From: AAAI-91 Proceedings. Copyright ©1991, AAAI (www.aaai.org). All rights reserved.

Characterizing Non-Intermittent Faults

Olivier Raiman, Johan de Kleer, Vijay Saraswat, Mark Shirley
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
raiman,dekleer,saraswat,shirley@parc.xerox.com

Abstract

A faulty component that behaves consistently over time is said to behave non-intermittently. For any given set of inputs, such a component will always generate the same output. Assuming that components fail non-intermittently is a common simplifying strategy used by diagnosticians, because (1) many real-world devices often fail this way, (2) this strategy removes the need to repeat experiments, and (3) this strategy allows information from independent examples of system behavior to be combined in relatively simple ways.

This paper extends the formal framework for diagnosis developed in [7, 12] to allow non-intermittency assumptions. In addition we show how the definitions can be easily integrated into ATMS-based diagnosis engines. Within our formulation, components can be individually assumed to be intermittent or non-intermittent.

Introduction

A central advantage of the model-based approach to diagnosis is that it can allow for unforeseen modes of failure. Early approaches achieved this by treating any deviation from correct behavior as faulty: each component is considered either good or in an "unknown" failure mode. The unknown mode makes no predictions about component behavior and is therefore consistent with any possible fault. As a consequence of using unknown modes, model-based diagnosis is extremely general and can be used for new components whose failure modes are not well-understood.

However, using only non-predictive failure modes can result in poor diagnostic discrimination. The essence of this problem is that non-predictive failure modes are consistent with any specific cause of failure and do not allow a diagnostic engine to distinguish between causes on the basis of, for instance, their likelihood. Using modes that predict misbehavior partially solves this problem [6, 14]. Thus, current methods use predictive fault modes to improve discrimination while retaining a non-predictive fault mode to maintain generality.

There remains a middle ground between predictive fault modes and unknown modes that has not been explored. This paper begins exploring this middle ground by considering non-intermittent faults, i.e., a fault mode where misbehavior is consistent over time. This mode makes a weak prediction about misbehavior: that the component output is a function of its inputs alone. It does not predict what that function is. These weak predictions improve diagnostic discrimination by combining evidence gathered over time.

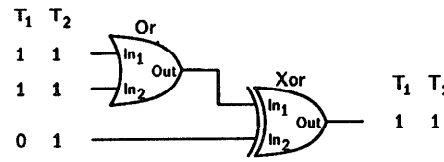

Figure 1: A Motivating Example

It is well-known that the notion of non-intermittency can be captured formally with an axiom stating that a component's outputs are a function of its inputs [1, 8, 10]. This paper builds on this observation in three ways: (1) we work out the details of incorporating non-intermittency axioms into the formal framework of [7, 12]; (2) we provide a very simple and efficient way to implement these axioms in ATMS-based diagnosis engines [3]; and (3) we examine the effect of non-intermittency axioms on the performance of our program, in particular, its ability to discriminate between competing explanations for component misbehavior.

In this paper we focus on devices whose components do not have internal state. In our formalization, components whose output depends on some internal state will be deemed intermittent. The definition can be extended to allow the output to be a function not just of the input but also a local state parameter, but such a discussion is beyond the scope of this paper.

A Motivating Example

Figure 1 shows a simple example where assuming non-intermittency improves diagnostic discrimination.

The circuit's inputs and outputs are marked with values observed at two different times T_1 and T_2. Note that at T_1, the circuit outputs a correct value and that at T_2 the circuit outputs an incorrect one. In this example, GDE, which models component behavior as either normal or unknown, implicates both components but cannot be more specific. However, by assuming the Or gate behaves non-intermittently, we can establish that the Xor gate is faulty as follows.

If Xor is good, then $In_1(Xor) = 1$ at T_1. This follows from $In_2(Xor) = 0$, $Out(Xor) = 1$ and the behavior of Xor. Similarly, if Xor is good, then $In_1(Xor) = 0$ at T_2. However, if Or behaves non-intermittently, then $In_1(Xor) = 1$ at T_2. This follows because Or has the same inputs at both T_1 and T_2 and must produce the same output. Thus we have a two contradictory predictions for the value of $In_1(Xor)$ at T_2. Either Xor is faulty or Or is behaving intermittently. Assuming non-intermittency means Xor is faulty.

