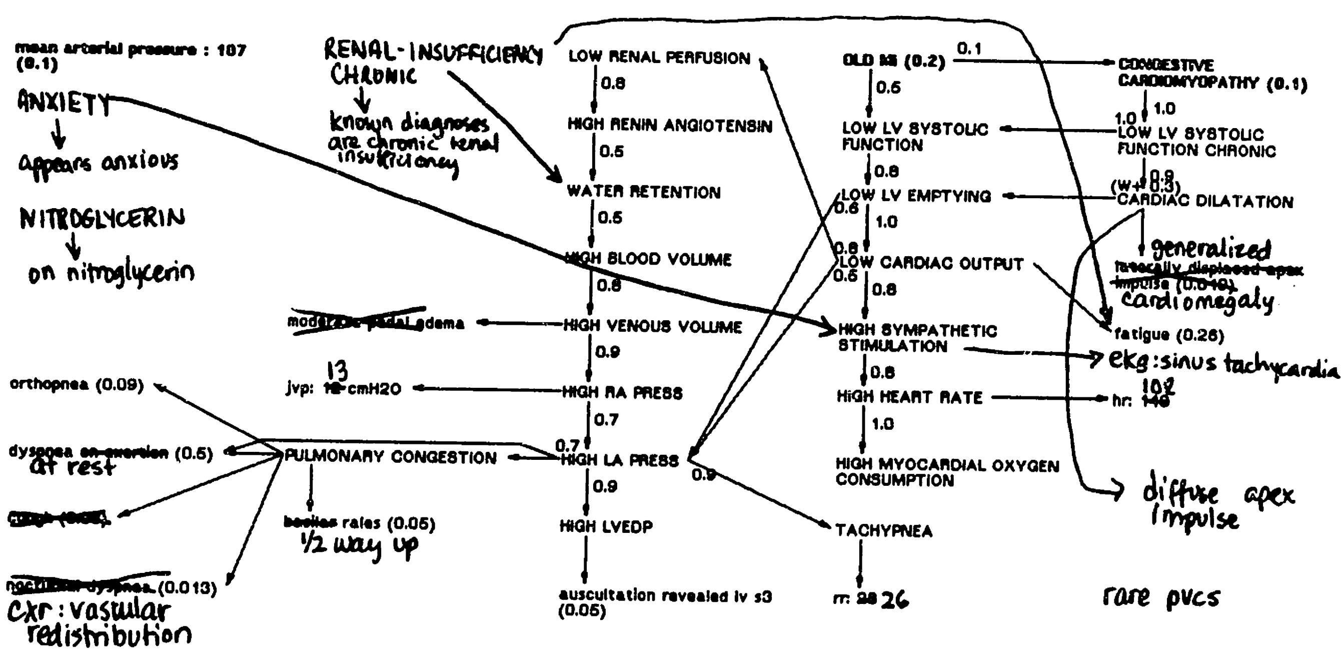
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Cody



Patient: KYLE
Heart Failure program's solution:



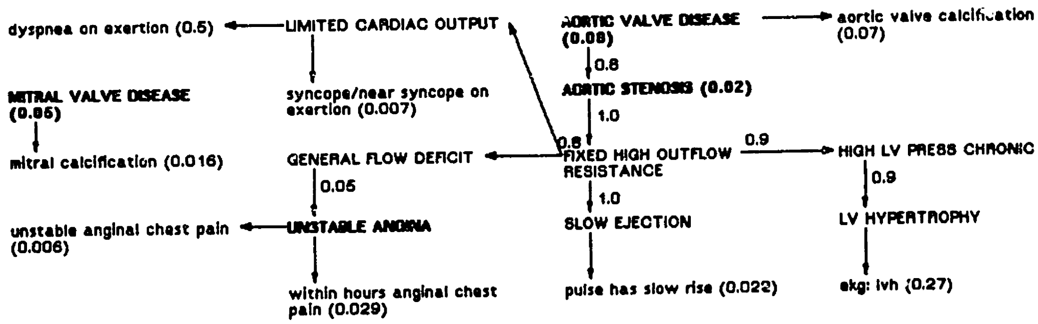
CASEY's result: wrong.
Tranferred from patient: Peter
Note: CASEY got most of this causal explanation right, but incorrectly attributed the patient's renal problems to heart failure due to an old myocardial infarction.



mean arterial pressure : 107 (0.1)
ANXIETY
appears anxious
NITROGLYCERIN
on nitroglycerin
RENAL-INSUFFICIENCY CHRONIC
known diagnoses are chronic renal insufficiency
LOW RENAL PERFUSION
0.8
HIGH RENIN ANGIOTENSIN
0.5
WATER RETENTION
0.5
HIGH BLOOD VOLUME
0.8
HIGH VENOUS VOLUME
0.9
HIGH RA PRESS
0.7
HIGH LA PRESS
0.9
HIGH LVEDP
auscultation revealed lv s3 (0.05)
moderate pedal edema
jvp: 13 cmH2O
orthopnea (0.09)
dyspnea at rest (0.5)
PULMONARY CONGESTION
cough
rales (0.05)
1/2 way up
(0.013)
cxr: vascular redistribution
OLD MI (0.2)
0.1
0.5
LOW LV SYSTOLIC FUNCTION
0.8
LOW LV EMPTYING
1.0
LOW CARDIAC OUTPUT
0.8
HIGH SYMPATHETIC STIMULATION
0.8
HIGH HEART RATE
1.0
HIGH MYOCARDIAL OXYGEN CONSUMPTION
TACHYPNEA
rr: 26
CONGESTIVE CARDIOMYOPATHY (0.1)
1.0
LOW LV SYSTOLIC FUNCTION CHRONIC
CARDIAC DILATATION
generalized
cardiomegaly
fatigue (0.26)
ekg: sinus tachycardia
hr: 102
diffuse apex impulse
rare pvcs

Patient: LARRY

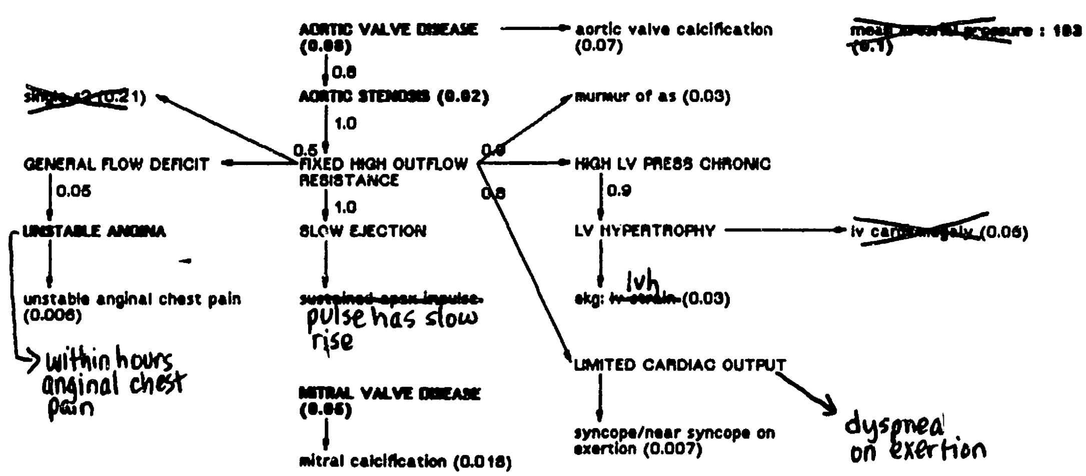
Heart Failure program's solution:



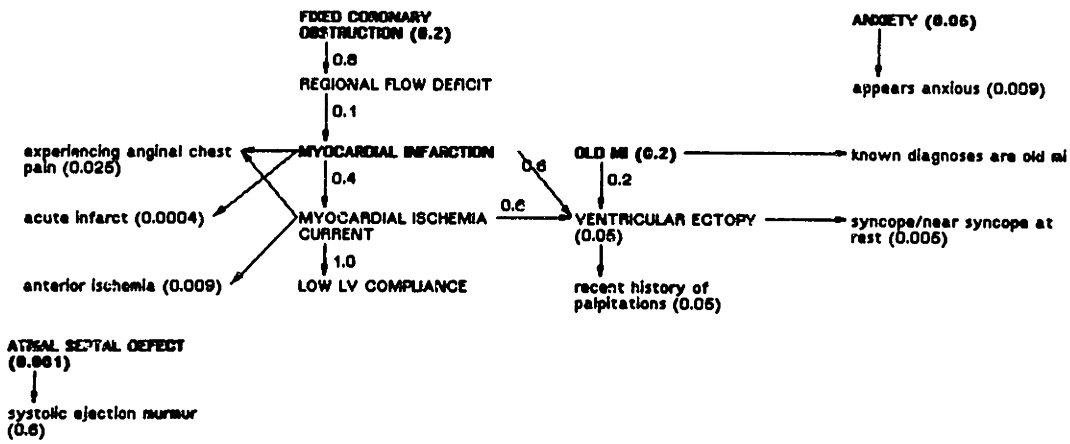
CASEY's result: identical.

Unexplained features: none.

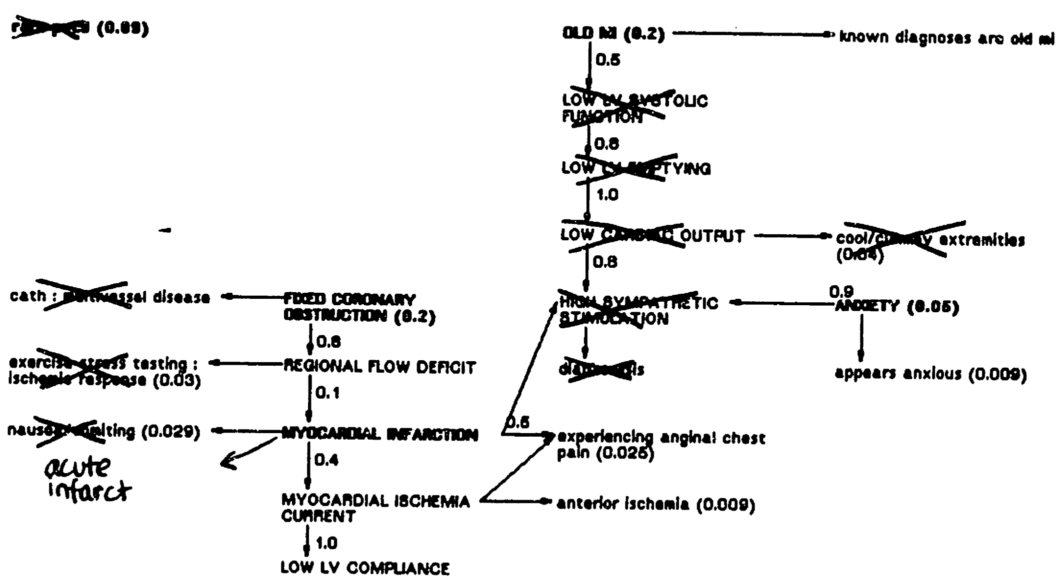
Tranferred from patient: Cal



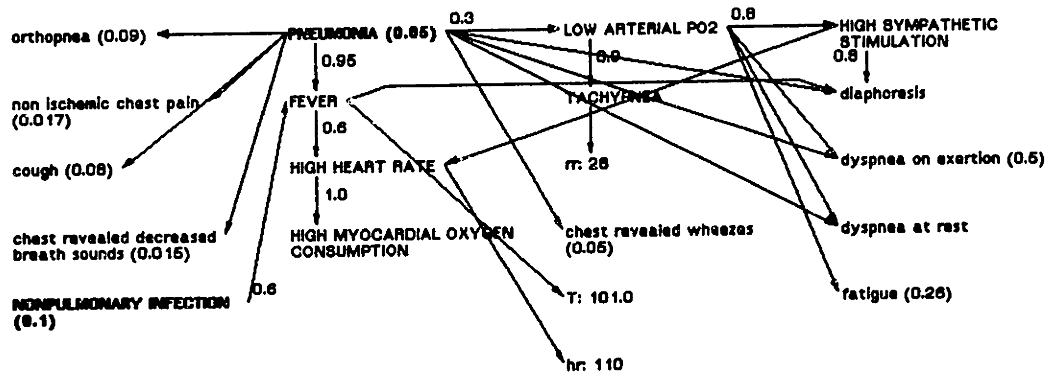
Patient: LEN
Heart Failure program's solution:



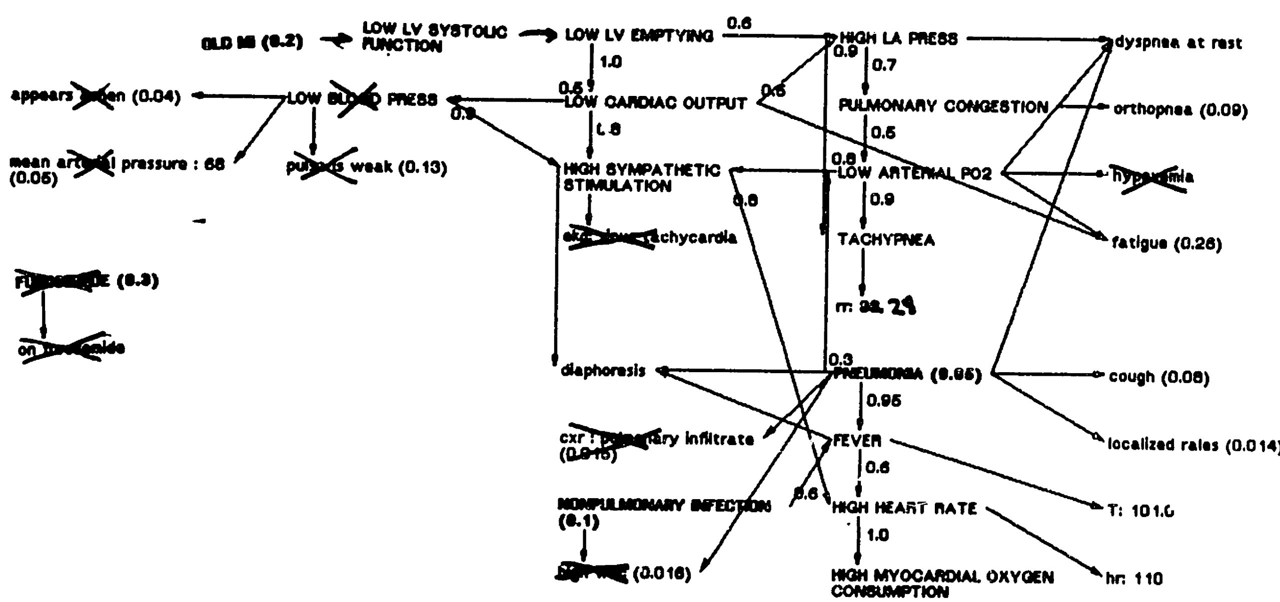
CASEY's result: satisfactory.
Unexplained features: syncope/near syncope at rest, systolic ejection murmur, recent history of palpitations.
Tranferred from patient: Umberto



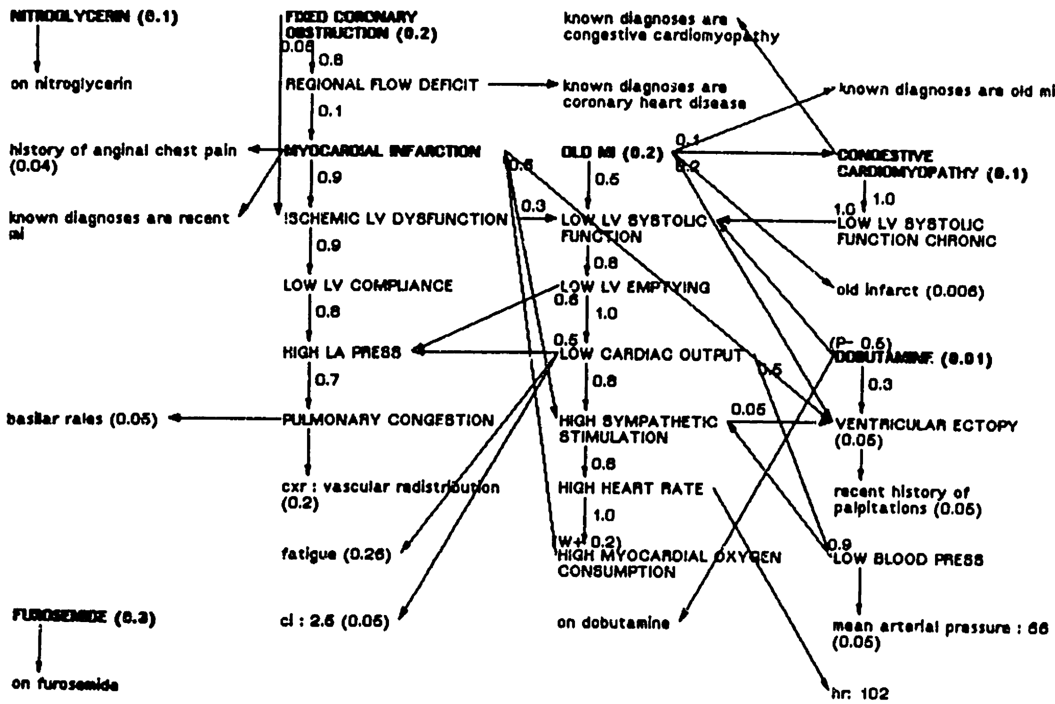
Patient: MARY
Heart Failure program's solution:



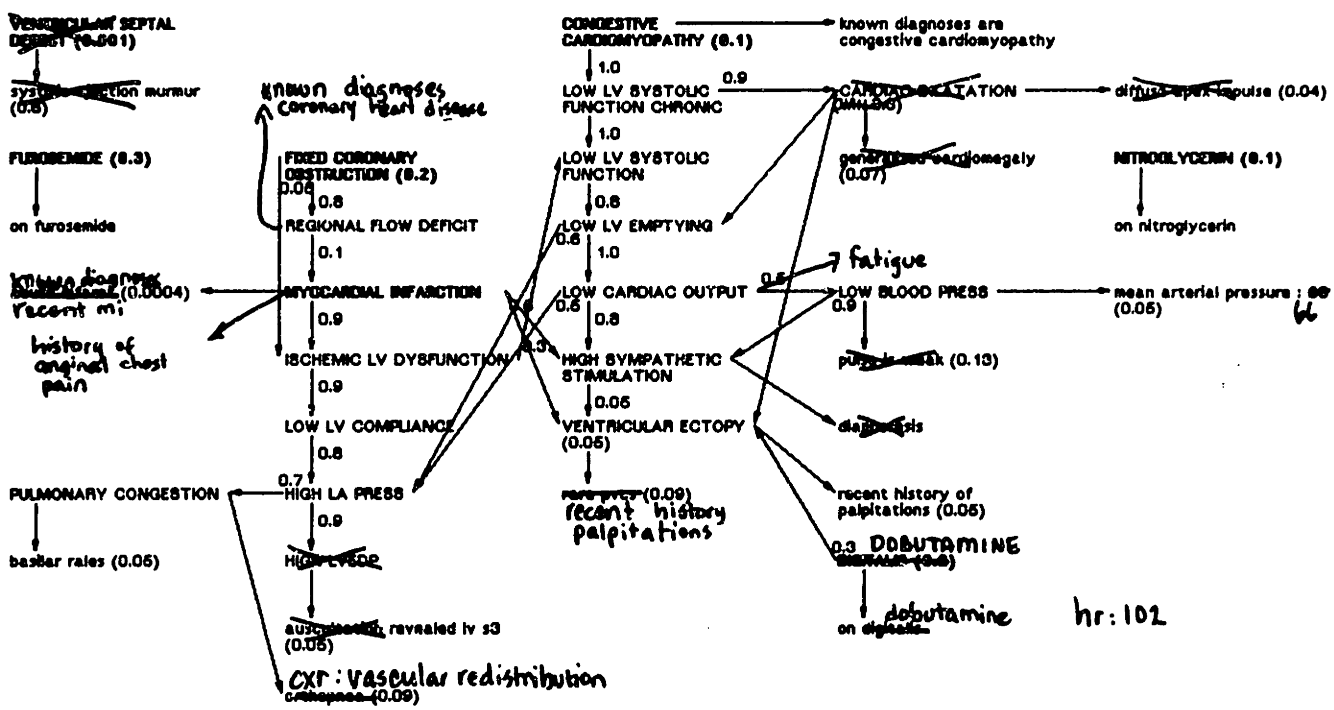
CASEY's result: wrong.
Unexplained features: none.
Tranferred from patient: Egbert
Note: reminded of Egbert, another patient with pneumonia and nonpulmonary infection.



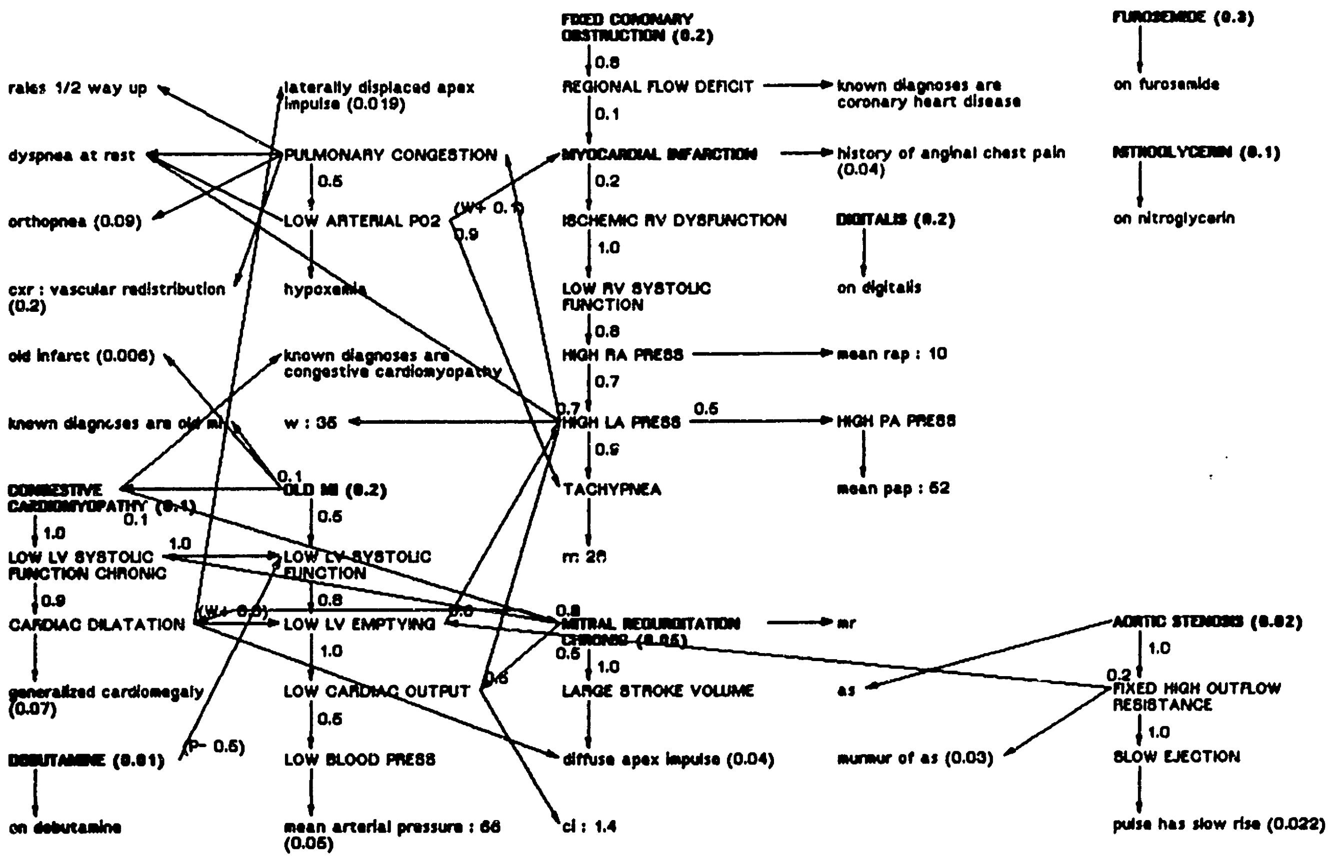
Patient: MAC
Heart Failure program's solution:



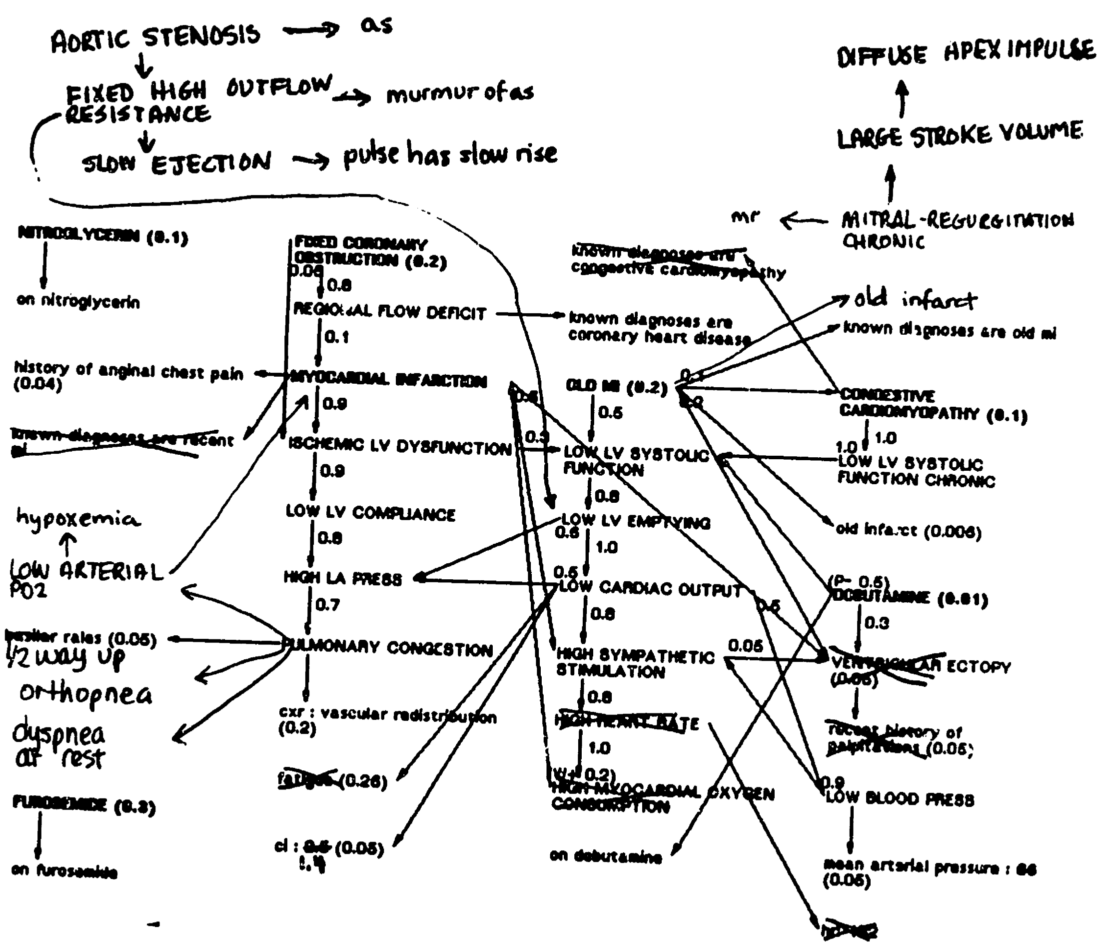
CASEY's result: satisfactory.
Unexplained features: heart rate: 102.
Tranferred from patient: Farley


CONGESTIVE CARDIOMYOPATHY (0.1)
known diagnoses are congestive cardiomyopathy
LOW LV SYSTOLIC FUNCTION CHRONIC
LOW LV SYSTOLIC FUNCTION
LOW LV EMPTYING
LOW CARDIAC OUTPUT
HIGH SYMPATHETIC STIMULATION
VENTRICULAR ECTOPY (0.05)
FUROSEMIDE (0.3)
on furosemide
FIXED CORONARY OBSTRUCTION (0.2)
REGIONAL FLOW DEFICIT
MYOCARDIAL INFARCTION
ISCHEMIC LV DYSFUNCTION
LOW LV COMPLIANCE
HIGH LA PRESS
PULMONARY CONGESTION
basilar rales (0.05)
LOW BLOOD PRESS
mean arterial pressure
NITROGLYCERIN (0.1)
on nitroglycerin
recent history of palpitations (0.05)
known diagnoses coronary heart disease
known diagnoses recent mi
history of anginal chest pain
fatigue
recent history palpitations
DOBUTAMINE
dobutamine
hr: 102
cxr: vascular redistribution

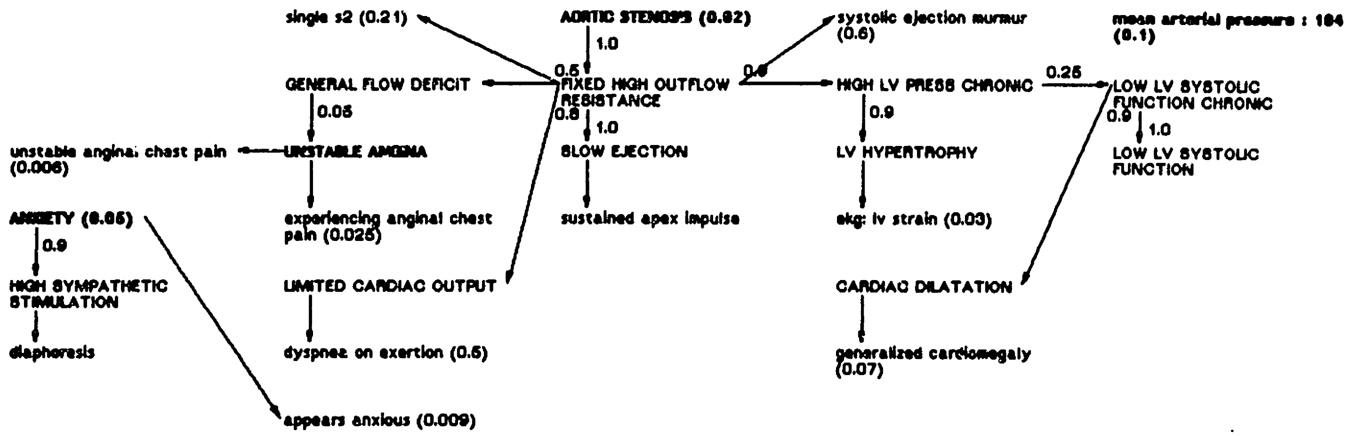
Patient: MELANIE
Heart Failure program's solution:



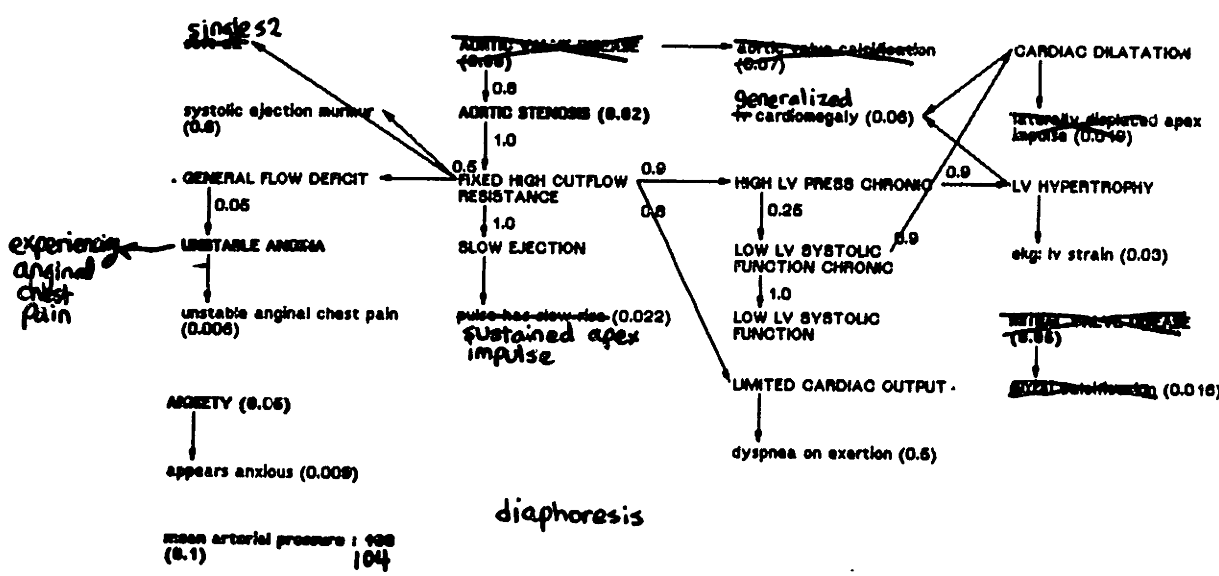
CASEY's result: wrong.
Note: CASEY gets much of this causal explanation correct, but is attempting to transfer it from "Mac," and so it mistakenly uses ISCHEMIC LV DYSFUNCTION instead of ISCHEMIC RV DYSFUNCTION.


AORTIC STENOSIS → as
FIXED HIGH OUTFLOW RESISTANCE → murmur of as
SLOW EJECTION → pulse has slow rise
DIFFUSE APEX IMPULSE
LARGE STROKE VOLUME
mr ← MITRAL-REGURGITATION CHRONIC
NITROGLYCERIN (0.1)
on nitroglycerin
FIXED CORONARY OBSTRUCTION (0.2)
REGIONAL FLOW DEFICIT
known diagnoses are coronary heart disease
known diagnoses are congestive cardiomyopathy
old infarct
known diagnoses are old mi
history of anginal chest pain (0.04)
MYOCARDIAL INFARCTION
OLD MI (0.2)
CONGESTIVE CARDIOMYOPATHY (0.1)
known diagnoses are recent mi
ISCHEMIC LV DYSFUNCTION
LOW LV SYSTOLIC FUNCTION
LOW LV SYSTOLIC FUNCTION CHRONIC
hypoxemia
LOW ARTERIAL PO2
LOW LV COMPLIANCE
LOW LV EMPTYING
old infarct (0.006)
HIGH LA PRESS
LOW CARDIAC OUTPUT
DOBUTAMINE (0.01)
basilar rales (0.05)
1/2 way up
orthopnea
dyspnea at rest
PULMONARY CONGESTION
HIGH SYMPATHETIC STIMULATION
VENTRICULAR ECTOPY (0.05)
cxr : vascular redistribution (0.2)
HIGH HEART RATE
recent history of palpitations (0.05)
fatigue (0.26)
HIGH MYOCARDIAL OXYGEN CONSUMPTION
LOW BLOOD PRESS
FUROSEMIDE (0.3)
on furosemide
ci : 0.5 (0.05)
on dobutamine
mean arterial pressure : 66 (0.05)

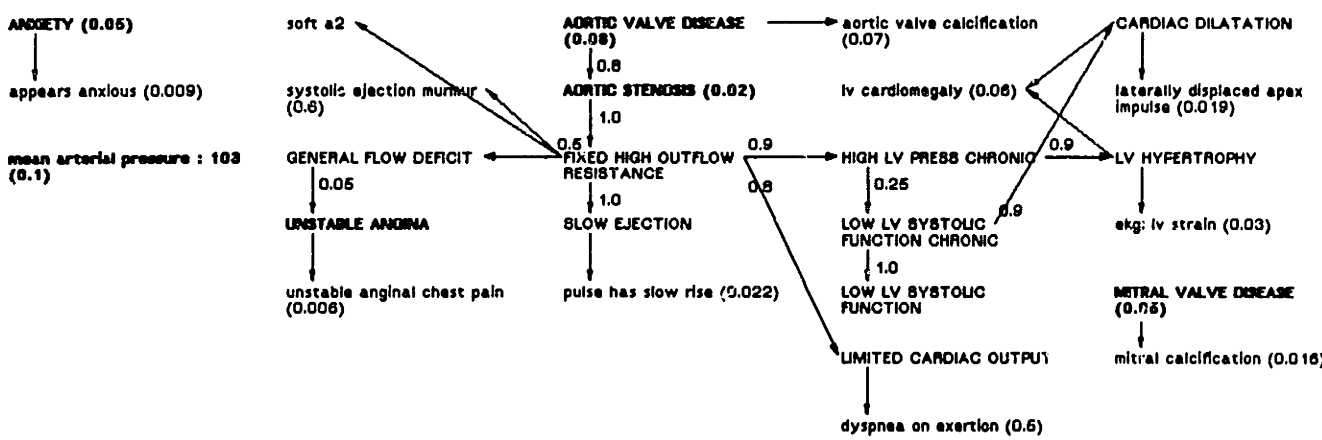
Patient: MARGARET
Heart Failure program's solution:



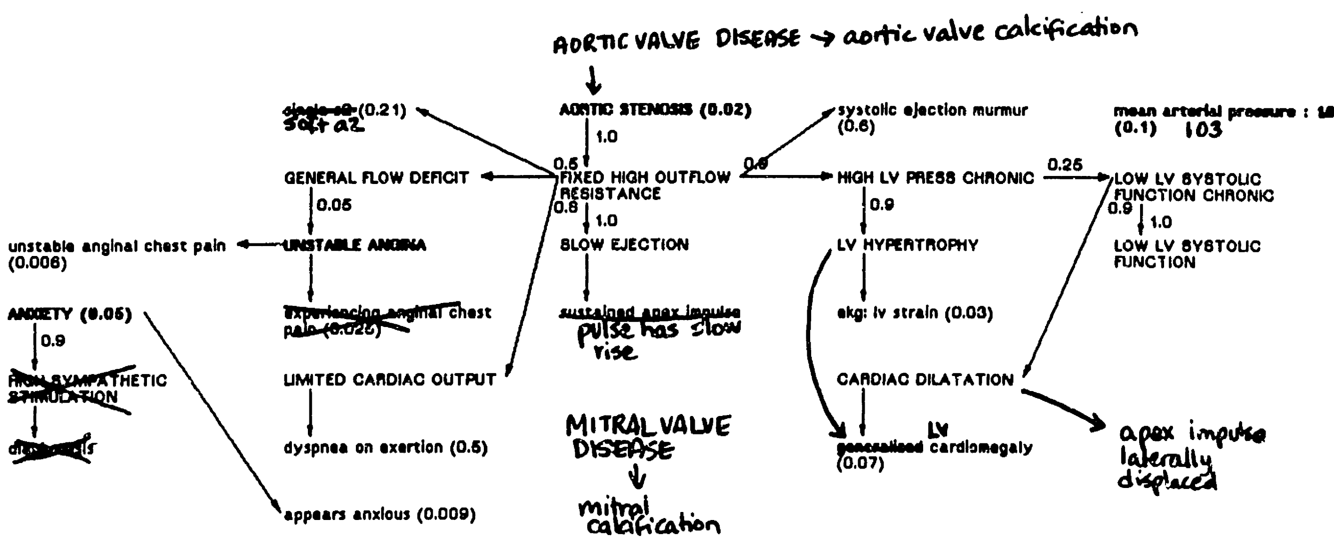
CASEY's result: satisfactory.
Unexplained features: diaphoresis.
Tranferred from patient: Natalie



