A Theory of Diagnosis from First Principles

Raymond Reiter
Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4; The Canadian Institute for Advanced Research

Recommended by Johan de Kleer and Daniel G. Bobrow

ABSTRACT

Suppose one is given a description of a system, together with an observation of the system's behaviour which conflicts with the way the system is meant to behave. The diagnostic problem is to determine those components of the system which, when assumed to be functioning abnormally, will explain the discrepancy between the observed and correct system behaviour.

We propose a general theory for this problem. The theory requires only that the system be described in a suitable logic. Moreover, there are many such suitable logics, e.g. first-order, temporal, dynamic, etc. As a result, the theory accommodates diagnostic reasoning in a wide variety of practical settings, including digital and analogue circuits, medicine, and database updates. The theory leads to an algorithm for computing all diagnoses, and to various results concerning principles of measurement for discriminating among competing diagnoses. Finally, the theory reveals close connections between diagnostic reasoning and nonmonotonic reasoning.

1. Introduction

In the theory and design of diagnostic reasoning systems there appear to be two quite different approaches in the literature.

In the first approach, often referred to as diagnosis from first principles, one begins with a description of some system—a physical device or real world setting of interest, say—together with an observation of the system's behaviour. If this observation conflicts with the way the system is meant to behave, one is confronted with a diagnostic problem, namely, to determine those system components which, when assumed to be functioning abnormally, will explain the discrepancy between the observed and correct system behaviour. For solving this diagnostic problem from first principles, the only available information is the system description, i.e. its design or structure, together with the observation(s) of the system behaviour. In particular, no

Artificial Intelligence **32** (1987) 57–95

heuristic information about system failures is available, for example, of the kind "When the system exhibits such and such aberrant behaviour, then in 90% of these cases, such and such components have failed." Notable examples of approaches to diagnostic reasoning from first principles are [4–7, 15, 16].

Under the second approach to diagnostic reasoning, which might be described as the experiential approach, heuristic information plays a dominant role. The corresponding diagnostic reasoning systems attempt to codify the rules of thumb, statistical intuitions, and past experience of human diagnosticians considered experts in some particular task domain. The structure or design of the corresponding real world system being diagnosed is only weakly represented, if at all. Successful diagnoses stem from the codified experience of the human expert being modeled, rather than from what is often referred to as "deep" knowledge of the system being diagnosed. A notable example of such an approach to diagnosis from experience is the MYCIN system [3].

As one will gather from its title, the current paper deals exclusively with the problem of diagnosis from first principles. Without in any way denying the importance of expert experience in diagnostic reasoning, we believe that a precise theoretical foundation for diagnosis from first principles will be a necessary ingredient in any general theory of diagnostic reasoning. The purpose of this paper is to provide such a theoretical foundation for diagnosis from first principles. Our theory primarily builds upon, and generalizes, the work of de Kleer [5] and Genesereth [7].

We begin by abstractly defining the concept of a system of interacting components. Initially, we choose first-order logic as a language for representing such systems, but as we shall eventually see, many different logics will lead to the same theory of diagnosis presented in this paper. Whatever one's choice of representation logic, the description within it of a system will specify how that system normally behaves on the assumption that all its components are functioning correctly. If we have available an observation of the system's actual behaviour and if this observation conflicts with (i.e. is logically inconsistent with) the way the system is meant to behave, then we have a diagnostic problem. The problem is to determine those system components which, when assumed to be functioning abnormally, will explain the discrepancy between the observed and correct system behaviour. These intuitions, coupled with our appeal to a logical system representation language, will allow us in Section 2 to formally define the concept of a diagnosis, including multiple fault diagnoses. Diagnoses need not be unique; there may be several competing explanations for the same faulty system.

The computational problem, then, is to determine all possible diagnoses for a given faulty system. After proving some preliminary results in Section 3, we derive an "algorithm"[1] in Section 4 for computing all diagnoses for a given faulty system. This algorithm has a number of virtues, not the least of which is its relative independence of the particular logic representing the system being diagnosed. By "relative independence" here we mean that the algorithm assumes the availability of a sound and complete theorem prover for the logic being used, but in all other respects is unconcerned with the underlying logic. A nice consequence of this decomposition is that special purpose theorem provers can be designed for particular diagnostic applications, for example, Boolean equation solvers for switching circuits. Such a special purpose theorem prover can then "hook into" the general purpose algorithm to yield a domain specific diagnostic algorithm.

As we remarked above, multiple, competing diagnoses can arise for a given faulty system. The normal approach to discriminating among competing diagnoses is to make system measurements, for example inserting probes into a circuit, or performing laboratory tests on a patient. In Section 5 we prove a variety of results about the conclusions which can legitimately be drawn from the results of certain system measurements.

Diagnostic reasoning turns out to be a form of nonmonotonic reasoning. In Section 6 we explore this connection, and show how the theory of diagnosis of this paper is related to default logic [17].

In Section 7 we consider the relationship of our theory of diagnosis to other research in this area. Finally, we summarize what we take to be the principal contributions of this work in Section 8.

2. Problem Formulation

2.1. Systems

We seek a very general theory of diagnosis, one which will account for diagnostic reasoning in a wide variety of task domains such as medicine, digital and analogue circuits, etc. To achieve the necessary generality, we appeal to first-order logic with equality as a language for representing task specific information.[2] Also in the interest of generality, we define the domain-independent concept of a system which is designed to formalize as abstractly as possible the concept of a component, and the concept of a collection of interacting components.

Definition 2.1. A *system* is a pair (SD, COMPONENTS) where:
(1) SD, the *system description*, is a set of first-order sentences;
(2) COMPONENTS, the *system components*, is a finite set of constants.

[1] The reason for the scare quotes will become evident later.

[2] Actually, all of the results of this paper continue to hold for a wide variety of logics, not just first-order. However, in order to provide a concrete development of the theory, we shall initially appeal only to first-order logic. In Section 6.1, we shall indicate how the results so obtained generalize to other logics.

In all intended applications, the system description will mention a distinguished unary predicate AB($\cdot$), interpreted to mean "abnormal."