Preliminaries

This section briefly summarizes the formal framework for diagnosis laid out in [7, 12], within which we will explore the formulation and consequences of non-intermittency.

Definition 1 *A diagnostic* system *is a triple* (SD, Comps, *Obs*) *where* SD, *the system description, is a set of (first-order) sentences,* Comps, *the system components, is a finite set of constants, and Obs, the set of observations is a set of (first-order) sentences.*

We use the term $In_i(x,t)$ or $Out_i(x,t)$ to indicate the value of the ith input or output of component x at observation-time t. If the component has only one input (output) we drop the subscript.[1] We adopt the modelling stance that a component is abnormal iff there is something physically wrong with it (or becomes so during the duration of diagnostic interest) which can cause it to produce incorrect results. Consequently, the key predicate Ab remains a one place predicate and does not depend on observation-time. The system description SD_0 of the circuit in Figure 1 can now be given in first-order predicate calculus (with equality) as follows.

The component library specifies the normal behavior of each type of primitive component, and also specifies the "ports" for each component.[2]

$$Org(c) \rightarrow (\neg Ab(c) \leftrightarrow \forall t.\ Out(c,t) = \mathbf{Or}(In_1(c,t), In_2(c,t))) \quad (1)$$

$$Xorg(c) \rightarrow (\neg Ab(c) \leftrightarrow \forall t.\ Out(c,t) = \mathbf{Xor}(In_1(c,t), In_2(c,t))) \quad (2)$$

$$Org(c) \rightarrow Port(Out(c,t)) \wedge Port(In_1(c,t)) \wedge Port(In_2(c,t)) \quad (3)$$

$$Xorg(c) \rightarrow Port(Out(c,t)) \wedge Port(In_1(c,t)) \wedge Port(In_2(c,t)) \quad (4)$$

In addition, we have a system-wide modelling assumption that each circuit is being analyzed as a boolean circuit. Hence every port can carry only boolean values:

$$Port(x) \rightarrow x = 0 \vee x = 1 \quad (5)$$

A model for a particular circuit is obtained by specifying the components of the circuit and their interconnections. In this case, we call the single or-gate in the circuit Or, and the single xor-gate Xor:

$$Org(Or) \quad (6)$$

$$Xorg(Xor) \quad (7)$$

$$Out(Or,t) = In_1(Xor,t) \quad (8)$$

The entire system description for this example, SD_0, is then given by Axioms 1–8.

The observations $Obs = Obs_1 \cup Obs_2$ are simply (for convenience we use Obs_i to refer to a subset of Obs pertaining to observation time T_i):

$$\begin{aligned} Obs_1 &= \{In_1(Or,T_1) = 1, In_2(Or,T_1) = 1, \\ &\quad In_2(Xor,T_1) = 0, Out(Xor,T_1) = 1\} \\ Obs_2 &= \{In_1(Or,T_2) = 1, In_2(Or,T_2) = 1, \\ &\quad In_2(Xor,T_2) = 1, Out(Xor,T_2) = 1\} \end{aligned} \quad (9)$$

As Ab remains a unary predicate, the conventional formalization of diagnosis applies without change:

Definition 2 *Given two sets of components C_p and C_n define $\mathcal{D}(C_p, C_n)$ to be the conjunction:*

$$\Big[\bigwedge_{c \in C_p} Ab(c)\Big] \wedge \Big[\bigwedge_{c \in C_n} \neg Ab(c)\Big].$$

A diagnosis *for* (SD, Comps, Obs) *is a formula* $\mathcal{D}(\Delta, \text{Comps} - \Delta)$ *(for* $\Delta \subseteq$ Comps*) such that* SD $\cup$ Obs $\cup \{\mathcal{D}(\Delta, \text{Comps} - \Delta)\}$ *is satisfiable.*

A classical approach to generating diagnoses is based on identifying conflicts, i.e., inconsistent beliefs about the state of the system. More precisely:

Definition 3 *Given a system* (SD,Comps,Obs)*, an* Ab-literal *is* $Ab(c)$ *or* $\neg Ab(c)$ *for some* $c \in$ Comps*. An* Ab-clause *is a clause consisting of Ab-literals. A* conflict *of the system* (SD,Comps,Obs) *is an Ab-clause entailed by* SD $\cup$ Obs*. A* minimal *conflict contains no other as a subclause.*

This strategy is based on the following theorem (proven in [7]) that states that diagnoses are fully characterized by the set of minimal conflicts.