Patient: NATALIE
Heart Failure program's solution:

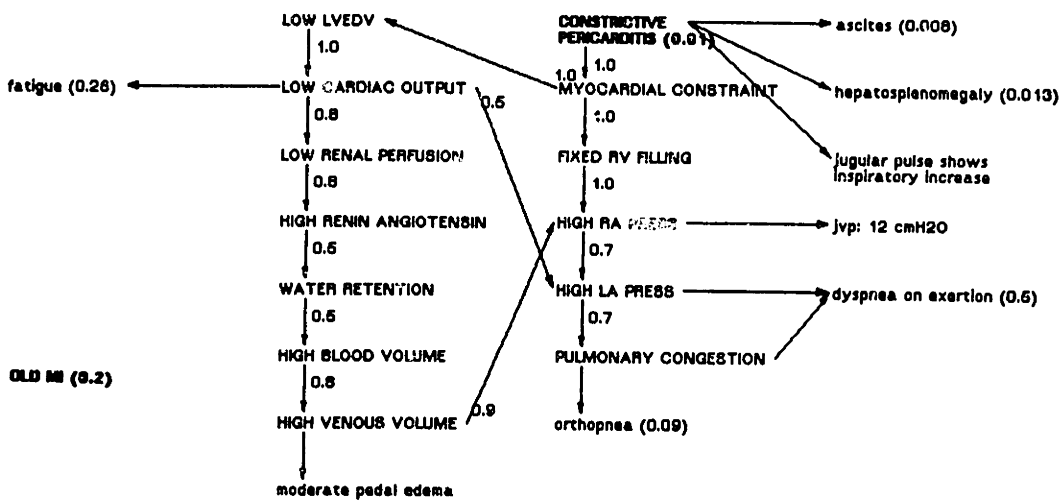


CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Margaret



Patient: NATE

Heart Failure program's solution:

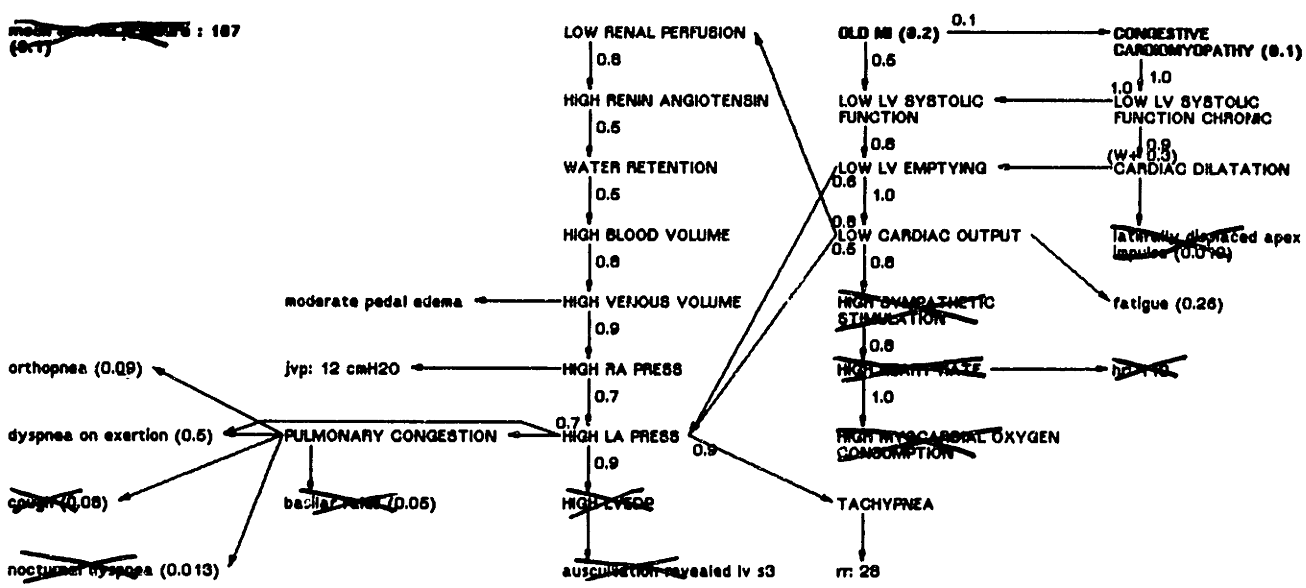


CASEY's result: gives up.

Unexplained features: jugular pulse, **hepatosplenomegaly, ascites.**

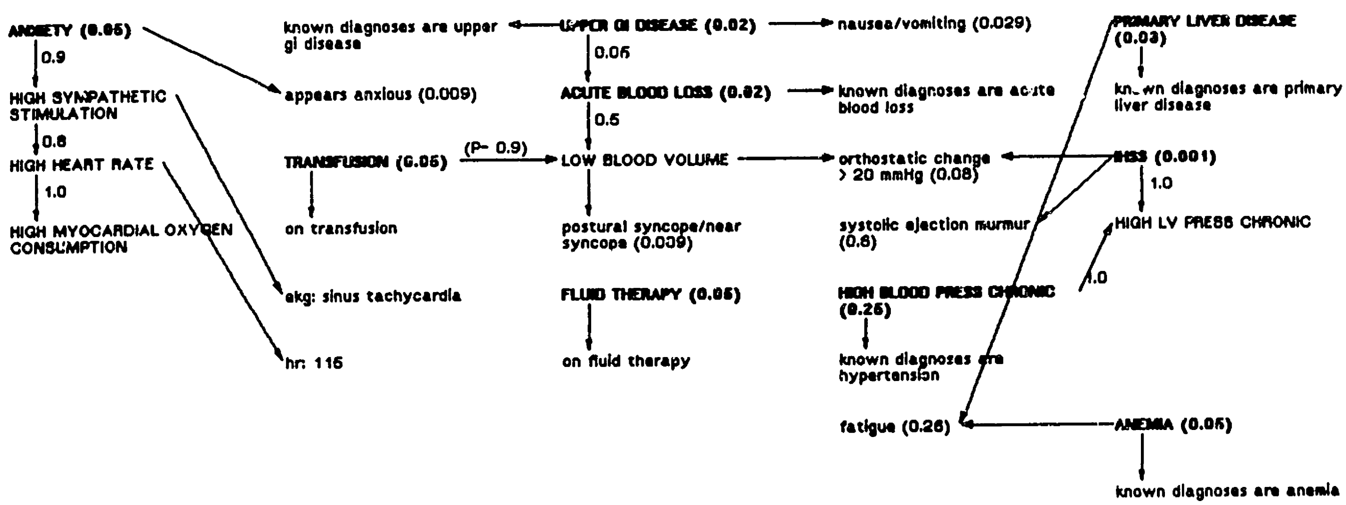
Tranferred from patient: Peter

Note: CASEY gets most of this causal explanation correct, but it mistakenly attributes the primary cause to CONGESTIVE CARDIOMYOPATHY.



Patient: NIGEL

Heart Failure program's solution:



CASEY's result: gives up.

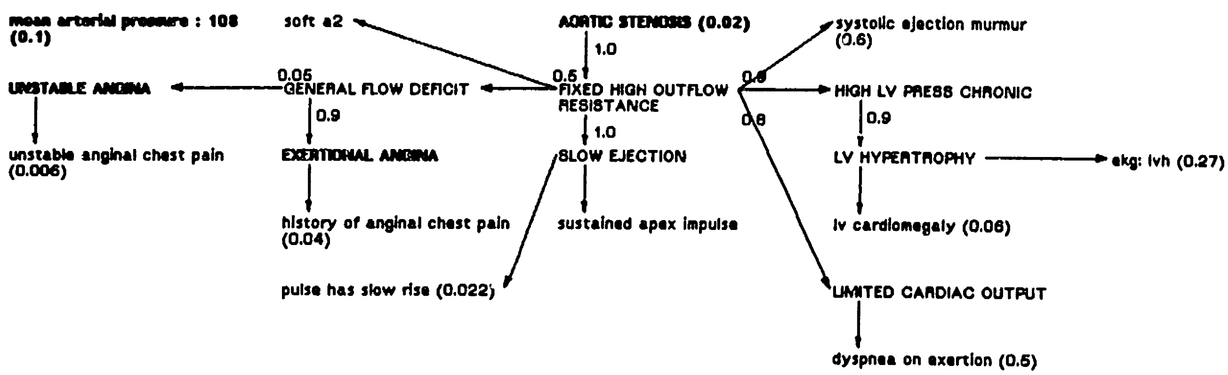
Unexplained features: meany symptoms unaccounted for.

Tranferred from patient: Chin

Note: Nigel is the only patient with this diagnosis in the memory.

Patient: OPRAH

Heart Failure program's solution:

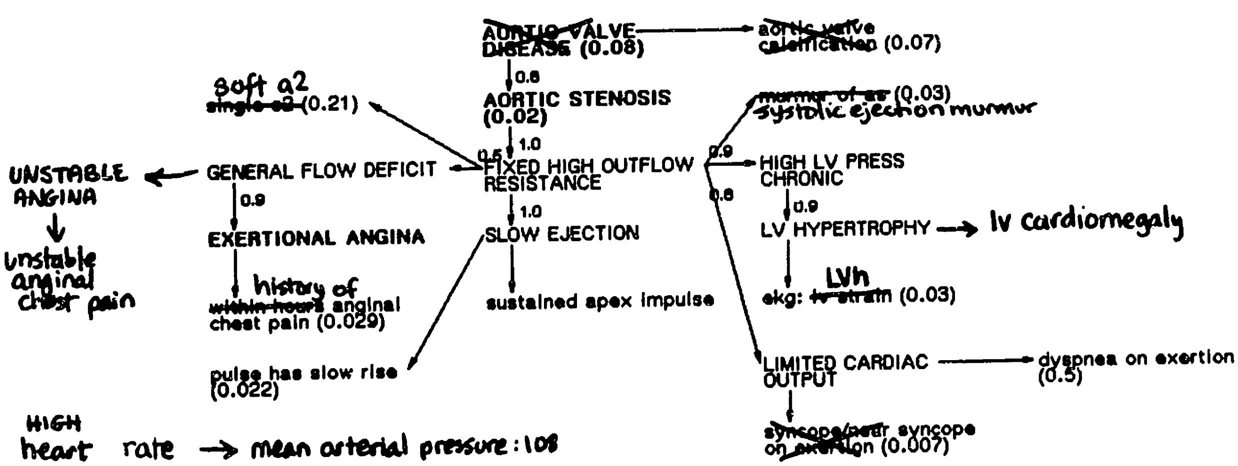


CASEY's result: satisfactory.

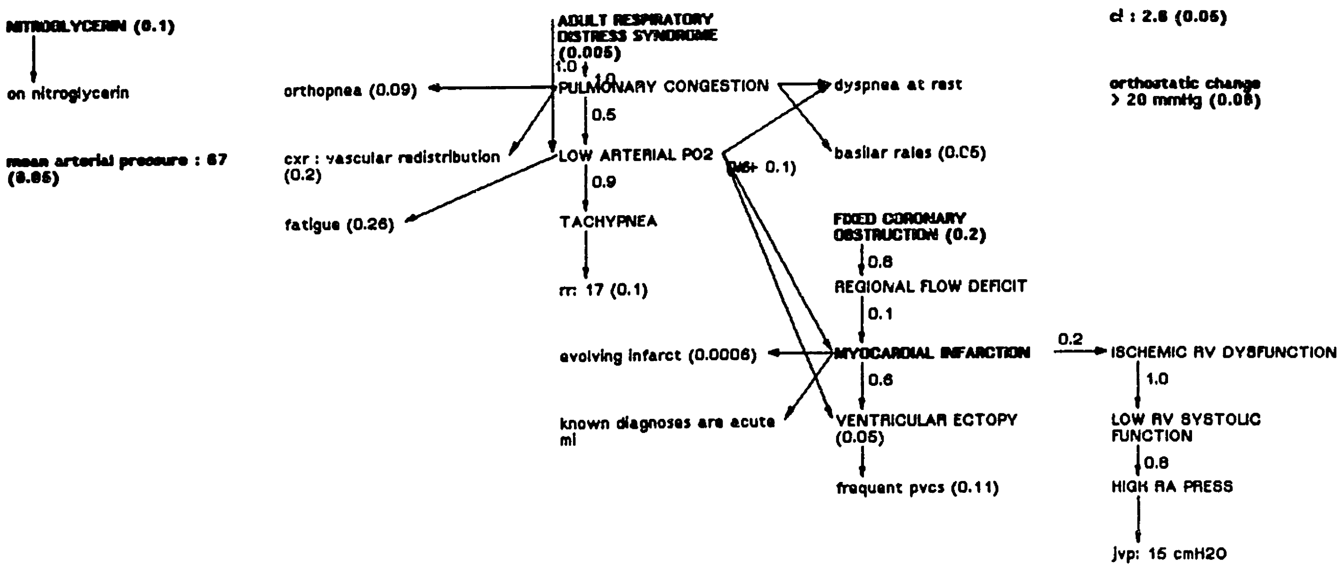
Unexplained features: unstable anginal chest pain.