Example 2.2. Figure 1 depicts the binary full adder used extensively by Genesereth [7] as an example. This adder may be represented by a system with components $\{A_1, A_2, X_1, X_2, O_1\}$ and the following system description:

$$\text{ANDG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{and}(\text{in1}(x), \text{in2}(x)),$$
$$\text{XORG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{xor}(\text{in1}(x), \text{in2}(x)),$$
$$\text{ORG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{or}(\text{in1}(x), \text{in2}(x)),$$
$$\text{ANDG}(A_1), \quad \text{ANDG}(A_2),$$
$$\text{XORG}(X_1), \quad \text{XORG}(X_2) \quad \text{ORG}(O_1),$$
$$\text{out}(X_1) = \text{in2}(A_2),$$
$$\text{out}(X_1) = \text{in1}(X_2),$$
$$\text{out}(A_2) = \text{in1}(O_1),$$
$$\text{in1}(A_2) = \text{in2}(X_2),$$
$$\text{in1}(X_1) = \text{in1}(A_1),$$
$$\text{in2}(X_1) = \text{in2}(A_1),$$
$$\text{out}(A_1) = \text{in2}(O_1).$$

In addition, the system description contains axioms specifying that the circuit inputs are binary valued:

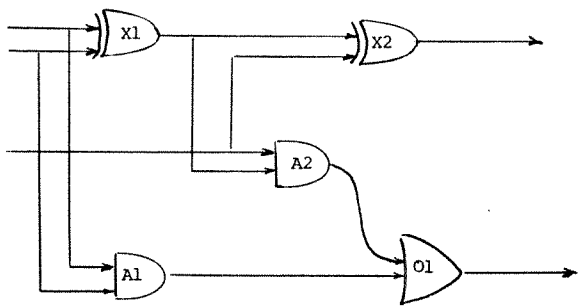

FIG. 1. A full adder. A_1 and A_2 are and gates; X_1 and X_2 are exclusive-or gates; O_1 is an or gate.

$$\text{in1}(X_1) = 0 \vee \text{in1}(X_1) = 1,$$
$$\text{in2}(X_1) = 0 \vee \text{in2}(X_1) = 1,$$
$$\text{in1}(A_1) = 0 \vee \text{in1}(A_1) = 1.$$

Finally there are axioms for a Boolean algebra over $\{0, 1\}$, which we do not specify here.

Typically, a system description describes how the system components *normally* behave by appealing to the distinguished predicate AB whose intended meaning is "abnormal." Thus, the first axiom in the example system description states that a normal (i.e. not ABnormal) and gate's output is the Boolean and function of its two inputs. Many other kinds of component descriptions are possible, e.g. "Normally an adult human's heart rate is between 70 and 90 beats per minute."

$$\text{ADULT}(x) \wedge \text{HEART-OF}(x, h) \wedge \neg\text{AB}(h) \supset \text{rate}(h) \geq 70 \wedge \text{rate}(h) \leq 90.$$

"Normally, if the voltage across a zener diode is positive and less than its breakdown voltage, the current through it must be zero."

$$\text{ZENER-DIODE}(z) \wedge \neg\text{AB}(z) \wedge$$
$$\text{voltage}(z) > 0 \wedge \text{voltage}(z) < \text{break-voltage}(z)$$
$$\supset \text{current}(z) = 0.$$

We can represent the fact that a fault in component c_1 will cause a fault in component c_2:

$$\text{AB}(c_1) \supset \text{AB}(c_2).$$

If we know all the ways components of a certain type can be faulted, we can express this by an axiom of the form:

$$\text{TYPE}(x) \wedge \text{AB}(x) \supset \text{FAULT}_1(x) \vee \cdots \vee \text{FAULT}_n(x).$$

By introducing several kinds of AB predicates, we can represent more general component properties, e.g. "Normally, a faulty resistor is either open or shorted."

$$\text{RESISTOR}(r) \wedge \text{AB}(r) \wedge \neg\text{AB}'(r)$$
$$\supset \text{OPEN}(r) \vee \text{SHORTED}(r).$$

The use of an AB predicate for system descriptions is borrowed from McCarthy [11] who exploits such a predicate in conjunction with his formalization of circumscription to account for various patterns of nonmonotonic common-sense reasoning. As we shall see in Section 6.2, this seemingly tenuous connection with nonmonotonic reasoning is in fact fundamental. Diagnosis provides an important example of nonmonotonic reasoning.

2.2. Observations of systems

Real world diagnostic settings involve observations. Without observations, we have no way of determining whether something is wrong and hence whether a diagnosis is called for.

Definition 2.3. An *observation* of a system is a finite set of first-order sentences. We shall write (SD, COMPONENTS, OBS) for a system (SD, COMPONENTS) with observation OBS.

Example 2.2 (continued). Suppose a physical full adder is given the inputs 1, 0, 1 and it outputs 1, 0 in response. Then this observation can be represented by:

$$\text{in1}(X_1) = 1,$$
$$\text{in2}(X_1) = 0,$$
$$\text{in1}(A_2) = 1,$$
$$\text{out}(X_2) = 1,$$
$$\text{out}(O_1) = 0.$$

Notice that this observation indicates that the physical circuit is faulty; both circuit outputs are wrong for the given inputs.

Notice also that distinguished inputs and outputs are features of digital circuits (and many man-made artifacts) not of the general theory we are proposing.

2.3. Diagnoses

Suppose we have determined that a system (SD, $\{c_1, \ldots, c_n\}$) is faulty, by which we mean informally that we have made an observation OBS which conflicts with what the system description predicts should happen if all its components were behaving correctly. Now $\{\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)\}$ represents the assumption that all system components are behaving correctly, so that $\text{SD} \cup \{\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)\}$ represents the system behaviour on the assumption that all its components are working properly. Hence the fact that the observation OBS conflicts with what the system should do were all its components behaving correctly can be formalized by:

$$\text{SD} \cup \{\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)\} \cup \text{OBS} \tag{2.1}$$

is inconsistent.

Intuitively, a diagnosis is a *conjecture* that certain of the components are faulty (ABnormal) and the rest normal. The problem is to specify which components we conjecture to be faulty. Now our objective is to explain the inconsistency (2.1), an inconsistency which stems from the assumptions $\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)$, i.e. that all components are behaving correctly. The natural way to explain this inconsistency is to retract enough of the assumptions $\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)$, so as to restore consistency to (2.1). But we should not be overzealous in this; retracting all of $\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_n)$ will restore consistency to (2.1), corresponding to the diagnosis that all components are faulty. We should appeal to:

The Principle of Parsimony. A diagnosis is a conjecture that some minimal set of components are faulty.