Theorem 1 *Suppose* (SD,Comps,Obs) *is a system,* Π *is its set of minimal conflicts, and* $\Delta \subseteq$ Comps*. Then* $\mathcal{D}(\Delta, \text{Comps} - \Delta)$ *is a diagnosis iff* $\Pi \cup \{\mathcal{D}(\Delta, \text{Comps} - \Delta)\}$ *is satisfiable.*

[1] Throughout this paper we adopt the "reverse Prolog" convention: identifiers denote variables iff they begin with a lower-case letter. As usual, free variables in axioms are assumed universally quantified.

[2] The binary function-symbols **Or** and **Xor** are assumed to be interpreted by the usual Boolean or and xor functions.

With each Obs_i we can associate its set of conflicts $con(SD, Comps, Obs_i)$. [3] With suitably weak restrictions on the nature of SD and Obs, it is not hard to show that:

$$con(SD, Comps, \bigcup_{i\in I} Obs_i) \vdash \bigcup_{i\in I} con(SD, Comps, Obs_i) \quad (10)$$

In the absence of fault models it may not be possible to do better—that is, every conflict generated from $\bigcup_{i\in I} Obs_i$ would be generated from some particular Obs_i $(i \in I)$.

However, if components are non-intermittent, more conflicts could be generated due to interactions among observation-times, as we now show.

Defining Non-Intermittency

Definition 4 *A component behaves non-intermittently if its outputs are a function of its inputs.*

What is important about this definition is what the output is *not* a function of: namely, observation-time. Hence, the definition sanctions the following inference: if C is a non-intermittent component, and at some time T if an input test vector $\overline{X}$ is applied to the component, and the output Z is observed, then in any other observation T' if $\overline{X}$ is supplied as input, Z will be observed as output.

For each non-intermittent component we add an axiom stating its output is a function of its inputs and the component itself. For a component C with k input ports, and one output port the axiom is:

$$Ni(C) \leftrightarrow \forall t.Out(C,t) = F(C, In_1(C,t), \ldots, In_k(C,t)) \quad (11)$$

Here F is a function symbol on which no other constraints are imposed by SD: it represents the unspecified function which the component exhibits. By making the component an argument, we need only introduce one such function symbol.

The Example Revisited

For the *Or-Xor* example, the axioms added are:

$$Ni(Or) \leftrightarrow \forall t.Out(Or,t) = F(Or, In_1(Or,t), In_2(Or,t)) \quad (12)$$

$$Ni(Xor) \leftrightarrow \forall t.Out(Xor,t) = F(Xor, In_1(Xor,t), In_2(Xor,t)) \quad (13)$$

$$Ni(Or) \quad (14)$$

$$Ni(Xor) \quad (15)$$

Let the conjunction of these axioms be denoted by NI_0. We now establish that the system-description $SD_0 \cup NI_0$ contains enough information to conclude that the Xor gate is abnormal, given observations Obs_1 and Obs_2; formally:

$$SD_0, NI_0, Obs_1, Obs_2 \vdash Ab(Xor) \quad (16)$$

[3] Note that each conflict in $con(SD, Comps, Obs_i)$ is independent of observation-time, since all Ab-literals are independent of observation-time.