Tranferred from patient: Adam

Note: CASEY included an additional state, high blood pressure.

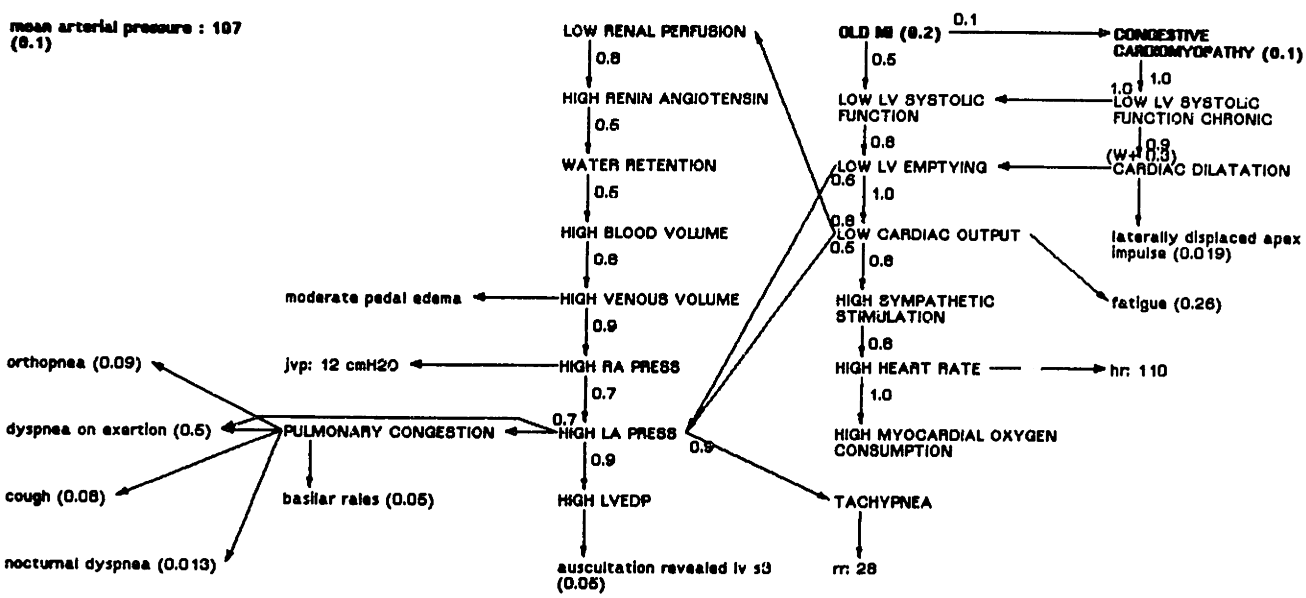


Patient: PATRICK
Heart Failure program's solution:

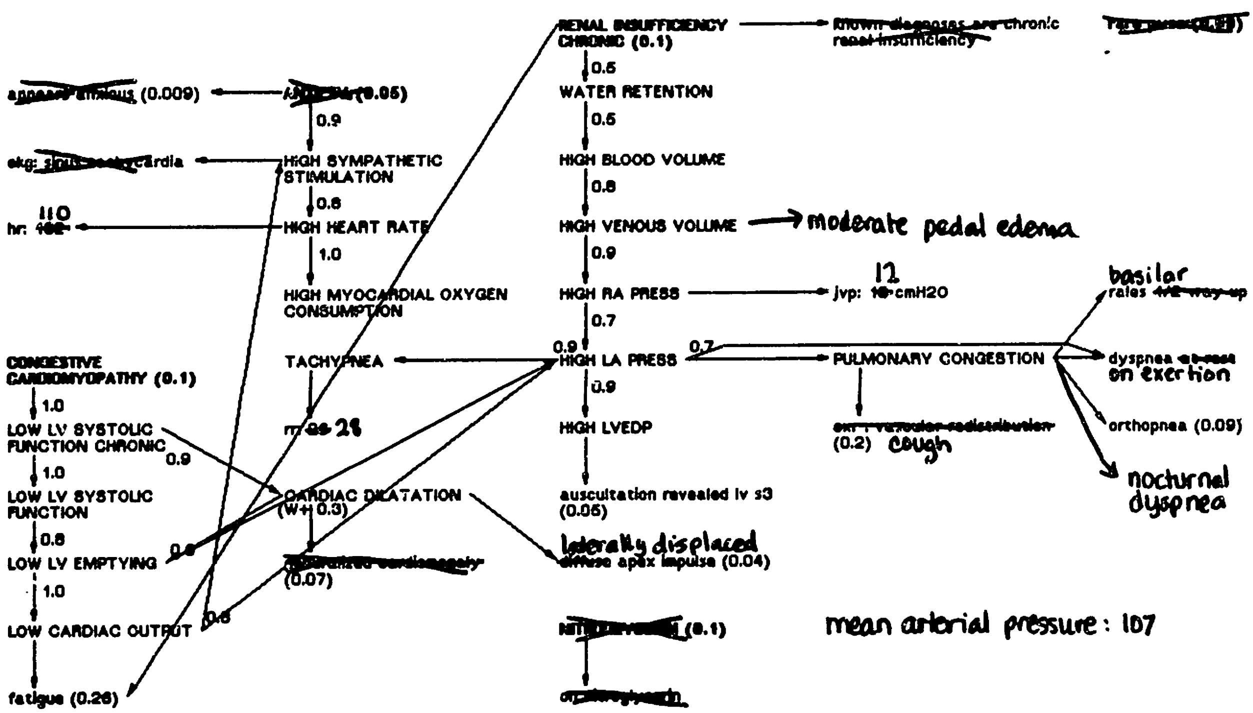


CASEY's result: gives up.
Note: CASEY chooses precedents with the diagnoses FIXED CORONARY OBSTRUCTION, MYOCARDIAL INFARCTION, and PULMONARY CONGESTION, but each of them leaves meany symptoms unaccounted for.

Patient: PETER
Heart Failure program's solution:

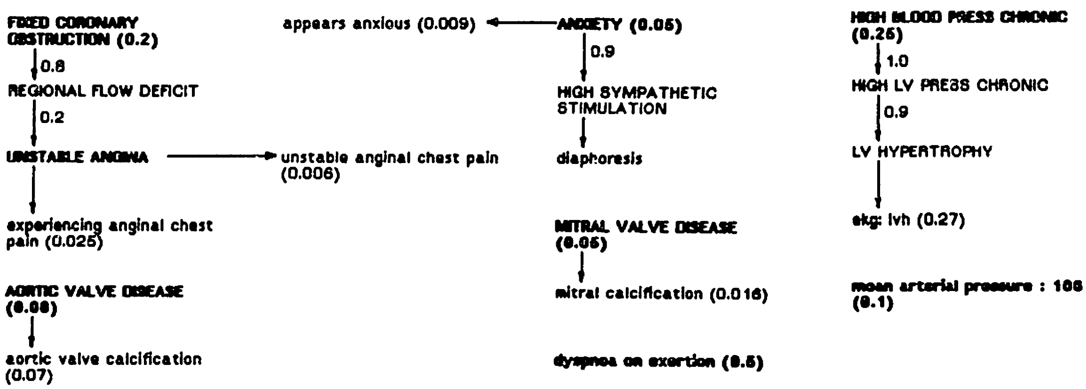


CASEY's result: wrong.
Unexplained features: cough.
Tranferred from patient: Kyle
Note: CASEY gets most of this causal explanation correct, but it mistakenly attributes the primary cause of the HIGH VENOUS VOLUME to RENAL INSUFFICIENCY CHRONIC.


RENAL INSUFFICIENCY CHRONIC (0.1)
WATER RETENTION
HIGH BLOOD VOLUME
HIGH VENOUS VOLUME
moderate pedal edema
HIGH RA PRESS
jvp: 12 cmH2O
HIGH LA PRESS
PULMONARY CONGESTION
basilar rales
dyspnea on exertion
orthopnea (0.09)
nocturnal dyspnea
(0.2) cough
HIGH LVEDP
auscultation revealed lv s3 (0.05)
laterally displaced
diffuse apex impulse (0.04)
mean arterial pressure: 107
(0.009)
(0.05)
HIGH SYMPATHETIC STIMULATION
ekg:
HIGH HEART RATE
hr: 110
HIGH MYOCARDIAL OXYGEN CONSUMPTION
TACHYPNEA
rr: 28
CONGESTIVE CARDIOMYOPATHY (0.1)
LOW LV SYSTOLIC FUNCTION CHRONIC
LOW LV SYSTOLIC FUNCTION
LOW LV EMPTYING
LOW CARDIAC OUTPUT
fatigue (0.26)
CARDIAC DILATATION
(W+ 0.3)
(0.07)
(0.1)

Patient: POLO

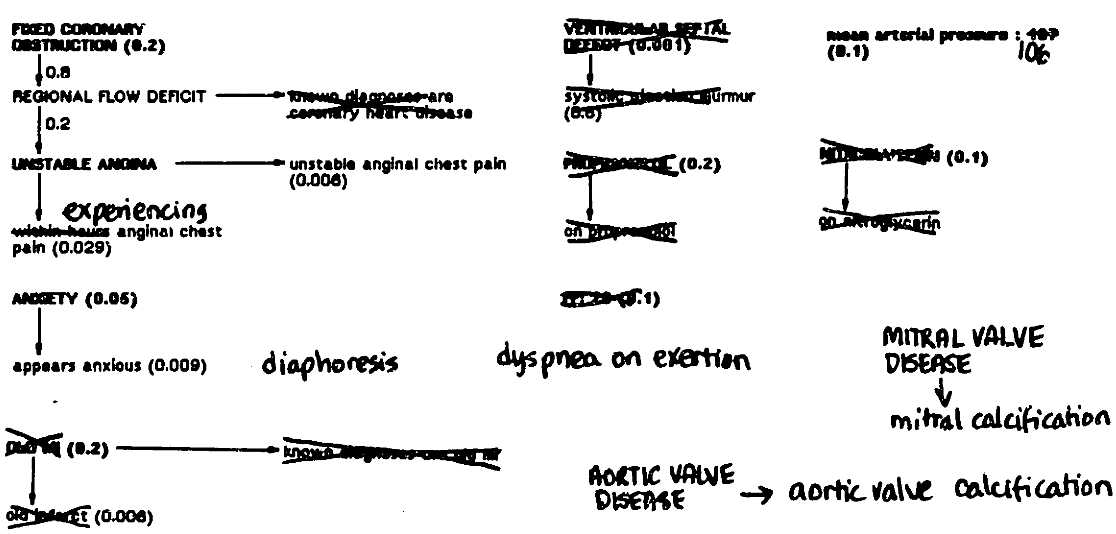
Heart Failure program's solution:



CASEY's result: satisfactory.

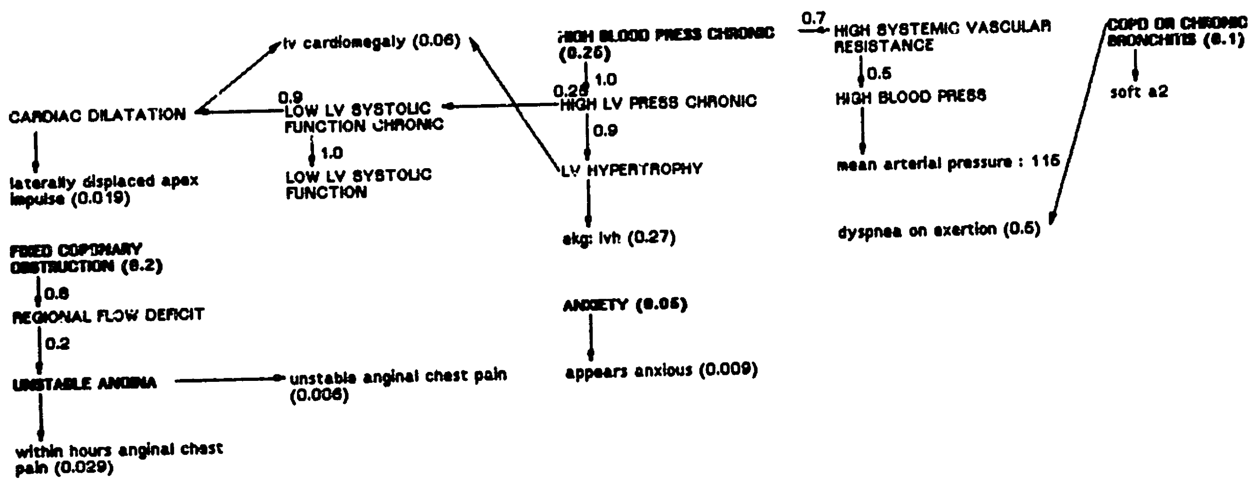
Unexplained features:diaphoresis, ekg: lvh, dyspnea on exertion.