This leads us to the following:

Definition 2.4. A *diagnosis* for (SD, COMPONENTS, OBS) is a minimal set $\Delta \subseteq$ COMPONENTS such that

$$\text{SD} \cup \text{OBS} \cup \{\text{AB}(c) \mid c \in \Delta\} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent.

In other words, a diagnosis is determined by a smallest set of components with the following property: The assumption that each of these components is faulty (ABnormal), together with the assumption that all other components are behaving correctly (not ABnormal), is consistent with the system description and the observation.

Example 2.2 (continued). For the full adder, there are three diagnoses: $\{X_1\}$, $\{X_2, O_1\}$, $\{X_2, A_2\}$.

2.4. Computing diagnoses: Decidability

The definition of a diagnosis appeals to a consistency test for arbitrary first-order formulae. Since there is no decision procedure for determining the consistency of first-order formulae, we cannot hope to compute diagnoses in the most general case. Nevertheless, there are many practical settings where consistency is decidable, hence diagnoses are computable.

For example, in the case of switching circuits like that of the full adder, it is sufficient, for the purpose of computing diagnoses, to determine whether a system of Boolean equations is consistent, i.e. has a solution, and this is decidable. Similarly, in the case of linear electronic circuits, we need only have the capacity to determine whether a system of linear equations has a solution. As we shall see in Section 7, at least one established model for medical diagnosis leads to a computable theory. The point is: we should not allow the undecidability of the general problem to prevent us from developing a theory of diagnosis because there are many practical settings in which the theory does provide effective computations.

This means that for any given application it will be necessary first to establish decidability of its diagnostic problem. If the problem turns out to be undecidable, heuristic techniques will be necessary. It is an interesting question to characterize classes of systems whose diagnostic problems are decidable, but we shall not pursue that question in this paper.

3. Some Consequences of the Definition

The first two results are simple consequences of the definition of a diagnosis (Definition 2.4); we omit their proofs.

Proposition 3.1. *A diagnosis exists for* (SD, COMPONENTS, OBS) *iff* SD ∪ OBS *is consistent.*

Proposition 3.2. { } *is a diagnosis* (*and the only diagnosis*) *for* (SD, COMPONENTS, OBS) *iff*

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS}\}$$

is consistent, i.e. iff the observation does not conflict with what the system should do if all its components were behaving correctly.

This is as it should be; we observe nothing wrong, so there is no reason to conjecture a faulty component.

Proposition 3.3. *If* Δ *is a diagnosis for* (SD, COMPONENTS, OBS), *then for each* $c_i \in \Delta$,

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \models \text{AB}(c_i).$$

Proof. If Δ is the empty set, then the result is true vacuously. Suppose then that $\Delta = \{c_1, \ldots, c_k\}$, and that the proposition is false, so that

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$
$$\cup \{\neg \text{AB}(c_1) \vee \cdots \vee \neg \text{AB}(c_k)\}$$

is consistent. Now $\neg \text{AB}(c_1) \vee \cdots \vee \neg \text{AB}(c_k)$ is logically equivalent to

$$\bigvee [\text{AB}(c_1)^{i_1} \wedge \cdots \wedge \text{AB}(c_k)^{i_k}]$$

where the disjunction is over all $i_1, \ldots, i_k \in \{0, 1\}$ such that at least one $i_j = 0$, and where

$$\text{AB}(c_j)^{i_j} = \begin{cases} \text{AB}(c_j) & , \text{ if } i_j = 1 , \\ \neg \text{AB}(c_j) & , \text{ if } i_j = 0 . \end{cases}$$

So we have that

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$
$$\cup \{\bigvee [\text{AB}(c_1)^{i_1} \wedge \cdots \wedge \text{AB}(c_k)^{i_k}]\}$$

is consistent, in which case, for some choice of $i_1, \ldots, i_k \in \{0, 1\}$ with at least one $i_j = 0$, we have that

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$
$$\cup \{\text{AB}(c_1)^{i_1} \wedge \cdots \wedge \text{AB}(c_k)^{i_k}\}$$

is consistent. But this says that Δ has a strict subset Δ' with the property that

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$
$$\cup \{\text{AB}(c) \mid c \in \Delta\}$$

is consistent, contradicting the fact that Δ is a diagnosis for (SD, COMPONENTS, OBS). □

Proposition 3.3 is rather interesting. It says that the faulty components Δ are logically determined by the normal components COMPONENTS − Δ.

The next result provides a simpler characterization of a diagnosis than does the original Definition 2.4.

Proposition 3.4. $\Delta \subseteq$ COMPONENTS *is a diagnosis for* (SD, COMPONENTS, OBS) *iff* Δ *is a minimal set such that*

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent.

Proof. ($\Rightarrow$) since Δ is a diagnosis,

$$\text{SD} \cup \text{OBS} \cup \{\text{AB}(c) \mid c \in \Delta\}$$
$$\cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent, so that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent. Moreover, by Proposition 3.3, for each $c_i \in \Delta$

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \cup \{\neg\text{AB}(c_i)\}$$

is inconsistent. The result now follows.

($\Leftarrow$) By the minimality of Δ, we must have, for each $c_i \in \Delta$, that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \cup \{\neg\text{AB}(c_i)\}$$

is inconsistent, i.e. for each $c_i \in \Delta$,

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \models \text{AB}(c_i)\,.$$

Moreover, by hypothesis,

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent. Hence

$$\text{SD} \cup \text{OBS} \cup \{\text{AB}(c) \mid c \in \Delta\}$$
$$\cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent. It remains only to show that Δ is a minimal set with this property in order to establish that Δ is a diagnosis. But this is easy, for if Δ had a strict subset Δ' with this property, then

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta'\}$$

would be consistent, contradicting the hypothesis of this proposition. □

4. Computing Diagnoses

Our objective in this section is to show how to determine all diagnoses for (SD, COMPONENTS, OBS). There is a direct generate-and-test mechanism based upon Proposition 3.4: Systematically generate subsets Δ of COMPONENTS, generating Δs with minimal cardinality first, and test the consistency of

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}\,.$$

The obvious problem with this approach is that it is too inefficient for systems with large numbers of components. Instead, we propose a method based upon a suitable formalization of the concept of a conflict set, a concept due originally to de Kleer [5].

4.1. Conflict sets and diagnoses

Definition 4.1. A *conflict set* for (SD, COMPONENTS, OBS) is a set $\{c_1, \ldots, c_k\} \subseteq$ COMPONENTS such that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_k)\}$$

is inconsistent.