From Obs_1, the axiom for the xor-gate (2), the fact that Xor is an xor-gate (7), and the connection axiom (8) we get:

$$SD_0, Obs_1 \vdash \neg Ab(Xor) \rightarrow Out(Or, T_1) = 1 \quad (17)$$

From Obs_1, and the non-intermittency assumption for Or (12,14) we get:

$$SD_0, NI_0, Obs_1 \vdash F(Or, 1, 1) = Out(Or, T_1) \quad (18)$$

Taken together, we get:

$$SD_0, NI_0, Obs_1 \vdash \neg Ab(Xor) \rightarrow F(Or, 1, 1) = 1 \quad (19)$$

In exactly the same way, from the second observation we get:

$$SD_0, NI_0, Obs_2 \vdash \neg Ab(Xor) \rightarrow F(Or, 1, 1) = 0 \quad (20)$$

Thus there is a conflict—(19) and (20) together imply (16).

Implementation

Model-based diagnosis systems such as Sherlock and GDE [5, 6] which are based on constraint propagation with a TMS are easily extended to exploit the non-intermittency axioms. The basic issue is that the ATMS is essentially propositional[4]. In particular, it does not do any first-order inference, such as substituting a term in an expression. This is both a source of strength (efficiency) and weakness (first order incompleteness). Thus the basic inference rules built into the ATMS need to be extended in order to allow it to handle some of the first order logical apparatus we have used to model non-intermittency. We briefly describe the necessary changes in this section.

First let us review how ATMS-based diagnosis engines function. For every component c, the diagnosis engine creates the ATMS assumptions (propositions) $\boxed{Ab(c)}$ and $\boxed{\neg Ab(c)}$. Every port value is represented as an ATMS node. (In addition the node $\perp \stackrel{def}{=} \boxed{false}$ stands for the inconsistent proposition.) The ATMS manipulates propositional formulas of the form $e \rightarrow n$, where n is an ATMS node and e is a conjunction of ATMS assumptions. Every inconsistent set of assumptions e corresponds directly to a conflict.

To extend this framework to handle multiple observations, some nodes must be further parameterized by observation-time. Each observation instant T is encoded as an explicit ATMS assumption (a *time-token*), $\boxed{t = T}$. For every component X each port-observation $In_i(X,t) = Y$ (respectively, $Out_j(X,t) = Y$) is represented as an ATMS node $\boxed{In_i(X,t) = Y}$ (respectively, $\boxed{Out_j(X,t) = Y}$). An observation made at $t = T$

[4] We designate the ATMS node corresponding to a formula ϕ by $\boxed{\phi}$ to emphasize that the ATMS is purely propositional and does not examine the internal structure of the formula at each node.

is recorded as being contingent on $\boxed{t = T}$. For example, the observation $In_1(Or, T) = 1$ is encoded as $\boxed{t = T} \rightarrow \boxed{In_1(Or, t) = 1}$.

Given this encoding of observation time we need to add three classes of inference rules to the ATMS. The first class we add codifies the observation that the first-order formula $(t = T_i \wedge t = T_j)$ is inconsistent, if T_i and T_j are distinct constants. This is accomplished by adding for every pair of distinct observation instants T_i and T_j the (ATMS consumer) rule: $\boxed{t = T_i} \wedge \boxed{t = T_j} \rightarrow \bot$.

The second class of rules is concerned with making the first order inference: $\forall t.(t = T \rightarrow \phi) \rightarrow \phi$ where t does not occur free in ϕ. Let e be any ATMS label (conjunction of assumptions), and n be either $\bot$ or any ATMS node representing $Y = g(X_1, \ldots, X_k)$ for $Y, X_1, \ldots, X_k$ any constants, and g any function symbol. An ATMS consumer is installed to make the inference:

$$\frac{(\boxed{t = T} \wedge e) \rightarrow n}{e \rightarrow n} \tag{21}$$

In particular, for $n = \bot$, a nogood consumer [4] is installed: whenever a nogood is detected which mentions a time assumption, this assumption is removed and the resulting nogood asserted. For example, if the ATMS discovers the nogood $\boxed{t = T} \wedge \boxed{Ab(A)} \wedge \boxed{Ab(B)} \rightarrow \bot$ the nogood $\boxed{Ab(A)} \wedge \boxed{Ab(B)} \rightarrow \bot$ is asserted. (This rule is used implicitly in Sherlock [6].) The case where $n \neq \bot$ works together with the next rule class and is implemented by an environment consumer [4].

