DIAGNOSIS WITH BEHAVIORAL MODES

Johan de Kleer
Brian C. Williams
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto CA 94304

Abstract

Diagnostic tasks involve identifying faulty components from observations of symptomatic device behavior. This paper presents a general diagnostic theory that uses the perspective of diagnosis as *identifying consistent modes of behavior,* correct or faulty. Our theory draws on the intuitions behind recent diagnostic theories to identify faulty components without necessarily knowing how they fail. To derive additional diagnostic discrimination we use the models for behavioral modes together with probabilistic information about the likelihood of each mode of behavior.

1 Introduction

> When you have eliminated the impossible, whatever remains, *however improbable,* must be the truth. — *Sherlock Holmes. The Sign of the Four.*

The objective of our research is to develop a general theory of diagnosis that captures a human diagnostician's predominant modes of reasoning. This theory is intended to serve as the conceptual foundation for computational systems that diagnose devices.

Early approaches [1, 4] to diagnosis used fault models to identify failure modes of faulty components that explain the observations made. The ability to predict failing components' behaviors provided powerful diagnostic discrimination. However, these techniques depend on the assumption that all failure modes are known *a priori* — an assumption that is sometimes warranted but is never guaranteed. The unacceptable result of not satisfying this assumption — faulty diagnoses — has led researchers to abandon this powerful approach.

The model-based diagnostic approach adopted by most recent researchers [3, 6, 10] provides a framework for diagnosing a device from correct behavior only. This approach is based on the observation that it is not necessary to determine how a component is failing to know that it is faulty — a component is faulty if its correct behavior (i.e., as specified by its manufacturer) is inconsistent with the observations. Since only correct behavior needs to be modeled, any knowledge about the behavior of component fault modes is ignored. This provides a fundamental advantage over earlier techniques requiring *a priori* knowledge of all fault modes. Unforeseen failure modes pose no difficulty. However, what is lost is the additional diagnostic discrimination derived from knowing the *likely* ways a component fails, and the ability to determine whether these failure modes are consistent with the observations. Thus, unlikely possibilities are entertained as seriously as likely ones. For example, as far as most model-based diagnostic approaches are concerned, a light bulb is equally likely to burn out as to become permanently lit (even if electrically disconnected).

Human diagnosticians, however, take great advantage of behavioral models of known failure modes, together with the likelihood that these modes will occur. Knowledge of fault modes is used to pinpoint faulty components faster, and to help determine specific repairs that must be made to the faulty components.

We view the central task of diagnosis as identifying the behavioral modes (correct or faulty) of all the components. Whether a mode is faulty or not is irrelevant. Our synthesis hypothesizes that it is not the notion of fault, but behavioral mode that is fundamental to diagnosis. Each component has a set of possible behavioral modes including an unknown mode which makes no predictions, and therefore can never conflict with the evidence. The unknown mode is included to allow for the possibility, albeit small, of unforeseen behavioral modes. This unknown mode is crucial because early diagnostic algorithms, when confronted with an unforeseen fault mode, either start making useless probes or simply give up. Our approach pinpoints the failing component as behaving in an unknown mode.

The introduction of fault models potentially introduces significant computational overhead for the diagnostic algorithms. Diagnosing multiple faults is inherently a combinatoric process. Introducing fault models exacerbates the process, by introducing multiple modes and possible behaviors to consider. To control the combinatorics we introduce computational techniques which focus reasoning on more probable possibilities first. These techniques, in effect, focus diagnostic reasoning only on those component behavioral modes that are more probable given the evidence. This set grows and shrinks as evidence is collected.

By using the new perspective of diagnosis as identifying probable behavioral modes, we are able to extend our earlier work on model-based diagnosis (the General Diagnostic Engine (GDE) [6]) to reason about modes of behavior. The resulting system we call *Sherlock.* GDE provides a general domain-independent architecture for diagnosing any number of simultaneous faults in a device given solely a description of its structure (e.g., electrical circuit schematic) and specifications of correct component behaviors (e.g., that resistors obey Ohm's law). Given a set of observations, GDE constructs hypotheses (called *diagnoses)* identifies the faulty components and suggests points where additional measurements (called *probes)* should be made to localize the diagnosis with as few measurements

as possible.