A conflict set for (SD, COMPONENTS, OBS) is *minimal* iff no proper subset of it is a conflict set for (SD, COMPONENTS, OBS).

Proposition 3.4 can be reformulated in terms of conflict sets as follows:

Proposition 4.2. $\Delta \subseteq$ COMPONENTS *is a diagnosis for* (SD, COMPONENTS, OBS) *iff* Δ *is a minimal set such that* COMPONENTS $- \Delta$ *is not a conflict set for* (SD, COMPONENTS, OBS).

Definition 4.3. Suppose C is a collection of sets. A *hitting set* for C is a set $H \subseteq \bigcup_{S \in C} S$ such that $H \cap S \neq \{\ \}$ for each $S \in C$. A hitting set for C is *minimal* iff no proper subset of it is a hitting set for C.

The following is our principal characterization of diagnoses, and will provide the basis for computing diagnoses:

Theorem 4.4. $\Delta \subseteq$ COMPONENTS *is a diagnosis for* (SD, COMPONENTS, OBS) *iff* Δ *is a minimal hitting set for the collection of conflict sets for* (SD, COMPONENTS, OBS).

Proof. ($\Rightarrow$) By Proposition 4.2, COMPONENTS $- \Delta$ is not a conflict set for (SD, COMPONENTS, OBS). Hence, every conflict set contains an element of Δ, so that Δ is a hitting set for the collection of conflict sets for (SD, COMPONENTS, OBS). We

must prove Δ is a minimal such hitting set. Now by Proposition 4.2, Δ is a minimal set such that COMPONENTS $- \Delta$ is not a conflict set. This means for each $c \in \Delta$ that $\{c\} \cup$ (COMPONENTS $- \Delta$) is a conflict set. From this it follows that Δ is a minimal hitting set for the conflict sets for (SD, COMPONENTS, OBS).

($\Leftarrow$) We use Proposition 4.2 to prove that Δ is a diagnosis for (SD, COMPONENTS, OBS) by showing that:

(1) COMPONENTS $- \Delta$ is not a conflict set for (SD, COMPONENTS, OBS),

(2) Δ is a minimal set with property (1) by proving, for each $c \in \Delta$, that $\{c\} \cup$ (COMPONENTS $- \Delta$) is a conflict set for (SD, COMPONENTS, OBS).

Proof of (1): If, on the contrary, COMPONENTS $- \Delta$ were a conflict set, then Δ would not hit it, contradicting the fact that Δ is a hitting set for all conflict sets.

Proof of (2): Every conflict set has the form $\Delta' \cup K$ where $\Delta' \subseteq \Delta$ and $K \subseteq$ COMPONENTS $- \Delta$. Moreover, for each $c \in \Delta$, some conflict set must contain c, for otherwise Δ would not be a minimal hitting set. We prove that some conflict set containing c is of the form $\{c\} \cup K$. For if not, then every conflict set containing c must have the form $\{c, c', \ldots\} \cup K$ where $c' \in \Delta$ and $c' \neq c$. But then $\Delta - \{c\}$ is a smaller hitting set than Δ, a contradiction. Hence, for each $c \in \Delta$ there is a conflict set of the form $\{c\} \cup K$ where $K \subseteq$ COMPONENTS $- \Delta$. But then $\{c\} \cup$ (COMPONENTS $- \Delta$) is also a conflict set. □

Notice that every superset of a conflict set for (SD, COMPONENTS, OBS) is also a conflict set. Because of this, we can easily prove the following:

> H is a minimal hitting set for the collection of all conflict sets for (SD, COMPONENTS, OBS) iff H is a minimal hitting set for the collection of all minimal conflict sets for (SD, COMPONENTS, OBS).

Combining this result with Theorem 4.4 we obtain an alternative characterization of diagnoses:

Corollary 4.5. $\Delta \subseteq$ COMPONENTS *is a diagnosis for* (SD, COMPONENTS, OBS) *iff* Δ *is a minimal hitting set for the collection of minimal conflict sets for* (SD, COMPONENTS, OBS).

Example 2.2 (continued). The full adder has two minimal conflict sets $\{X_1, X_2\}$ and $\{X_1, A_2, O_1\}$ corresponding, respectively, to the inconsistency of

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(X_1), \neg\text{AB}(X_2)\}$$

and

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(X_1), \neg\text{AB}(A_2), \neg\text{AB}(O_1)\}\ .$$

There are three diagnoses, given by the minimal hitting sets for $\{X_1, X_2\}$ and $\{X_1, A_2, O_1\}$: $\{X_1\}$, $\{X_2, A_2\}$, $\{X_2, O_1\}$.

De Kleer and Williams [6] have independently proposed a characterization of diagnoses which corresponds to our Corollary 4.5. However, the major difference between their result and ours is that, while theirs derives from sound intuitions, it is based upon an unformalized approach to diagnosis, while our results have been derived from initial formal definitions.

4.2. Computing hitting sets

Our approach to computing diagnoses is based upon Theorem 4.4 and therefore requires computing all minimal hitting sets for the collection of conflict sets for (SD, COMPONENTS, OBS). Accordingly, in this section, we focus on computing the minimal hitting sets for an arbitrary collection of sets. The approach we shall propose will be particularly appropriate in a diagnostic setting.

Definition 4.6. Suppose F is a collection of sets. An edge-labeled and node-labeled tree T is an *HS-tree* for F iff it is a smallest tree with the following properties:

(1) Its root is labeled by "$\surd$" if F is empty. Otherwise, its root is labeled by a set of F.

(2) If n is a node of T, define $H(n)$ to be the set of edge labels on the path in T from the root node to n. If n is labeled by $\surd$, it has no successor nodes in T. If n is labeled by a set Σ of F, then for each $\sigma \in \Sigma$, n has a successor node n_σ joined to n by an edge labeled by σ. The label for n_σ is a set $S \in F$ such that $S \cap H(n_\sigma) = \{\ \}$ if such a set S exists. Otherwise, n_σ is labeled by $\surd$.

Example 4.7. Figure 2 is an *HS*-tree for $F = \{\{2,4,5\}, \{1,2,3\}, \{1,3,5\}, \{2,4,6\}, \{2,4\}, \{2,3,5\}, \{1,6\}\}$.