Tranferred from patient: Cody



Patient: RANDY

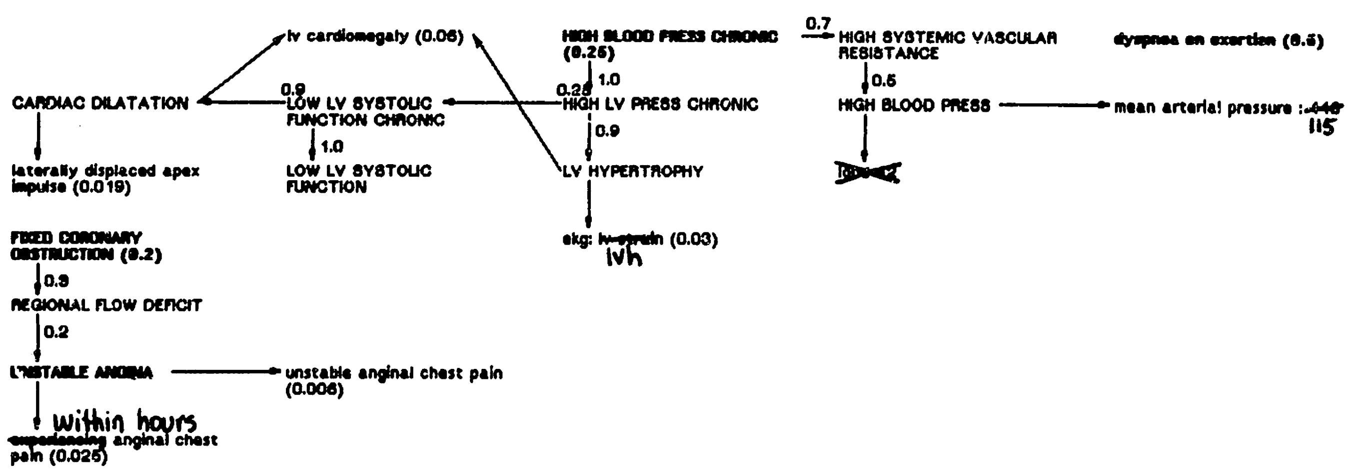
Heart Failure program's solution:



CASEY's result: satisfactory.

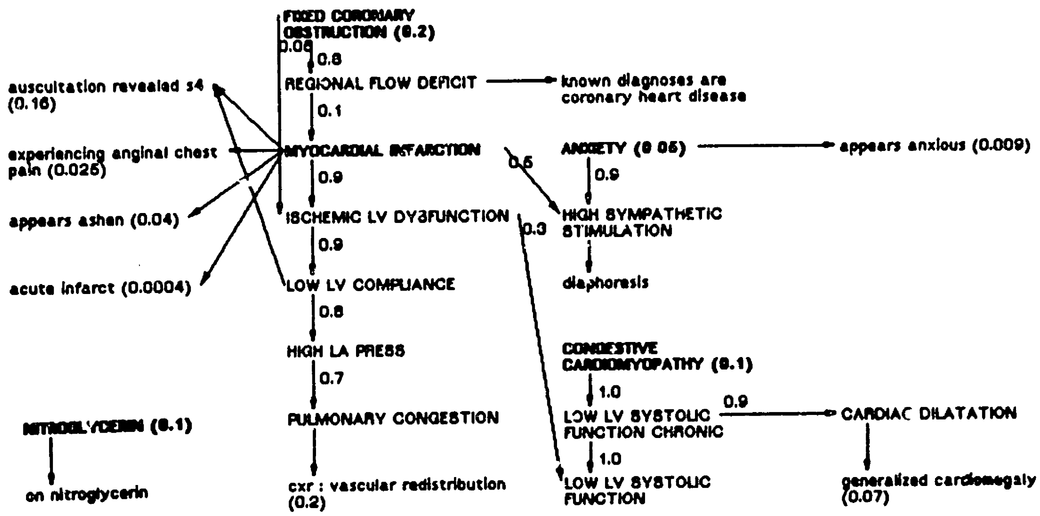
Unexplained features: dyspnea on exertion, **soft a2**.

Tranferred from patient: Thadeus



Patient: SALVATORE

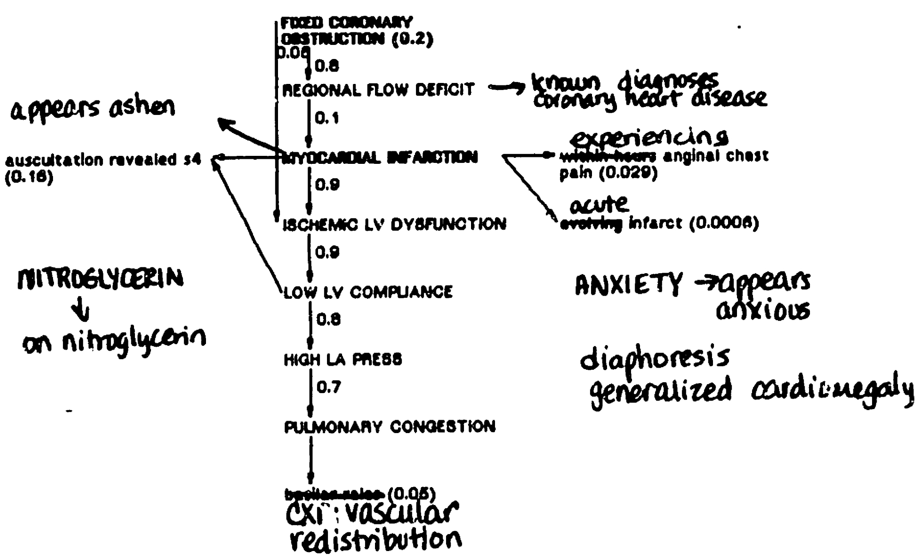
Heart Failure program's solution:



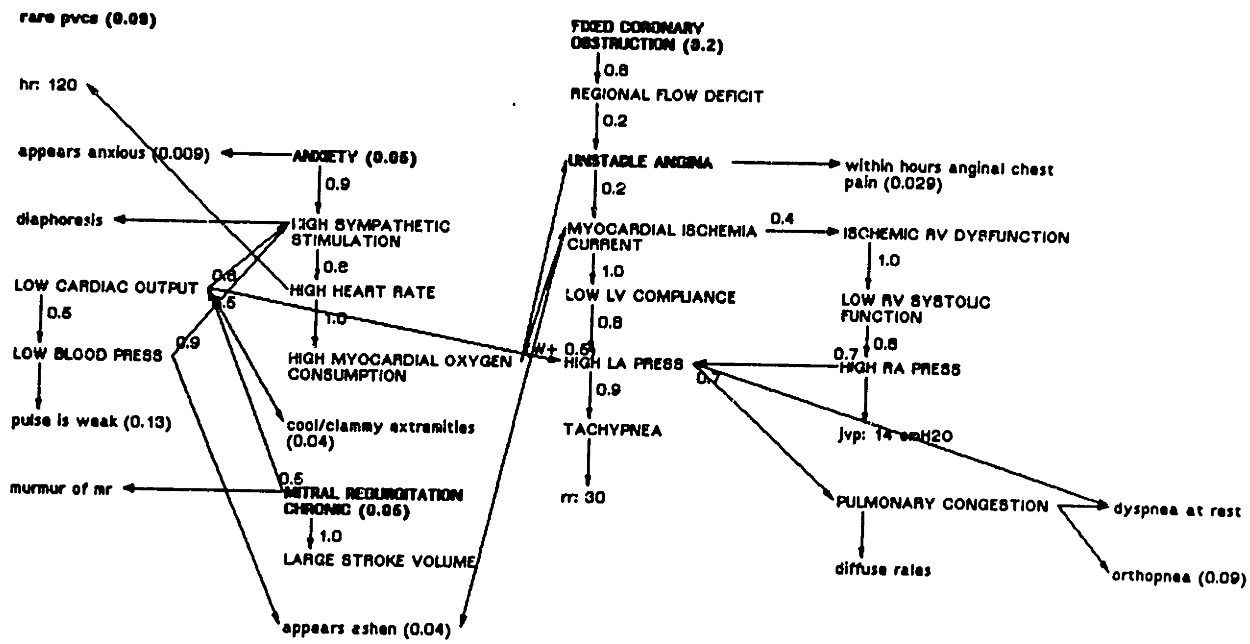
CASEY's result: satisfactory.

Unexplained features: generalized cardiomegaly, diaphoresis.

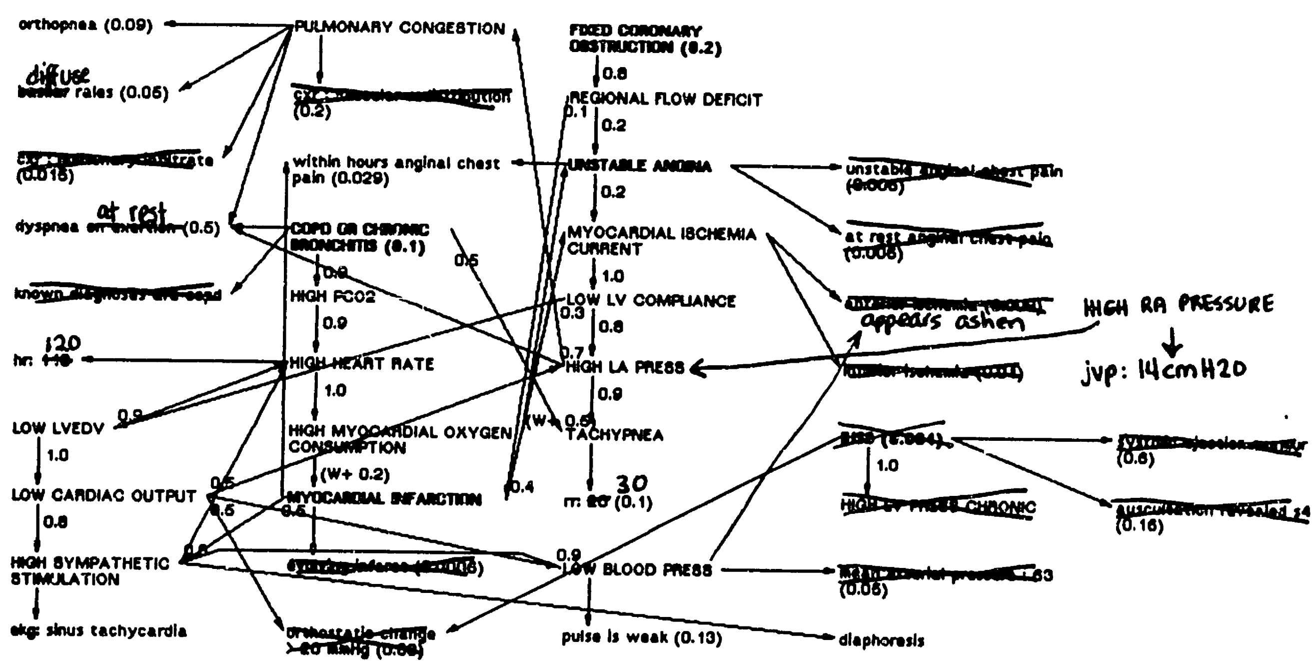
Tranferred from patient: Heywood



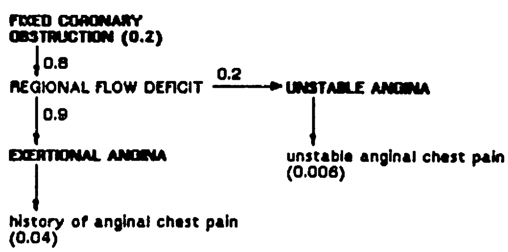
Patient: SAM
Heart Failure program's solution:



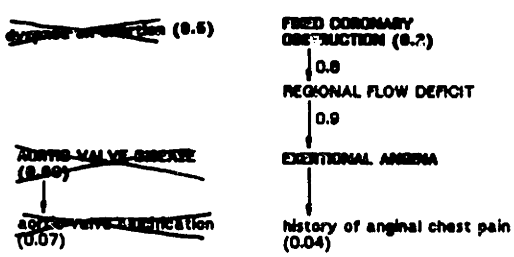
CASEY's result: satisfactory.
Unexplained features: murmur of mitral regurgitation.
Tranferred from patient: Doyle
Note: CASEY's solution is almost correct, however it cannot link the state HIGH RA PRESS to the state MYOCARDIAL ISCHEMA CURRENT.


orthopnea (0.09)
PULMONARY CONGESTION
FIXED CORONARY OBSTRUCTION (0.2)
diffuse
rales (0.05)
(0.2)
REGIONAL FLOW DEFICIT
(0.015)
within hours anginal chest pain (0.029)
UNSTABLE ANGINA
unstable
at rest
dyspnea (0.5)
COPD OR CHRONIC BRONCHITIS (0.1)
MYOCARDIAL ISCHEMIA CURRENT
at rest
HIGH PCO2
LOW LV COMPLIANCE
HIGH RA PRESSURE
appears ashen
120
hr:
HIGH HEART RATE
HIGH LA PRESS
jvp: 14cmH2O
LOW LVEDV
HIGH MYOCARDIAL OXYGEN CONSUMPTION
TACHYPNEA
LOW CARDIAC OUTPUT
MYOCARDIAL INFARCTION
30
rr: (0.1)
HIGH LV PRESS CHRONIC
(0.6)
(0.16)
HIGH SYMPATHETIC STIMULATION
LOW BLOOD PRESS
(0.05)
ekg: sinus tachycardia
pulse is weak (0.13)
diaphoresis

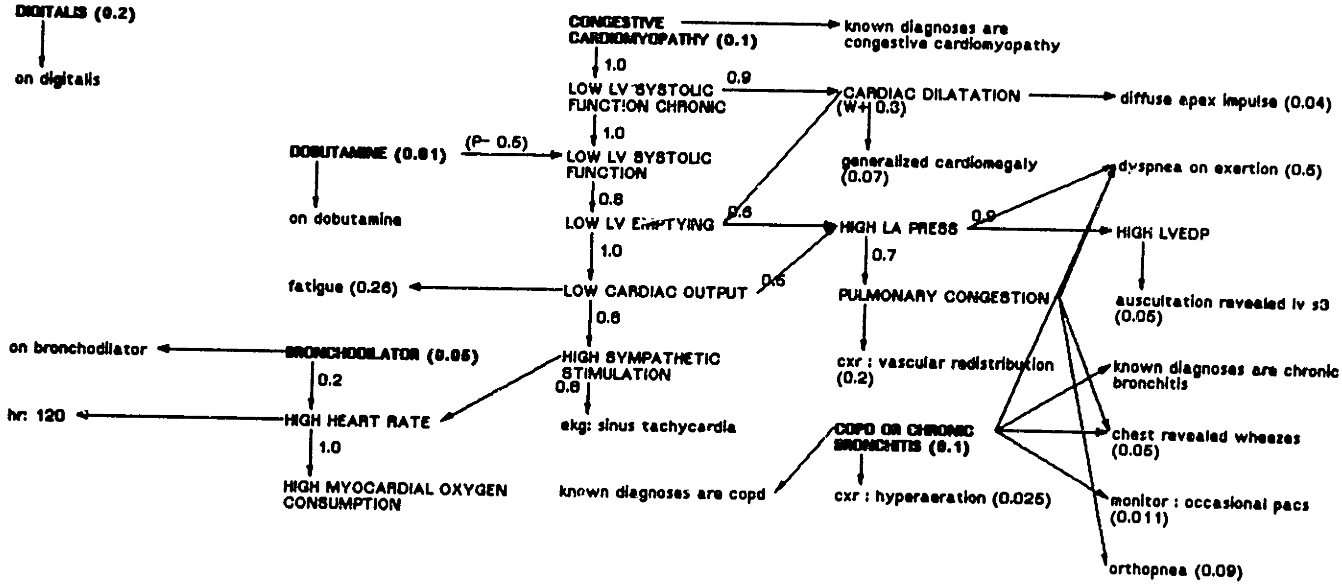
Patient: SARAH
Heart Failure program's solution:



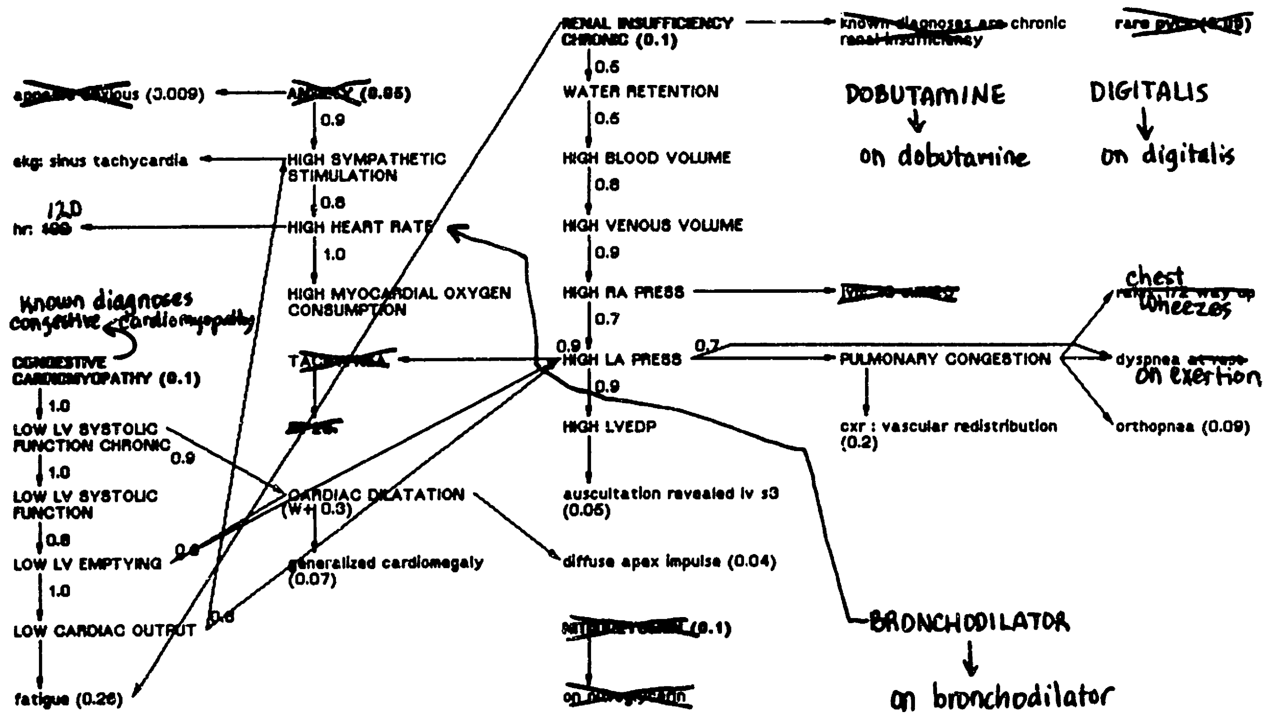
CASEY's result: satisfactory.
Unexplained features: unstable anginal chest pain.
Tranferred from patient: Uri



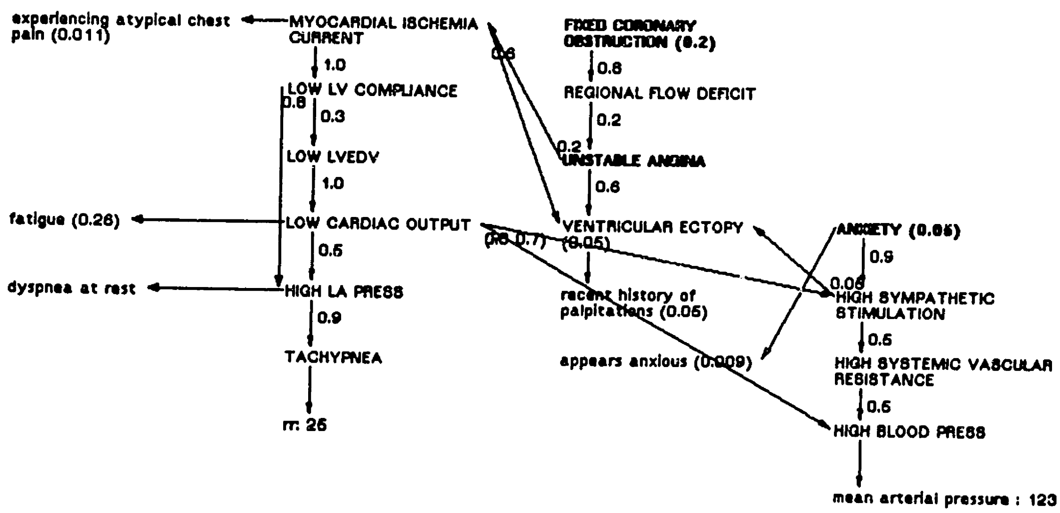
Patient: TAKIS
Heart Failure program's solution:



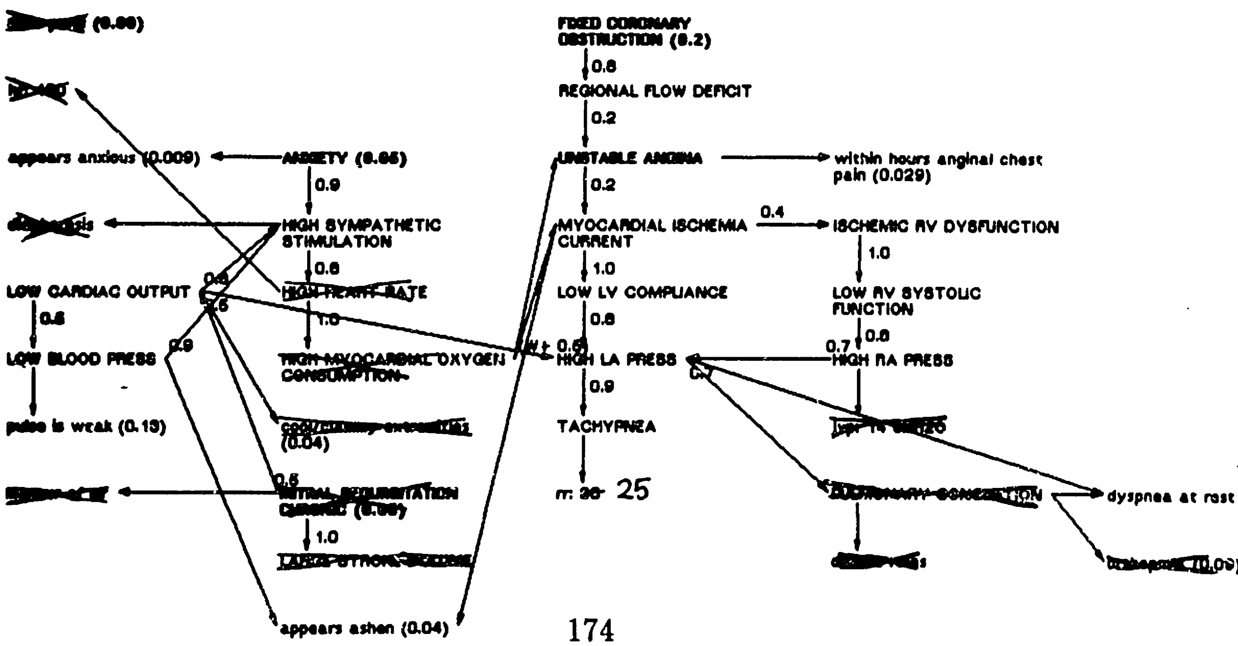
CASEY's result: wrong.
Unexplained features: none.
Tranferred from patient: Kyle
Note: CASEY's solution correctly accounts for all of this patient's symptoms. However, it does not get rid of the diagnosis of CHRONIC RENAL INSUFFICIENCY and its associated states that link to HIGH LA PRESSURE.