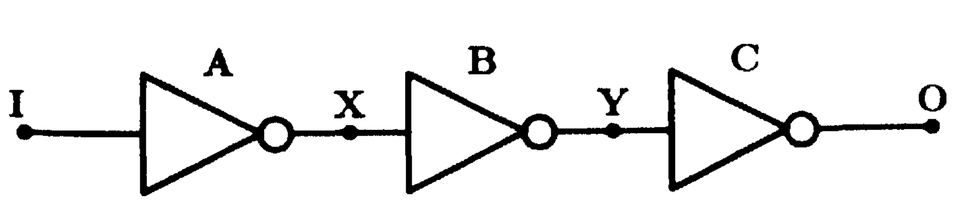
We have implemented our approach by extending GDE to Sherlock, and have tested it on a variety of digital circuits — from simple three inverter circuits, to ALUs consisting of 400 gates with 4 behavioral modes each. Sherlock exploits knowledge of failure modes to pinpoint faults more equally and identify in what mode components are functioning. Sherlock is described more fully in [7].

2 Related work

Exploiting the use of fault models has recently become an active research area [11, 12, 13, 16]. In particular, Holtzblatt's [12] generalization of GDE incorporates the notion of behavioral modes in a similar spirit to Sherlock. But Holtzblatt's GMODS system is missing many key features of Sherlock such as accommodating unexpected failures, incorporating probabilistic information to rank diagnoses and guide probing, incorporating most-probable-first heuristics to limit the computational complexity which arises for larger devices, and combining evidence gathered from multiple observations of a device. As GMODS does not use probabilistic information it relies on an expensive hyperresolution rule to rule out fault modes and cannot focus reasoning on more probable diagnoses. Struss [16] argues against the use of probabilistic information and the use of an unknown mode. Instead he employs a resolution rule and controls reasoning to introduce appropriate fault modes only when necessary. Through the use of an alternative architecture which redefines the notion of fault, Rairnon [13] achieves some of the advantages of knowledge of fault modes without having to incorporate them. Hamscher [11] incorporates fault models with his generalization of GDE called XDE.

3 Diagnosis with modes

The perspective of diagnosis as identifying probable behavioral modes is best appreciated through an example. Consider the simple three inverter circuit shown below. Suppose that the input (/) is set to zero, and that, al-



though the output *(O)* should be one if functioning correctly, it is measured to be zero. Without knowledge of fault modes, all three inverters are equally likely to be faulted. If we knew that inverters (almost) always failed with output stuck-at-1, then we could infer that inverter *B* was likely to be faulted. Thus knowledge of failure modes can provide significant diagnostic information.

Knowledge of failure modes is also important to decide what measurement to make next. If all faults were equally ^kely, measuring *X orY* provides equal information. However, suppose we know that inverters *A* and *B* almost always fail by having their output stuck-at-1, and that inverter *C* almost always fails by having its output stuck-at-0 (because it is designed differently to drive an external load). Given that knowledge, it is unlikely that inverter *A* is failing, as its most common fault does not explain the symptom. If operating correctly, *A's* output should be one, so *A* being stuck-at-1 would not explain the incorrect value being observed at the device's output. However, the likely failures of inverters *B* and *C* are consistent with the symptom since either explains the deviation from expected behavior. Hence the diagnostician should measure at *Y* next to determine which of the two inverters is failing.

The objective of the the diagnosis dictates the granularity of Sherlock's analysis. Sometimes the objective is to identify which components are failing and how. Sometimes the task is simply to identify the failing components so that they can be replaced. Sometimes the task is to identify all the behavioral modes (good and bad). Sometimes diagnosis considers multiple test vectors, and sometimes there is only one. To accommodate these possibilities Sherlock mubi be told which modes it must discriminate among.

It is important to note that even if it is diagnostically unimportant to distinguish between some behavioral modes, knowledge of behavioral modes still helps Sherlock. Suppose a component has two faulty modes, *M* of high probability, and *Mi* of low probability, which we are not interested in discriminating between. If a measurement eliminates *M* from consideration then the *(posterior)* probability that the component is faulty becomes low.

4 Framework

This section presents the overall framework including definitions for basic terminology and equations for computing the relevant probabilistic information. Section 5 presents heuristics for avoiding the combinatorial explosion resulting from moving from GDE to Sherlock.

The key conceptual extension to GDE is the introduction of behavioral modes. The extension is very easy as GDE can be viewed as having two behavioral modes (the good one and the faulty one with unspecified behavior) per component. In Sherlock there are simply more behavioral modes per component.

The *structure* of the device to be diagnosed specifies the components and their interconnections. Components are described as being in one of a set of distinct *modes,* where each mode captures a physical manifestation of the component (e.g., a valve being open, closed, clogged or leaky). The behavior of each component is characterized by describing its behavior in each of its distinct modes. We require that a component can be in only one mode at a time. W. also require that a faulty component must remain in the same mode for all test vectors (in the exceptional case where a fault cannot be modeled this way, its behavior is captured by the unknown mode). Other than these there are very few restrictions on behavior models: a model can make incomplete predictions, the set of modes can be incomplete, and the predictions of different modes can overlap.