The following results are obvious for any *HS*-tree for a collection F of sets:

(1) If n is a node of the tree labeled by $\surd$, then $H(n)$ is a hitting set for F.

(2) Each minimal hitting set for F is $H(n)$ for some node n of the tree labeled by $\surd$.

Notice that the sets of the form $H(n)$ for nodes labeled by $\surd$ do not include all hitting sets for F. The important point for our purpose is that they do include all minimal hitting sets for F. Our objective is to determine various tree pruning techniques to allow us to generate as small a subtree of an *HS*-tree as is possible, while preserving the property that the subtree so generated will give us all minimal hitting sets for F. In addition, we wish to minimize the number of accesses to F required to generate this subtree, where by an access to F we mean the computation required to determine the label of a node in this subtree. Such a computation of a label for node n is determined (at least conceptually) by searching F for a set S such that $S \cap H(n) = \{\ \}$. If such an S

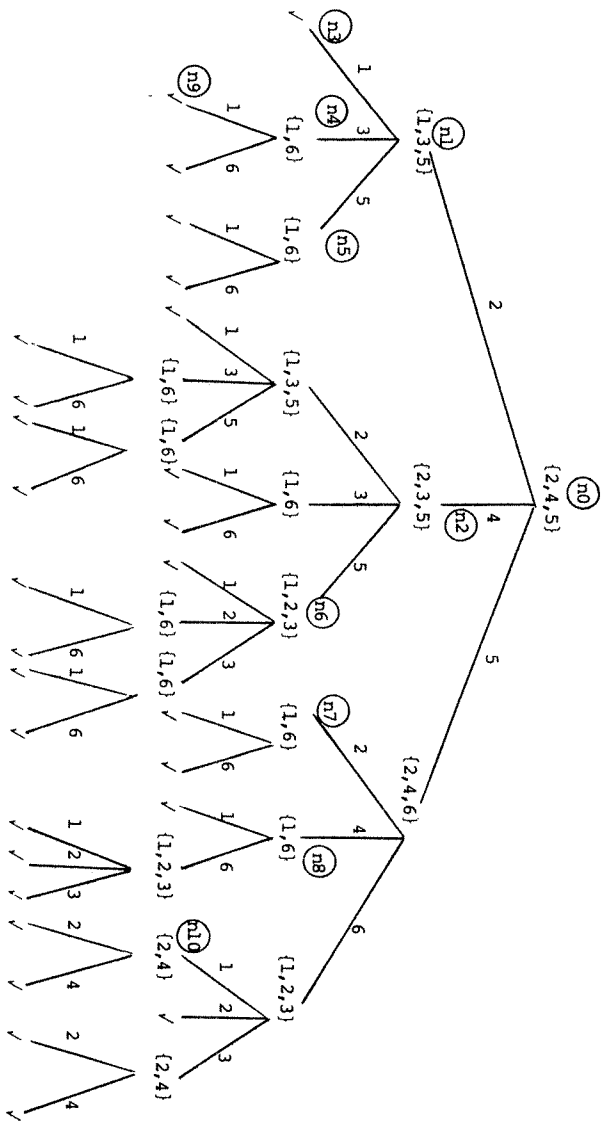

FIG. 2. An *HS*-tree for $F = \{\{2,4,5\}, \{1,2,3\}, \{1,3,5\}, \{2,4,6\}, \{2,4\}, \{2,3,5\}, \{1,6\}\}$.

is found, node n is labeled by S, else it is labeled by $\surd$. For our purposes, this computation requiring an access to F must be treated as extremely expensive. This is so because for us, F will be the set of all conflict sets for (SD, COMPONENTS, OBS). Moreover, F will not be explicitly available, but will instead be implicitly defined. An access to F will be the computation of a conflict set, and this will require a call to a theorem prover. Clearly, we will want as few such accesses to F as possible.

The natural way to reduce accesses to F in generating an *HS*-tree is to reuse node labels which have already been computed. For example, if the *HS*-tree of Fig. 2 was generated breadth-first, generating nodes at any fixed level in the tree in left-to-right order, then node n_2 could have been assigned the same label as n_1, namely, $\{1,3,5\}$, since $H(n_2) \cap \{1,3,5\} = \{\ \}$. For the same reason, all of the nodes labeled $\{1,6\}$ other than node n_4 require no access to F; their labels can be determined from the tree itself as the previously computed label for n_4.

Next, we consider three tree pruning devices for *HS*-trees which preserve the property that the resulting pruned *HS*-tree will include all minimal hitting sets for F.

(1) Notice that in Fig. 2 $H(n_6) = H(n_8)$. Moreover, we could have reused the label of n_6 for n_8. This means that the subtrees rooted at n_6 and n_8 respectively could be identically generated had we chosen the reused label for n_8. Thus, n_8's subtree is redundant, and we can close node n_8. Similarly, $H(n_7) = H(n_5)$ so we can close node n_7.

(2) In Fig. 2, $H(n_3) = \{1,2\}$ is a hitting set for F. Therefore, any other node n of the tree for which $H(n_3) \subseteq H(n)$ cannot possibly define a smaller hitting set than $H(n_3)$. Since we are only interested in minimal hitting sets, such a node n can be closed. In Fig. 2, node n_9 is an example of such a node which we can close. The computational advantage of recognizing that node n_9 can be closed is that we need not access F to determine that n_9's label is $\surd$.

(3) The following is a simple result about minimal hitting sets: If F is a collection of sets, and if $S \in F$ and $S' \in F$ with S a proper subset of S', then $F - \{S'\}$ has the same minimal hitting sets as F.

We can use this result to prune the *HS*-tree of Fig. 2. Notice that the label $\{2,4\}$ of node n_{10} is a proper subset of $\{2,4,5\}$, the label of the previously generated node n_0. This means that, in generating the label of n_{10} we have discovered that F contains a strict subset $\{2,4\}$ of $\{2,4,5\}$, another set of F. Thus, in generating the *HS*-tree, we could have labeled n_0 by the smaller set $\{2,4\}$, instead of $\{2,4,5\}$. In other words, the edge from n_0 labeled 5 and the entire subtree beneath this edge are redundant; they can be removed from the tree while preserving the property that the resulting pruned tree will yield all minimal hitting sets.