The third class of rules is concerned with allowing the substitution of equals for equals. Let $Y, X_1, \ldots, X_k$ be constants, and G an arbitrary function symbol. An ATMS consumer is installed to make the inference:

$$\frac{\begin{array}{l} e \rightarrow \boxed{y = Y} \\ \bigwedge_{i=1}^{k} e_i \rightarrow \boxed{x_i = X_i} \\ e' \rightarrow \boxed{y = G(x_1, \ldots, x_k)} \end{array}}{e \wedge (\bigwedge_{i=1}^{k} e_i) \wedge e' \rightarrow \boxed{Y = G(X_1, \ldots, X_k)}} \tag{22}$$

(Note that when such a consumer is invoked, it creates the node $\boxed{Y = G(X_1, \ldots, X_k)}$ if it does not already exist.)

The second two classes of rule are central to implementing non-intermittency. For every component having a non-intermittency function, we create an instance of the rule 22 with G being its non-intermittency function. If the component is non-intermittent, then e' is empty (i.e., true). (In the fuller version of the paper, we show that it is possible to distinguish among intermittent and non-intermittent faults by making the non-intermittence of a **component** an *assumption*, rather than an assertion. In such cases e' becomes non-empty.) The consumer we construct for each component triggers on finding a consistent set of assumptions which assigns values to every one of $y, x_1, \ldots, x_k$ and in which the non-intermittency function holds. In the simple case where e' is empty, the consumer will conclude that[5] $\boxed{t = T} \wedge \boxed{\mathcal{D}(C_p, C_n)} \rightarrow \boxed{Y = G(X_1, \ldots, X_k)}$. By an application of Rule 21, the assumption $\boxed{t = T}$ can be dropped, and the resultant formula is recorded as the ATMS justification:

$$\boxed{\mathcal{D}(C_p, C_n)} \wedge \boxed{x_1 = X_1} \wedge \cdots \wedge \boxed{x_k = X_k} \rightarrow \boxed{y = Y}.$$

Probability

The method for updating the probabilities of candidate diagnoses presented in [5, 6] easily extends to the situation where there are multiple observation times and a non-intermittency rule. Bayes' rule allows us to update the conditional probability of a candidate diagnosis C_l given a new piece of evidence E:

$$p(C_l|E) = \frac{p(E|C_l)p(C_l)}{p(E)}. \tag{23}$$

The denominator, $p(E)$ is just a normalization and is not critical to determine the relative probability rankings. $p(C_l)$ is either the prior or was computed as a result of the previous piece of evidence that was obtained. The central issue is the determination of $p(E|C_l)$. In the framework of [5, 6] every piece of evidence was simply an assertion that a particular circuit quantity had a particular value, i.e., $x_i = v_{ik}$. Now a piece of evidence is an assertion that a quantity has a value at a particular time, $x_i(t) = v_{ik}$. The conditional probability $p(E|C_l)$ is then evaluated as follows:

1. If $x_i(t) = v_{ik}$ is predicted by C_l given the preceding evidence, then $p(x_i(t) = v_{ik}|C_l) = 1$.
2. If $x_i(t) = v_{ik}$ is inconsistent with C_l and the preceding evidence, then $p(x_i(t) = v_{ik}|C_l) = 0$.
3. If $x_i(t) = v_{ik}$ is neither predicted by nor inconsistent with C_l and the preceding evidence, then we make the presupposition that every possible value for x_i is equally likely. Hence, $p(x_i(t) = v_{ik}|C_l) = \frac{1}{m}$ where m is the number of possible values x_i might have (in a conventional digital circuit $m = 2$).