A component's modes consists of a set describing the component's proper behavior (e.g., the valve being on or off), and a set describing faulty behavior (e.g., the valve being clogged or leaky). When there is only one mode for

proper behavior we abbreviate it as *G* (for "good"). Formally, a behavioral mode is a predicate on components, which is true of a device exactly when the device is in that behavioral mode. Every component has an unknown mode, *U* with no behavioral model, representing all (failure) modes whose behaviors are unknown.

In our example, we consider four behavioral modes of a digital inverter: good (abbreviated G), output stuck-at-1 (abbreviated 51), output stuck-at-0 (abbreviated 50), and an unknown failure mode (abbreviated *U).* The axioms for the behavior of the inverter are:

$$INVERTER(x) \rightarrow [$$
$$[G(x) \rightarrow [IN(x) = 0 \equiv OUT(x) = 1]] \wedge$$
$$[S1(x) \rightarrow OUT(x) = 1] \wedge$$
$$[S0(x) \rightarrow OUT(x) = 0]].$$

The unknown behavioral mode *U(x)* has no model.

Given the model library and the device structure Sherlock directly constructs a set of axioms 5D, called the *system description* [14].

An *observation* is a set of literals describing the outcomes of measurements (e.g., {/ = 0,X = 1,0 = 0}) for a test vector which has been applied to the device. The *evidence* consists of a set of observations (e.g., {{/ = 0, *X* — 1,0 = 0},{I = 1,0 = 0}}).[1] This definition of allows us to incorporate accumulated evidence from different test vectors.

A *candidate* assigns a behavioral mode to every component of the device. Intuitively, a diagnosis is a candidate that is consistent with the evidence, however, we distinguish between a diagnosis for a particular observation and a diagnosis for all the evidence. A *diagnosis for an observation* is a candidate that is consistent with the observation — formally, that the union of the system description, the candidate, and the observation is logically consistent[2]. Formally a candidate is a set of literals, e.g., *{G(A), G(B), U(C)}.* To distinguish sets representing candidates we write *[G{A),G(B),U(C)].* Note that in GDE a candidate is represented by the set of failing components, while in Sherlock a candidate is represented by a set that assigns a behavioral mode to every component. Thus, the Sherlock candidate *[G(A),G(B), U(C)]* corresponds to the GDE candidate [C].

In combining information from different observations we need to treat good and bad modes differently. By definition, a component manifests the same failure mode throughout all observations. However, if a component is in a good mode (e.g., valve is on) in one observation there is no reason to believe it should be in the same good mode for another test vector. If components have only a single good mode, combining information from multiple test vectors is straight-for ward. Namely, a *diagnosis for the evidence* is a set of literals such that for every observation, the union of the system description, the candidate, and the observation is logically consistent. For brevity we operate within one observation, in the remainder of this paper, unless otherwise indicated. However, it is important to bear in mind that many of the design decisions underlying Sherlock only make sense when multiple observations are taken into consideration.

Like GDE, we make the basic assumption that components fail independently (which is sometimes unfounded) and that the prior probabilities of finding a component in a particular mode are provided. Recall that, although the behaviors of the different modes may sometimes overlap, we require that each mode captures a distinct physical state or condition of the component. Thus, the probabilities of all the modes of a component always sum to one. Under these assumptions, the prior probability that a particular candidate C_1 is the actual one is:

$$p(C_l) = \prod_{m \in C_l} p(m).$$

where *p(m)* denotes the prior probability of behavior mode m being manifested (i.e., a particular component being in a particular mode).

As candidates are eliminated, the probabilities of the remaining diagnoses must increase. (On occasion a candidate is eliminated purely as a result of the device's topology in which case the probability is adjusted by a renormalization.) Usually candidates are eliminated as a result of measurements. Bayes rule allows us to calculate the conditional probability of the candidates given that point x, is measured to be vik (unless otherwise indicated, all probabilities are conditional on evidence previously accumulated. See [6] for more details):

$$p(C_l | x_i = v_{ik}) = \frac{p(x_i = v_{ik} | C_l) p(C_l)}{p(x_i = v_{ik})}.$$

The denominator, p(x, = vik), is just a normalization. *p(C1)* was computed as a result of the previous measurement (or is the prior). Finally, $p(x_t = vik|C)$ is determined as follows:

1. If $x_i = v_{ik}$ **is predicted by** *C*\ given the evidence so far **then** $p(x_i = v_{ik}|C_l) = 1$.
2. If $x_i = v_{ik}$ **is inconsistent**, with *C*\ and the evidence **then** $p(x_i = v_{ik}|C_l) = 0$.
3. If x, = *vik* is neither predicted by nor inconsistent with *C*\ and the evidence then we make the presupposition (sometimes invalid) that every possible value for x, is equally likely. Hence, $p(x_i = v_{ik}|C_l) = \frac{1}{m}$ where *m* is the number of possible values x, might have (in a conventional digital circuit *m* = 2). Intuitively, this provides a bias for candidates which predict a measurement over those that don't.