Notice that this tree pruning device appears unnecessarily clumsy. We waited until node n_{10} was generated and labeled by $\{2,4\}$ before noticing that F therefore contains a set $\{2,4\}$ which is a proper subset of another set $\{2,4,5\}$

of F. Why not simply prescan F, remove from F all supersets of sets in F, and use the resulting trimmed F to generate an *HS*-tree? In the example of Fig. 2 we could first have removed $\{2, 4, 5\}$ and $\{2, 4, 6\}$ from F before generating its *HS*-tree. The reason we did not do this is, as we have already remarked, for our purposes F will be *implicitly* defined as the set of all conflict sets for (SD, COMPONENTS, OBS). Since we will not have available an explicit enumeration of these conflict sets, we cannot perform a preliminary subset test on them.

We summarize our method for generating a pruned *HS*-tree for F as follows:

(1) Generate the *HS*-tree breadth-first, generating nodes at any fixed level in the tree in left-to-right order.

(2) Reusing node labels: If node n is labeled by the set $S \in F$, and if n' is a node such that $H(n') \cap S = \{\ \}$, label n' by S. (We indicate that the label of n' is a reused label by underlining it in the tree.) Such a node n' requires no access to F.

(3) Tree pruning:

(i) If node n is labeled by $\surd$ and node n' is such that $H(n) \subseteq H(n')$, close n', i.e. do not compute a label for n'; do not generate any successors of n'.

(ii) If node n has been generated and node n' is such that $H(n') = H(n)$, then close n'. (We indicate a closed node in the tree by marking it with "×".)

(iii) If nodes n and n' have been respectively labeled by sets S and S' of F, and if S' is a proper subset of S, then for each $\alpha \in S - S'$ mark as redundant the edge from node n labeled by α. A redundant edge, together with the subtree beneath it, may be removed from the *HS*-tree while preserving the property that the resulting pruned *HS*-tree will yield all minimal hitting sets for F. (We indicate a redundant edge in a pruned *HS*-tree by cutting it with ")(".)

Figure 3 depicts such a pruned *HS*-tree for the example of Fig. 2.

In view of the preceding discussion, the following result should be clear:

Theorem 4.8. *Let F be a collection of sets, and T a pruned HS-tree for F, as previously described. Then $\{H(n) \mid n$ is a node of T labeled by $\surd\}$ is the collection of minimal hitting sets for F.*

Example 4.7 (continued). For the set F of Fig. 3, the minimal hitting sets are:

$$\{1, 2\}, \quad \{2, 3, 6\}, \quad \{2, 5, 6\}, \quad \{4, 1, 3\}, \quad \{4, 1, 5\}, \quad \{4, 3, 6\}.$$

The computation of these hitting sets required 13 accesses to F.

4.3. Computing all diagnoses

A conceptually simple approach to computing diagnoses can be based upon Theorems 4.4 and 4.8 as follows: First compute the collection F of all conflict

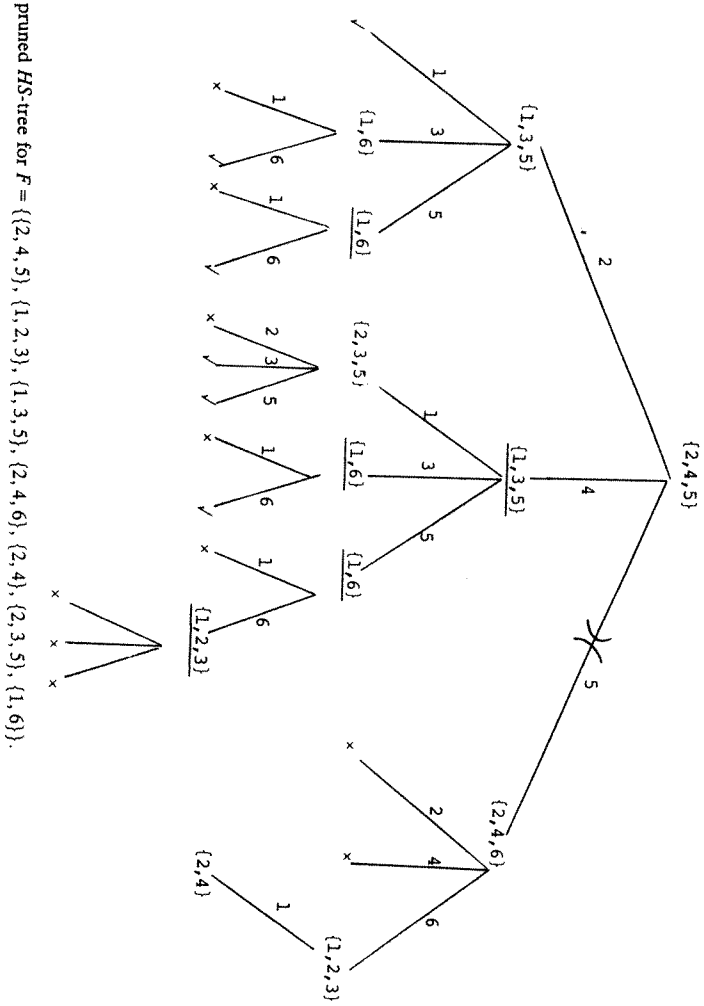

Fig. 3. A pruned *HS*-tree for $F = \{\{2, 4, 5\}, \{1, 2, 3\}, \{1, 3, 5\}, \{2, 4, 6\}, \{2, 4\}, \{2, 3, 5\}, \{1, 6\}\}$.

sets for (SD, COMPONENTS, OBS), then use the method of pruned *HS*-trees to compute the minimal hitting sets for F. These minimal hitting sets will be the diagnoses.

The problem, then, is to systematically compute all conflict sets for (SD, COMPONENTS, OBS). Recall that $\{c_1, \ldots, c_k\} \subseteq$ COMPONENTS is a conflict set iff SD $\cup$ OBS $\cup \{\neg AB(c_1), \ldots, \neg AB(c_k)\}$ is inconsistent. Now if SD $\cup$ OBS $\cup \{\neg AB(c_1), \ldots, \neg AB(c_k)\}$ is inconsistent, so is SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$. So, using a sound and complete theorem prover, compute all refutations of SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$ and for each such refutation, record the AB instances entering into the refutation. If $\{\neg AB(c_1), \ldots, \neg AB(c_k)\}$ is the set of AB instances used in such a refutation, then $\{c_1, \ldots, c_k\}$ is a conflict set.

For example, Fig. 4 gives a stylized resolution style refutation tree for SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$ in which the AB instances entering into the refutation are explicitly indicated. This refutation yields the conflict set $\{c_1, c_5, c_7\}$.