RENAL INSUFFICIENCY CHRONIC (0.1)
chronic renal insufficiency
WATER RETENTION
HIGH BLOOD VOLUME
HIGH VENOUS VOLUME
HIGH RA PRESS
HIGH LA PRESS
HIGH LVEDP
PULMONARY CONGESTION
cxr : vascular redistribution (0.2)
auscultation revealed lv s3 (0.05)
diffuse apex impulse (0.04)
dyspnea
orthopnea (0.09)
chest wheezes
on exertion
DOBUTAMINE
on dobutamine
DIGITALIS
on digitalis
BRONCHODILATOR
on bronchodilator
ekg: sinus tachycardia
HIGH SYMPATHETIC STIMULATION
hr: 120
HIGH HEART RATE
HIGH MYOCARDIAL OXYGEN CONSUMPTION
known diagnoses congestive cardiomyopathy
CONGESTIVE CARDIOMYOPATHY (0.1)
LOW LV SYSTOLIC FUNCTION CHRONIC
LOW LV SYSTOLIC FUNCTION
LOW LV EMPTYING
LOW CARDIAC OUTPUT
fatigue (0.26)
CARDIAC DILATATION (W+) 0.3)
generalized cardiomegaly (0.07)

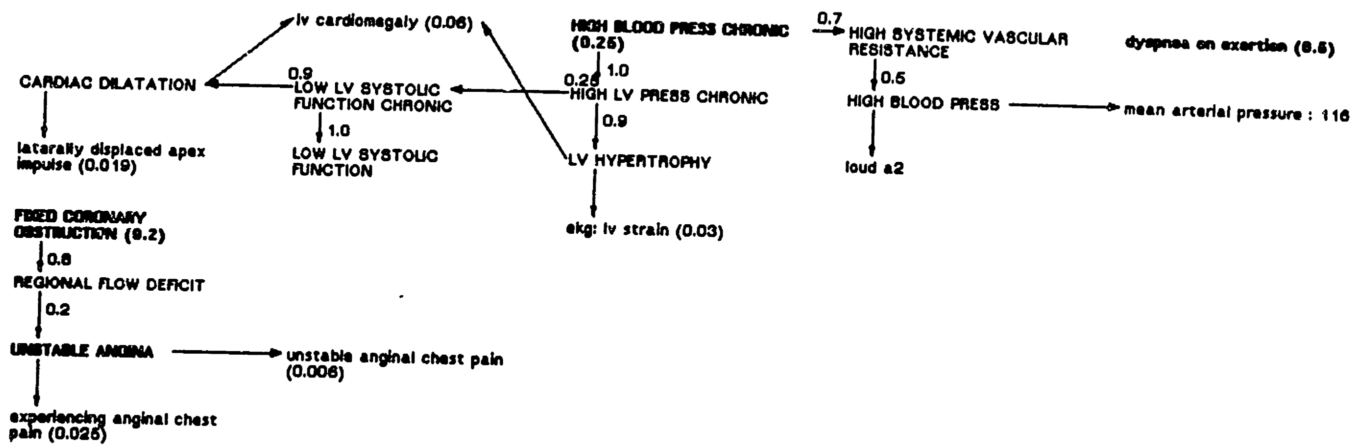
Patient: TED
Heart Failure program's solution:



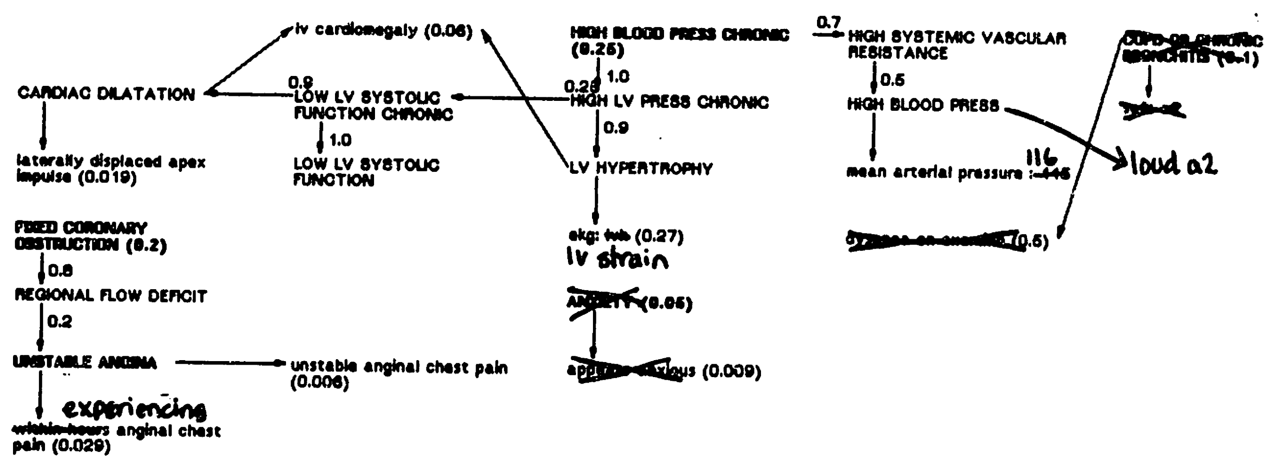
CASEY's result: wrong.
Tranferred from patient: Sam
Note: CASEY incorrectly attributes this patient's findings to the **state** MYOCARDIAL **ISCHEMIA CURRENT.**



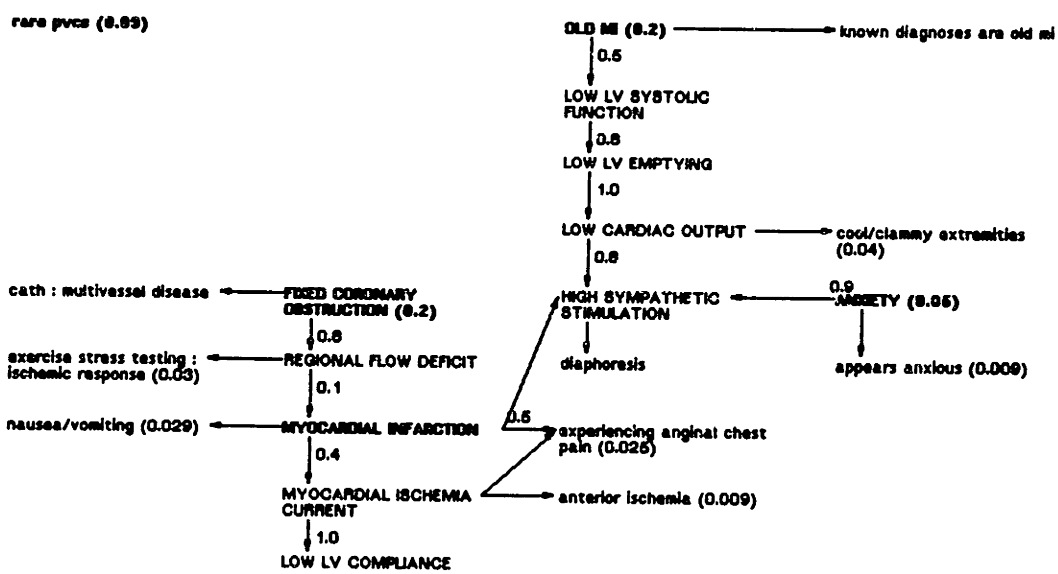
Patient: THADEUS
Heart Failure program's solution:



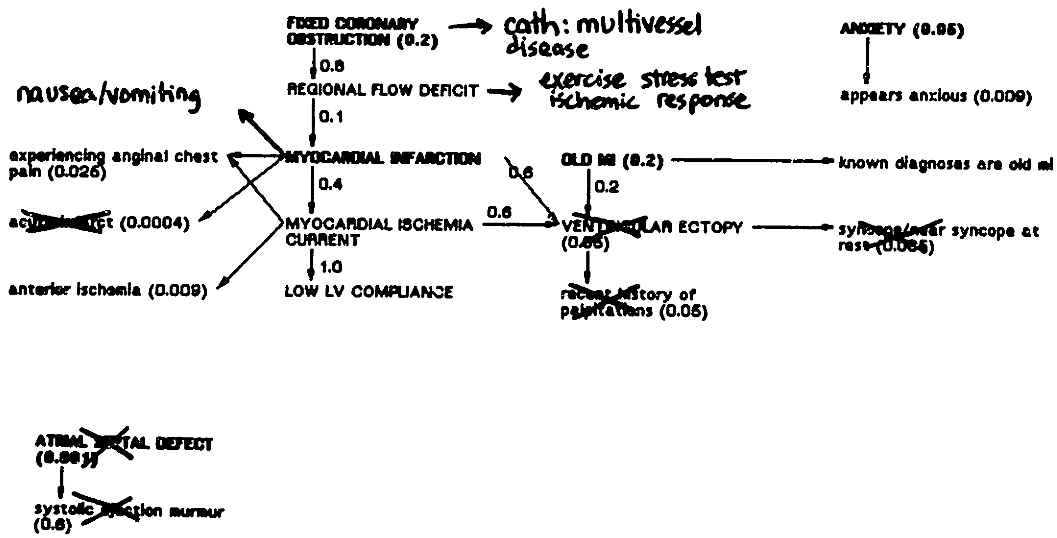
CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Randy



Patient: UMBERTO
Heart Failure program's solution:

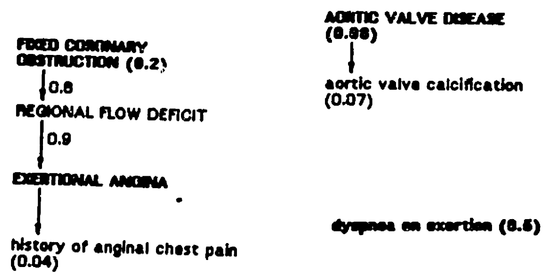


CASEY's result: satisfactory.
Unexplained features: cold, clammy extremities, diaphoresis.
Tranferred from patient: Len

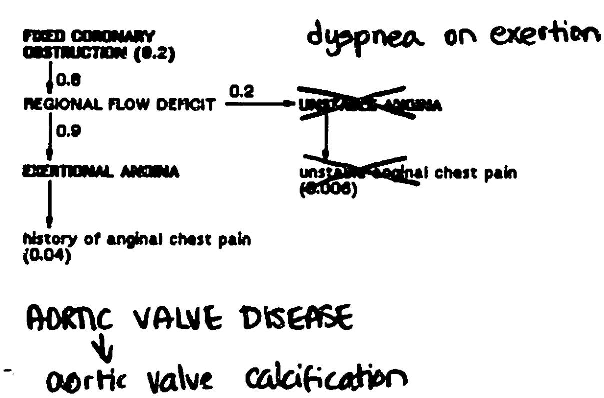


FIXED CORONARY OBSTRUCTION (0.2)
cath: multivessel disease
0.8
REGIONAL FLOW DEFICIT
exercise stress test ischemic response
0.1
nausea/vomiting
experiencing anginal chest pain (0.025)
MYOCARDIAL INFARCTION
0.4
MYOCARDIAL ISCHEMIA CURRENT
1.0
LOW LV COMPLIANCE
(0.0004)
anterior ischemia (0.009)
0.6
0.6
OLD MI (0.2)
0.2
known diagnoses are old mi
ECTOPY
history of
(0.05)
syncope at
ANXIETY (0.05)
appears anxious (0.009)
ATRIAL DEFECT
systolic murmur (0.8)

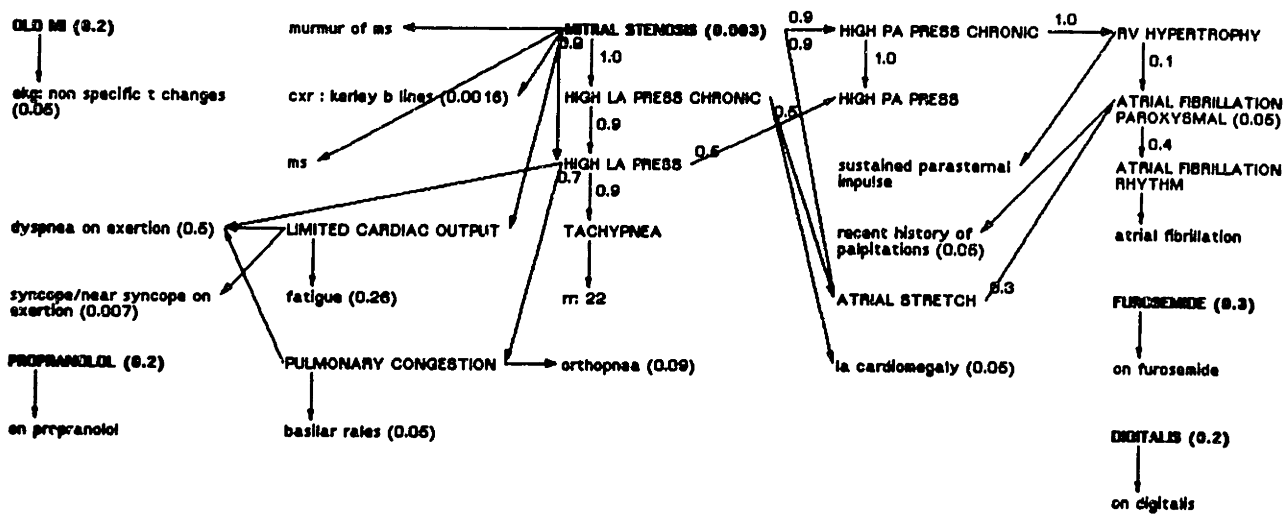
Patient: URI
Heart Failure program's solution:



CASEY's result: identical.
Unexplained features: none.
Tranferred from patient: Sarah



Patient: WILLIAM
Heart Failure program's solution:



CASEY's result: gives up.
Note: CASEY could not account for many of the findings in this case. No other case of MITRAL STENOSIS was in the case memory.