Throughout the diagnostic session, the probability of any particular observation x, = vik is bounded below by the sum of the current probabilities of the candidates that entail it and bounded above by one minus the sum of the current probabilities of the candidates that are inconsistent with it. See [6] for the estimate used. Similarly, the probability that a component is in a particular mode is given by the sum of the current probabilities of the candidates in which it appears.

pThe process of generating good test vectors is outside the present scope of our theory.

[2]Note that by this definition some candidates may be eliminated as diagnoses on the basis of no observations whatsoever. For example, consider hypothetical models for two inverters in series where the first inverter had a mode output-stuck-at-1 and the second had a mode input-stuck-at-0. Note also that the candidate in which every component is operating in its unknown mode is always a candidate unless the combination of the system description and any observation by itself is inconsistent.

4.1 Varieties of diagnostic tasks

In order to determine what next measurement is likely to provide the most information, Sherlock must determine the likelihood of hypothetical measurement outcomes and its consequences on the candidate space. The different diagnostic objectives dictate differing scoring functions. Sherlock is asked to discriminate among some modes and not others; by supplying Sherlock with sets of discrimination specifications — (a set of modes that are not to be discriminated). The discrimination specification partitions the diagnoses into a set of d-partitions. The goal of diagnosis is to identify the probable d-partitions and to suggest measurements which best pinpoint the actual one. For example, it may only be important to discriminate between good and faulty behavior. In this case, the most probable d-partition identifies which components have to be replaced. In the simple case where the objective is to discriminate among all behavioral modes, then every d-partition is just a singleton set consisting of a single diagnosis. Note that, in general, the different diagnoses within a single d-partition make different predictions. Although it may be unimportant to discriminate among them as far as the overall diagnostic objective is concerned, it is important to keep them separate to correctly compute the probabilities of measurement outcomes.

The specific approach used to select measurements is a *minimum entropy* technique — pick that measurement to make next that will yield, on average, the minimum entropy *H* (or conversely that measurement which extracts maximum information):

$$H = -\sum p(D_l) log\ p(D_l).$$

Where *p(D1)* is the probability of a d-partition given evidence. This, in turn, requires computing the candidate probabilities given a hypothetical outcome. Fortunately this is computable fairly directly using Bayes rule (see [6] for details). The expected entropy resulting from measuring x, is:

$$H_e(x_i) = \sum_{k=1}^{k=m} p(x_i = v_{ik}) H(x_i = v_{ik}),$$

where vik are the possible measurement outcomes and *H(xi = Vik)* is the entropy of the resulting set of d-partitions. Information theory tells us that, given certain assumptions, the measurement chosen by this scoring function will on average enable Sherlock to make the fewest number of measurements to identify the actual d-partition to a certain level of confidence. This approach (see examples in [6]) almost always suggests the optimum measurement common sense would suggest. The subsequent examples restate entropy as a cost function: ideal measurements have 0 cost, and useless measurements have cost 1.

If there are multiple test vectors, far greater care must be taken. Suppose the objective is to identify the faulty components and how they are faulted. In this case Sherlock need only discriminate among faulty modes. The d-partitions for the overall objective are the intersection of those obtained from each of the multiple test vectors. In computing He(x,) we must take care to use these global d-partition, but only use the relevant candidates for determining $p(x = V_{ik})$ for a test vector. Thus, Sherlock identifies not only the best place to measure but also the best test vector (given the test vector set with which it has been supplied) under which to make the measurement.

4.2 Algorithms commmon to GDE and Sherlock

Sherlock, like GDE, exploits an assumption-based truth maintenance system (ATMS)[5]. Every literal stating that some component is in some behavioral mode is represented by an ATMS assumption. A literal indicating measurement outcome (e.g., *IN(A)* = 0) is represented by an ATMS premise[3]. The underlying Sherlock algorithms are similar to those of GDE except components can have multiple modes.