Without non-intermittency, the behavior of abnormal components at different times is probabilistically independent. With the non-intermittency assumption, events occurring at one time condition events that may occur at other times. As a result, non-intermittency can change the probability of a component being faulty. For instance, allowing intermittent behavior in our motivating example (figure 1) and assuming components

[5] By $\boxed{\mathcal{D}(C_p, C_n)}$ we mean the conjunction $\left[\bigwedge_{c \in C_p} \boxed{Ab(c)}\right] \wedge \left[\bigwedge_{c \in C_n} \neg \boxed{Ab(c)}\right]$.

are equally unlikely to fail, we can conclude:

$$p(Ab(Xor)|\{Obs_1, Obs_2\}) = 0.5$$
$$p(Ab(Or)|\{Obs_1, Obs_2\}) = 0.5$$

With non-intermittency:

$$p(Ab(Xor)|\{Obs_1, Obs_2\}) = 1.0$$
$$p(Ab(Or)|\{Obs_1, Obs_2\}) < 0.5$$

In addition the method for selecting the best probe to make next (developed in [5, 6]) easily extends to the situation where there are multiple observation times and a non-intermittency rule. A probe is a measurement of a particular quantity at particular time. The best probe minimizes the expected entropy of the probability distribution of the candidate diagnoses.

Modeling

Incomplete models can cause components to appear intermittent. Within our framework any component behavior which deviates from the functional is considered to be intermittent. This can have some surprising consequences. Suppose, for example, two TTL buffers have their outputs tied together. If both buffers are supplied a 1 input, then the output will be 1 if both are behaving normally. But if the two inputs are 0 and 1, the output will be 0. (In TTL, a component driving 0 will overcome another driving one on the same node, hence this sort of configuration is called a wired-and function.

By our definition, one buffer will be interpreted as having an intermittent fault, because given an input of 1 its output is sometimes 0 and sometimes 1. This is a problem because the buffer isn't actually faulted. The difficulty is that the output of a TTL buffer is, in reality, not purely a function of its input but a function of the circuitry attached to its output and potentially all sorts of other quantities like temperature. As a consequence a component which appears intermittently faulted may, in fact, be perfectly correct.

The two buffer example arises out of a poor circuit model – the wired-and configuration was not modeled explicitly. A similar situation can arise via misassembly if, for instance, two buffers are supposed to be installed in series but the second one is erroneously installed in the reverse direction (or has a failure mode to this effect). Here a fault in the second buffer can cause the first to appear intermittently faulted (by our definition).

The moral of these examples is that non-intermittency assumptions should be defeasible, i.e, we should leave some room for intermittent behavior. One should always include some option that intermittent behavior is possible, because that mode will include, in effect, any situation where the modeling assumptions are violated. Thus our formulation of non-intermittency is a step towards reasoning about switching models (e.g., the bridge-fault example of [1]) in the formal framework we as a field are building up (e.g., [7, 12]).

A second moral is that what is considered intermittent behavior is a property of the level at which the system is modeled. If a component's behavior is a function of quantities (e.g., output load or temperature) not capturable at the given modeling level, any behavior which depends on such quantities will appear as intermittent behavior. This seems as it should be. Presumably all behavior is functional in practice, except we may not know what inputs (e.g., loads, temperature, alpha particles) actually depends on.

Empirical Results

In this section we report on several different experiments we performed in order to evaluate the utility of the non-intermittency rule. If the actual fault is non-intermittent and the diagnostic task involves more than one observation time, then assuming non-intermittency improves diagnostic precision. This improvement increases with the number of observation times. Note that the improvement in diagnostic precision is rarely as good as in the *Or–Xor* example. This should not be surprising: non-intermittency is a fairly weak assumption and therefore only provides dramatic advantage in some cases. Many of our experiments show only modest improvement in diagnostic precision which at first glance appears disappointing. But is is important to see this gain from the correct perspective. Although improvements in hardware and software will significantly improve the efficiency of diagnostic algorithm, no amount of hardware and software innovation will affect diagnostic precision much. Precision is only determined by our models and how much circuit-specific knowledge available. From a modeling point of view, presuming non-intermittency is an extremely simple, if not trivial change, but it yields a significant increase in diagnostic precision. In any particular task, the computational costs of the improved algorithm must be weighed against such costs as probing, and applying new inputs to the device.