Therefore, one approach to computing all conflict sets for (SD, COMPONENTS, OBS) is to invoke a sound and complete theorem prover which computes all refutations of SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$, and which, for each such refutation, records the AB instances entering into the refutation in order to determine the corresponding conflict set.

Unfortunately, there is a serious problem with this approach: the conflict sets do not stand in a 1-1 relationship with the refutations of SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$. There will be refutations which are inessential variants of each other. Figure 5 illustrates two resolution refutations which, although different refutations, involve the same AB instances. A refutation-based approach to the computation of conflict sets ought not compute such

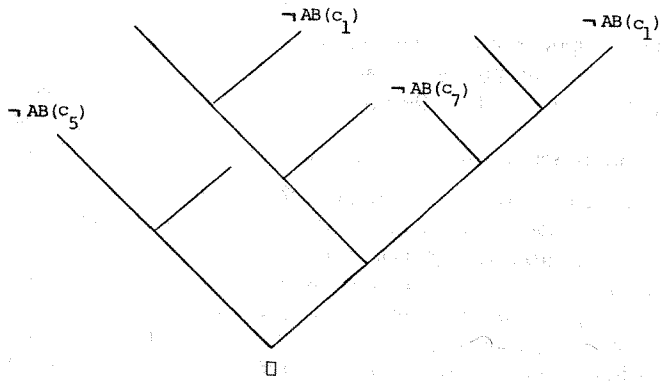

FIG. 4. Resolution style refutation tree for SD $\cup$ OBS $\cup \{\neg AB(c) \mid c \in$ COMPONENTS$\}$.

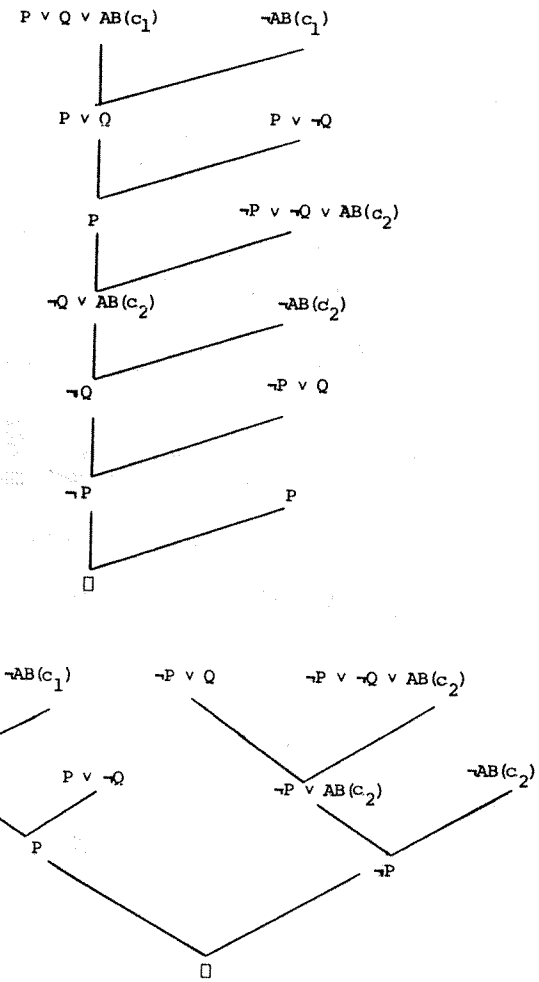

FIG. 5. Two resolution refutations involving the same AB instances.

inessential variants. If we were relying on a resolution theorem prover for computing refutations, then fixing on some particular resolution strategy (e.g. linear resolution, resolution with literal ordering, etc.) might help with this problem. Depending upon a particular style of theorem prover would not be a good idea, however. We should not restrict the underlying theorem proving system since particular applications might benefit from special purpose theorem provers, e.g. constraint propagation techniques for diagnosis, as used by Davis

[4] and de Kleer and Williams [6], or specialized Boolean equation solvers. So our problem is to prevent the computation of inessential variants of refutations, without imposing any constraints on the nature of the underlying theorem proving system. As we shall see, our algorithm for computing minimal hitting sets handles this problem very nicely. In fact, the underlying theorem prover is relieved of all responsibility for the systematic generation of all conflict sets; this responsibility for determining the order in which conflict sets are computed, and when they have all been determined, is assumed by our algorithm for computing minimal hitting sets. The role of the theorem prover is simply to return a suitable conflict set when so requested by the algorithm for generating pruned *HS*-trees.

We now develop our "algorithm"[3] for computing all diagnoses for (SD, COMPONENTS, OBS). Our approach is based upon Theorem 4.4 and therefore requires all minimal hitting sets for the collection F of conflict sets for (SD, COMPONENTS, OBS). The minimal hitting set calculation will involve generating a pruned *HS*-tree for F, as per Theorem 4.8, but with one significant difference: F will not be given explicitly. Instead, suitable elements of F will be computed, as required, while the *HS*-tree is being generated.

Recall that in generating a pruned *HS*-tree for a collection, F, of sets, a node n of the tree can be assigned a label in one of two ways:

(1) By reusing a label S previously determined for some other node n' whenever $H(n) \cap S = \{\ \}$; in this case, no access to F is required since n's label is obtained from that part of the pruned *HS*-tree generated thus far.

(2) By searching F for a set S such that $H(n) \cap S = \{\ \}$. If such a set S can be found in F, n is labeled by S, otherwise by $\surd$. In this case the set F must be accessed; n's label cannot be determined without F.

Now it should be clear that the set F need not be given explicitly. The only time that F is needed is in case (2) above. Therefore, to generate a pruned *HS*-tree for F, we only require a function which, when given $H(n)$, returns a set S such that $H(n) \cap S = \{\ \}$ if such a set S exists in F, and $\surd$ otherwise. We now exhibit such a function when F is the collection of conflict sets for (SD, COMPONENTS, OBS). Let TP(SD COMPONENTS, OBS) be a function with the property that whenever (SD, COMPONENTS) is a system and OBS an observation for that system, TP(SD, COMPONENTS, OBS) returns a conflict set for (SD, COMPONENTS, OBS) if one exists, i.e. if SD $\cup$ OBS $\cup$ $\{\neg$AB$(c) \mid c \in$ COMPONENTS$\}$ is inconsistent, and returns $\surd$ otherwise. It is easy to see that any such function TP has the following property: If $C \subseteq$ COMPONENTS, then TP(SD, COMPONENTS $- C$, OBS) returns a conflict set S for (SD, COMPONENTS, OBS) such that $C \cap S = \{\ \}$ if such a set S exists, and $\surd$ otherwise. It follows that we can generate a pruned *HS*-tree for F, the collection of conflict sets for (SD, COMPONENTS, OBS) as described in Section 4.2 except that whenever a node n of this tree needs an access to F to compute its label, we label n by TP(SD, COMPONENTS $- H(n)$, OBS). From this pruned *HS*-tree T we can extract the set of all minimal hitting sets for F, namely $\{H(n) \mid n$ is a node of T labeled by $\surd\}$. By Theorem 4.4, this is the set of diagnoses for (SD, COMPONENTS, OBS).