Sherlock computes the diagnoses by first constructing a set of conflicts. A *conflict* is a set of component behavioral modes which is inconsistent with the system description and some observation (i.e., a conflict is represented by an ATMS nogood). A conflict contains at most one behavioral mode per component. As in GDE, we represent the set of conflicts compactly in terms of the minimal conflicts, since conflicts are ordered by set-inclusion: every superset of a conflict is necessarily a conflict as well.

Intuitively, a minimal conflict identifies a small kernel set of component behavioral modes which violates some observation. It is easily shown that a candidate is a diagnosis iff it does not contain any minimal conflict. Thus, the complete set of diagnoses is computable from the minimal conflicts alone. Thus, Sherlock attempts to determine the minimal conflicts (in ATMS terminology these are minimal nogoods) as these provide the maximum diagnostic information.

Sherlock is typically used with a sound but incomplete prediction facility. Although soundness guarantees the conflicts Sherlock discovers are indeed conflicts, incompleteness sometimes makes it impossible to identify the minimal conflicts and consequently fails to rule out candidates as diagnoses. In the rest of this paper by minimal conflicts we simply mean the set of unsubsumed conflicts found by Sherlock, and by diagnosis we mean a candidate not ruled out by one of these conflicts. The consequences of incompleteness are not catastrophic and usually result in only a minimal degradation in diagnostic performance. This issue is discussed in more detail in [6].

In order to select the next measurement (and under which test vector) to make, Sherlock must evaluate the effects of a hypothetical measurement. To do so, Sherlock must be able to determine what possible outcomes hold in which candidates. Sherlock computes the sets of behavioral modes which support each possible outcome. If an outcome follows from a set of behavior modes, then it necessarily follows from any superset. Therefore, Sherlock need only record with each possible outcome the minimal sets of behavior modes upon which it depends. Thus a possible measurement outcome holds in a candidate if a set of behavioral modes supporting the outcome is a subset of the candidate. Each set of behavioral modes supporting an outcome is represented by an ATMS *environment* and

[3]To implement the search strategy discussed in the next section these literals have to be assumptions as well but this is outside the scope of this paper.

the set of all environments for an outcome is represented by an ATMS *label.* The details for this algorithm can be found in [5, 6]. In a later section we work through a simple example illustrating Sherlock's functioning.

5 Controlling the combinatorics

The presence of behavioral modes has two immediate consequences affecting the algorithms: (1) there are far more behavioral modes to reason about, and (2) the concept of minimal diagnoses which was so useful to GDE is now virtually meaningless. For example, if there are *n* components, each with *k* behavioral modes, there are k^n candidates which might have to be considered (as opposed to GDE's 2^n). Together these consequences make Sherlock significantly slower than GDE. This potential combinatorial explosion manifests itself in two ways in Sherlock. First, the set of conflicts, as well as the sets of behavioral modes underlying possible outcomes (i.e., the ATMS labels, explodes). This causes the prediction phase of Sherlock to explode. Second, the number of possible diagnoses is exponential, causing candidate generation to explode. Thus, the Sherlock architecture adds two tactics beyond those used in GDE to keep the combinatorial explosion under control. These tactics apply to GDE as well as Sherlock.

The focussing tactics do not affect the set of diagnoses produced (or probabilities ratios among them). We first present our strategy for the diagnostic objective of identifying all fault modes, and then later show it can be modified to find the best cf-partitions. The basic idea is to focus reasoning to the subset of the diagnoses (called *leading diagnoses)* that satisfy the following conditions:

- All leading diagnoses have higher probability than all non-leading diagnoses.
- There are no more than ibi (usually *k*\ — S) leading diagnoses. The exception is that all diagnoses having probability approximately equal to the Jbxth diagnosis are included (to accommodate roundoff difficulties).
- Candidates with probability less than ^th (usually A2 = 100) of the best diagnosis are not considered.
- The diagnoses need not include more than k_3 (usually &3 = -75) of the total probability mass of the candidates.

This approach focusses candidate generation to a small tractable set of leading diagnoses.

The primary remaining source of combinatorial explosion is the size of the ATMS labels for Sherlock's predictions. This is dealt with using a generalization of the focussing strategies outlined in [9] and are similar to some suggested in [8]. To handle this both the ATMS and the underlying constraint propagator used by Sherlock are restricted to focus their reasoning only on the leading diagnoses or tentative leading diagnoses. No prediction is made unless its results hold (i.e., one of its environments is a subset of some focus environment) in the current focus. Furthermore, no environment is added to any ATMS label unless it holds in some current focus. If the ATMS discovers an environment not part of any current diagnosis, it does not add it to the prediction's label and instead stores it on its "blocked" label.