Improvement in Diagnoses Eliminated

In this experiment we inserted single faults into an adder circuit, provided n randomly selected sets of observations (consisting of all input values and one possibly wrong output value) taking care that at least one set showed faulty behavior, and calculated the reduction in the number of incorrect diagnoses resulting from the non-intermittency rule. (An incorrect diagnosis is anything other than the fault we inserted.) We repeated this 100 times for each possible stuck-at fault and averaged the results. Total improvement (labeled total), the fraction of incorrect diagnoses eliminated, clearly increases with the number of observation sets. We also show how total improvement breaks down into the fraction of cases where there was some improvement (cases) and the improvement in just those cases (local).

Sets	1	2	3	4	5	6
Total	0	0.085	0.121	0.154	0.191	0.212
Cases	0	0.241	0.307	0.392	0.458	0.493
Local	0	0.353	0.394	0.392	0.417	0.429

Table 1: Fraction of diagnoses eliminated

Sets	1	2	3	4	5	6
A2	0	0.01	0.08	0.10	0.15	0.20
P3	0	0.04	0.06	0.09	0.10	0.24

Table 2: Fraction of probes eliminated.

Improvement in Number of Probes

This experiment measures the reduction in cost-of-diagnosis as measured by the number of probes needed to identify a fault. As before, we inserted stuck-at faults into the circuit and generated sets of observations randomly. Then the diagnostic engine probed the circuit until it achieved high confidence in a diagnosis (i.e., the information gain of the next probe measured by change in entropy was below 0.001). We did not allow the engine to select new inputs, but forced it to reuse one of the original sets of inputs for each probe. The experiment was performed with a small adder, A2, and a small parity circuit, P3. The total number of probes averaged 3.1; each entry in the table shows the fraction of these probes eliminated. Note that the improvement is low, although improvements such as this are significant if probing costs are high and computation is cheap.

Improvement in External Probes

This experiment is similar to the last. It shows a more dramatic improvement in the common case where probing inside the device is expensive but applying new inputs is cheap. This time, we inserted a stuck-at fault into the circuit and generated one set of observations that showed misbehavior. We restricted the diagnostic engine to probe circuit outputs but allowed it choose a set of inputs which maximized the probe's utility. The engine probed until it had correctly identified the fault or had identified a set of faults indistinguishable considering inputs and outputs alone. For the adder circuit, the engine averaged 10.0 probes without the non-intermittency rule and 3.6 probes with it. In addition, the sets of indistinguishable faults produced using non-intermittency are sometimes smaller. In this case, therefore, assuming non-intermittency simultaneously reduces the number of probes and improves diagnostic precision.

Conclusion

This paper has presented a formal framework for describing non-intermittent behavior, shown a simple way to extend ATMS-based diagnosis engines to incorporate this, and presented empirical evidence showing positive benefit. Our experiments have shown the result of many trials with small devices. We have tried larger devices and observed even larger benefit, however we are still working to gather statistically significant data.

Acknowledgments

Conversations with Daniel G. Bobrow, Brian Falkenhainer and Brian Williams helped clarify many of these concepts.

References

[1] Davis, R., Diagnostic Reasoning Based on Structure and Behavior, *Artificial Intelligence* **24** (1984) 347–410.

[2] 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.

[3] 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 Kaufmann, 1987), 280–297.

[4] de Kleer, J., Problem solving with the ATMS, *Artificial Intelligence* **28** (1986) 197–224.

[5] 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.

[6] de Kleer, J. and Williams, B.C., Diagnosis with behavioral modes, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 104–109.

[7] de Kleer, J., Mackworth A., and Reiter R., Characterizing Diagnoses and Systems, in: *Proceedings AAAI-90*, Boston, MA (1990).

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

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

[10] Poole, D.L., Default Reasoning and Diagnosis as Theory Formation, Technical Report CS-86-02, University of Waterloo.

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

[12] Reiter, R., A theory of diagnosis from first principles, *Artificial Intelligence* **32** (1987) 57–95. Also in *Readings in Non-Monotonic Reasoning*, edited by Matthew L. Ginsberg, (Morgan Kaufmann, 1987), 352–371.

[13] Struss, P., and Dressler, O., "Physical negation" — Integrating fault models into the general diagnostic engine, in: *Proceedings IJCAI-89* Detroit, MI (1989) 1318-1323.