Unfortunately, there is a bootstrapping problem. The leading diagnoses cannot be accurately identified without sufficient minimal conflicts. The reasoning cannot produce enough minimal conflicts unless there are leading diagnoses to focus on. Another complication is that Sherlock cannot correctly evaluate the probability of a candidate via Bayes rule unless it is in the focus.

The following is an outline of the procedure Sherlock uses to identify the leading diagnoses and consists of a backtracking best-first search coupled with focussing tactics just discussed. The normalization factor of Bayes rule is left out in the search since it does not change the probability ordering of diagnoses and is the same for all candidates. The search estimates the probability of a tentative diagnosis — a candidate which is consistent with the predictions (more precisely contains no known conflict as a subset), but which has not yet been focussed upon — to be simply its prior probability (corrected by the normalization). This is an upper bound of its correct probability. Focussing the attention of the predictor on the tentative diagnosis might produce a conflict which eliminates it (i.e., drives its probability to zero) or it might be discovered that the diagnosis does not predict every measurement outcome (in which case its probability needs to be adjusted downwards by Bayes rule). Using these techniques the following search guarantees that it finds the same leading diagnoses an unfocussed Sherlock would find.

1. If, according to the criteria, there are sufficient leading candidates, stop. Let *b* be the upper-bound of the probabilities of the diagnoses which are reachable from the next place to push the best-first search forward. The key test is: is 6 less than the leading candidates?
2. Continue a best-first search for the next highest-probability (estimated by its upper bound) candidate which accounts for all the minimal conflicts.
3. Focus the predictor on the candidate (i.e., by unblocking the ATMS labels and permitting consumer execution). This finds any conflicts. It also finds any new predictions which follow from this candidate but which haven't been discovered earlier.
4. If the candidate contains a conflict, go to step 1.
5. Compute the probability of the candidate according to Bayes rule by multiplying its probability by -^ where n is the number of times the candidate fails to predict some measurement outcome.
6. Go to 1.

This search may find more than the required number of diagnoses because the corrected probability of a best next candidate may be much lower than estimated. Although such candidates are diagnoses, they are not necessarily the leading ones.

Thus far we presumed that it is important to discriminate between all modes and that the d-partitions are the simple diagnoses. If it is not important to discriminate among certain modes, the preceding algorithms must be modified to identify d-partitions.

To identify d-partitions efficiently requires some subtle changes to the best-first search. Whenever a diagnosis is found, all the other candidates potentially in the same *d*-partition must be identified to fill out the d-partition and

accurately calculate its probability. However, this alone is insufficient to ensure that our previous algorithm finds the best (/-partitions because the probability of the best diagnoses within a d-partitions is only a lower bound on the probability of the (/-partition it could be part of. Therefore, we must modify the search such that each diagnoses is scored (only for the purposes of the search) by an upper bound of the probability of the d-partition it is part of. For each component, the 'probability' score assigned to each mode is the sum of the prior probabilities of that mode and all modes later in the mode ordering among which Sherlock is not required to discriminate. As a result of this ordering the 'probability' of a diagnosis is an upper-bound of the d-partition of which it is a member. Once one diagnosis of a d-partition is found, the remaining members of the d-partitions are filled out and its probability correctly computed. (Sherlock incorporates heuristics that avoid filling out the (/-partition with extremely low probability diagnoses.) As a result, Sherlock finds the leading d-partitions meeting the criteria for simple diagnoses just laid out.

6 A simple example

To demonstrate the basic ideas of Sherlock's operation with fault modes consider the three inverter example. We set the focussing thresholds as indicated earlier and the diagnostic objective to identify the mode of every component. However, as this example is tractable without using any focussing heuristics, we also show the correct values (i.e., having computed all the diagnoses) in parentheses. Suppose every inverter is modeled as described earlier, with *A* and *B* tending to fail stuck-at-1 and *C* tending to fail stuck-at-0:

A	*B*	*C*
$p(G(A)) = .99$	$p(G(B)) = .99$	$p(G(C)) = .99$
$p(S1(A)) = .008$	$p(S1(B)) = .008$	$p(S1(C)) = .001$
$p(S0(A)) = .001$	$p(S0(B)) = .001$	$p(S0(C)) = .008$
$p(U(A)) = .001$	$p(U(B)) = .001$	$p(U(C)) = .001$

With no observations Sherlock finds the single leading diagnosis (all diagnosis probabilities are normalized):

$$p[G(A), G(B), G(C)] = 1(.970).$$

The unfocussed Sherlock finds $4^3 = 64$ diagnoses (as there are no symptoms anything could be happening). Given a zero input (I = 0), Sherlock computes the following outcomes and their supporting environments. The final column indicates the additional environments an unfocussed Sherlock discovers):

$X = 0,$		$\{S0(A)\}$
$X = 1,$	$\{G(A)\}$	$\{S1(A)\}$
$Y = 0,$	$\{G(A), G(B)\}$	$\{S1(A), G(B)\}\{S0(B)\}$
$Y = 1,$		$\{S0(A), G(B)\}$ $\{S1(B)\}$
$O = 0,$		$\{S0(A), G(B), G(C)\}$
		$\{S1(B)G(C)\}$ $\{S0(C)\}$
$O = 1,$	$\{G(A), G(B), G(C)\}$	$\{S1(A), G(B), G(C)\}$
		$\{S0(B), G(C)\}$ $\{S1(C)\}$

The first line states that X = 0 under assumption 50(A), or equivalently that $X = 0$ in every candidate which includes *S0(A).* However, the focussed Sherlock finds no label for $X = 0$ as it does not hold in the single leading diagnosis. Intuitively, the last line states that the output is one if either (1) all the components are good (which is the leading diagnosis), (2) the first inverter is stuck-at-1 and the other two are good, (3) the second inverter is stuck-at-0, and the final inverter is good, or (4) the last inverter is stuck-at-1.

If we apply the minimum entropy technique we find costs ($ denotes the cost function, and the focussed Sherlock cost is shown first followed by the correct cost in parentheses):

$$\$(X) = 1(.99),\ \$(Y) = 1(.95),\ \$(O) = 1(.91).$$

All these costs are high because there is no evidence that a fault necessarily exists. The costs are all equal for the focussed Sherlock because there is only one leading diagnosis and therefore nothing can be learned by further measurement.

Suppose *O* is measured to be zero. There are four minimal conflicts (because each set of the minimal environments supporting *O* = 1 is now a conflict):

$$\{G(A), G(B), G(C)\}\ \ \{S1(A), G(B), G(C)\}$$
$$\{S0(B), G(C)\}\ \ \{S1(C)\}$$

Sherlock notices the leading candidate is eliminated and continues best-first search to find the following leading diagnoses:

$$p([G(A), G(B), S0(C)]) = 0.432(.426)$$
$$p([G(A), S1(B), G(C)]) = 0.432(.426)$$
$$p([S0(A), G(B), G(C)]) = 0.054(.053)$$
$$p([G(A), G(B), U(C)]) = 0.027(.027)$$
$$p([U(A), G(B), G(C)]) = 0.027(.027)$$
$$p([G(A), G(C), U(B)]) = 0.027(.027)$$
$$Total = .999(.986)$$

The next highest probability diagnoses being:

$$p([S1(A), G(B), S0(C)]) = .003(.003).$$

The figures in parentheses indicate the correct probabilities (initially there were 64 diagnoses, now 42 diagnoses remain) and Sherlock has identified the leading ones.

There are four important things to notice about this list of leading diagnoses. First, even though the Sherlock algorithm is running with *ki* = 5, it finds 6 candidates because it expands its search slightly to make sure candidates are not eliminated simply by round off errors. Second, the top 6 of the 42 candidates contain 98.6% of the probability mass. Third, the heuristic estimates are quite accurate - they are equal to the correct values normalized by .986. Fourth, although diagnosis *[S0(A),G(B),G(C)]* has the same prior probability of the three candidates *[G(A),G(B)yU(C)I* *[U(A),G(B),G(C)]* and *[G(A),G(C),U(B)],* after the two measurements its probability is twice that of the others. This is because the three candidates predict no value of the output and hence their posterior probability is reduced by one-half by Bayes rule.

Given the leading diagnoses, the resultant probabilities of behavior modes are:

	A	*B*	*C*
$p(G)$	.918(.911)	.541(.538)	.541(.538)
$p(S1)$	0(.007)	.432(.434)	0
$p(S0)$	.054(.054)	0(.0005)	.432(.435)
$p(U)$	.027(.028)	.027(.027)	.027(.027)

The table indicates that the major failure modes to consider are *C* stuck-at-0, and *B* stuck-at-1 and that all other faults are unlikely.

The resulting ATMS labels are (for this simple example focussing no longer has any affect on labels):

$X = 0, \quad \{G(B), G(C)\}\ \{S0(A)\}$
$X = 1, \quad \{G(A)\}\ \{S1(A)\}$
$Y = 0, \quad \{G(A), G(B)\}\ \{G(B), S1(A)\}\ \{S0(B)\}$
$Y = 1, \quad \{G(C)\}\ \{G(B), S0(A)\}\ \{S1(B)\}$
$\{S1(A), G(B), G(C)\}\{S0(B), G(C)\}\ \{S1(C)\}$

Suppose that we applied a second test vector with I=1 (the first test vector was I = *0),* and evaluated the hypothetical measurements:

$$\$(X_1) = .72(.72), \$(X_2) = .94(.92), \$(Y_1) = .31(.31)$$

$$\$(Y_2) = .91(.90), \$(O_2) = .89(.89).$$

Thus we see that measuring *Y* using the first test vector (I = 0) is the best measurement. This is because measuring *Y* will differentiate between the two high probability candidates. However, measuring *O* under the second test vector *(I* = 1) is useful as well. Suppose O = 0 under the second test vector. The resulting probabilities are:

$$p([G(A), G(B), S0(C)]) = 0.450(.444)$$
$$p([G(A), S1(B), G(C)]) = 0.450(.444)$$
$$p([S0(A), G(B), G(C)]) = 0.056(.056)$$
$$p([G(A), G(B), U(C)]) = 0.014(.014)$$
$$p([U(A), G(B), G(C)]) = 0.014(.014)$$
$$p([G(A), G(C), U(B)]) = 0.014(.014)$$

Although measuring O = 0 again does not eliminate any diagnosis, it provides further evidence that a component is not behaving in some unknown mode, thus slightly raising the probabilities of the first three diagnoses.

7 Acknowledgments

We appreciate discussion and comments from Daniel G. Bobrow, Brian Falkenhainer, Adam Farquhar, Alan Mackworth, Sanjay Mittal, Olivier Raimon, Mark Shirley and Peter Struss on the topics of this paper.

References

[1] Brown, J.S., Burton, R. R. and de Kleer, J., Pedagogical, natural language and knowledge engineering techniques in SOPHIE 1, II and III, in: D. Sleeman and J.S. Brown (Eds.), *Intelligent Tutoring Systems,* (Academic Press, New York, 1982) 227-282.

[2] Davis, R., Diagnostic Reasoning based on structure and behavior, *Artificial Intelligence* 24 (1984) 347-410.

[3] Davis, R., and Hamscher, W., Model-based reasoning: Troubleshooting, in *Exploring artificial intelligence,* edited by H.E. Shrobe and the American Association for Artificial Intelligence, (Morgan Kaufman, 1988), 297-346.

[4] de Kleer, J., Local methods of localizing faults in electronic circuits, Artificial Intelligence Laboratory, AIM-394, Cambridge: M.I.T., 1976.

[5] de Kleer, J., An assumption-based truth maintenance system, *Artificial Intelligence* 28 (1986) 127-162. Also in *Readings in NonMonotonic Reasoning,* edited by Matthew L. Ginsberg, (Morgan Kaufman, 1987), 280-297.

[6] de Kleer, J. and Williams, B.C., Diagnosing multiple faults, *Artificial Intelligence* 32 (1987) 97-130. Also in *Readings in NonMonotonic Reasoning,* edited by Matthew L. Ginsberg, (Morgan Kaufman, 1987), 372-388.

[7] de Kleer, J. and Williams, B.C., Diagnosis as identifying consistent modes of behavior, SSL Paper P88-00207, 1989.

[8] Dressier, O. and A. Farquhar, Problem solver control over the ATMS, Siemens Technical report INF 2 ARM-13-89, Munich, 1989.

[9] Forbus, K.D. and de Kleer, J., Focusing the ATMS, *Proceedings of the National Conference on Artificial Intelligence,* Saint Paul, MN (August 1988), 193-198.

[10] Genesereth, M.R., The use of design descriptions in automated diagnosis, *Artificial Intelligence* 24 (1984) 411-436.

[11] Hamscher, W.C., Model-based troubleshooting of digital systems, Artificial Intelligence Laboratory, TR-1074, Cambridge: M.I.T., 1988.

[12] Holtzblatt, L.J., "Diagnosing multiple failures using knowledge of component states," *IEEE Proceedings on AI applications,* 1988, 139-143.

[13] Raimon, O., Diagnosis as a trial: The alibi principle, IBM Scientific Center, 1989.

[14] Reiter, Raymond, A theory of diagnosis from first principles, *Artificial Intelligence* 32 (1987) 57-95. Also in *Readings in NonMonotonic Reasoning,* edited by Matthew L. Ginsberg, (Morgan Kaufman, 1987), 352-371.

[15] Struss, P., A framework for model-based diagnosis, Siemens TR INF2 ARM-10-88.

[16] Struss, P., and Dressier, O., "Physical negation" — Integrating fault models into the general diagnostic engine, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* Detroit, MI (August 1989).