We have proved the correctness of the following "algorithm":

Algorithm. DIAGNOSE(SD, COMPONENTS, OBS).

{COMMENT: (SD, COMPONENTS) is a system and OBS is an observation of the system. TP is any function with the property that TP(SD, COMPONENTS, OBS) returns a conflict set for (SD, COMPONENTS, OBS) if one exists, i.e. if SD $\cup$ OBS $\cup$ $\{\neg$AB$(c) \mid c \in$ COMPONENTS$\}$ is inconsistent, and returns $\surd$ otherwise. DIAGNOSE(SD, COMPONENTS, OBS) returns the set of all diagnoses for (SD, COMPONENTS, OBS).}

Step 1. Generate a pruned *HS*-tree T for the collection F of conflict sets for (SD, COMPONENTS, OBS) as described in Section 4.2 except that whenever, in the process of generating T a node n of T needs an access to F to compute its label, label that node by TP(SD, COMPONENTS $- H(n)$, OBS).

Step 2. Return $\{H(n) \mid n$ is a node of T labeled by $\surd\}$.

Example 2.2 (continued). *The full adder.* Figure 6 shows a possible pruned *HS*-tree for the full adder example, as computed by DIAGNOSE(SD, $\{X_1, X_2, A_1, A_2, O_1\}$, OBS) where SD and OBS are the system description and observation described earlier for the full adder. Recall that $\{X_1, X_2, A_1, A_2, O_1\}$ are the components of this system. The root node of Fig. 6 is labeled by a call to TP(SD, $\{X_1, X_2, A_1, A_2, O_1\}$, OBS) which we are supposing returns $\{X_1, X_2\}$. Node n_1 is labeled by a call to TP(SD, $\{X_2, A_1, A_2, O_1\}$, OBS). Since SD $\cup$ OBS $\cup$ $\{\neg$AB$(X_2), \neg$AB$(A_1), \neg$AB$(A_2), \neg$AB$(O_1)\}$ is consistent, this call returns $\surd$. Node n_2 is labeled by a call to TP(SD, $\{X_1, A_1, A_2, O_1\}$, OBS) which we are supposing returns $\{X_1, A_2, O_1\}$. Node n_3 is marked closed by an *HS*-tree pruning rule. Node n_4 is labeled by a call to TP(SD, $\{X_1, A_1, O_1\}$, OBS) which returns $\surd$ since SD $\cup$ OBS $\cup$ $\{\neg$AB(X_1),

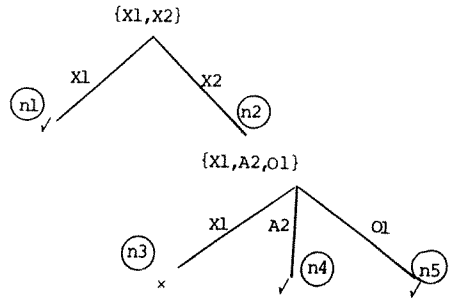

FIG. 6. Computing all diagnoses for the full adder.

[3] The scare quotes serve as a reminder that the general problem is undecidable.

$\neg$AB(A_1), $\neg$AB(O_1)} is consistent. Similarly, node n_5 is labeled $\surd$ by a call to TP(SD, $\{X_1, A_1, A_2\}$, OBS). The set of all diagnoses can now be read from the tree of Fig. 6: $\{\{X_1\}, \{X_2, A_2\}, \{X_2, O_1\}\}$. Five calls to TP were required. Figure 6 is, of course, not the only possible computation of the diagnoses for the full adder. The particular trees one obtains depend upon what the function TP returns. Figure 7 shows a different possible pruned *HS*-tree for the full adder, corresponding to a different function TP. Notice that in this case the root node is labeled by a nonminimal conflict set returned by TP(SD, $\{X_1, X_2, A_1, A_2, O_1\}$, OBS). Notice also that the *HS*-tree pruning algorithm marks one of the edges (labeled A_1) redundant after TP returns a strict subset $\{X_1, A_2, O_1\}$ of the roots node's label. Six calls to TP were required for this example.

We make several remarks about algorithm DIAGNOSE:

(1) No two calls by DIAGNOSE to TP will ever return the same conflict set. This a simple consequence of the way node labels are determined in generating a pruned *HS*-tree. As a result, the theorem prover underlying TP need not compute the same conflict set in two inessentially different ways, as was the case for example in Fig. 5. Moreover, for any two calls by DIAGNOSE to TP, the later call will never return a superset of the earlier call. A consequence of this is that normally DIAGNOSE will explicitly compute only a small subset of all possible conflict sets for (SD, COMPONENTS, OBS). For example, in Fig. 6, DIAGNOSE computes only two of the possible 12 conflict sets, while in Fig. 7 it computes three. This is important because the computation of a conflict set requires an expensive call to a theorem prover.

(2) The function TP may be realized computationally in many different ways. One way, as we remarked earlier, is to use a complete refutation based theorem prover which records the AB instances entering into the refutations it computes. Another way to compute a conflict set is to use a theorem prover to directly derive, from SD $\cup$ OBS as premises, a disjunction of AB instances, i.e. a formula of the form AB(c_1) $\vee \cdots \vee$ AB(c_k), for then, since SD $\cup$ OBS $\vdash$ AB(c_1) $\vee \cdots \vee$ AB(c_k), we have SD $\cup$ OBS $\cup$ {$\neg$AB(c_1), . . . , $\neg$AB(c_k)} inconsistent, whence $\{c_1, \ldots, c_k\}$ is a conflict set. This appears to be the basis for computing suspects used by Genesereth's DART program [7].

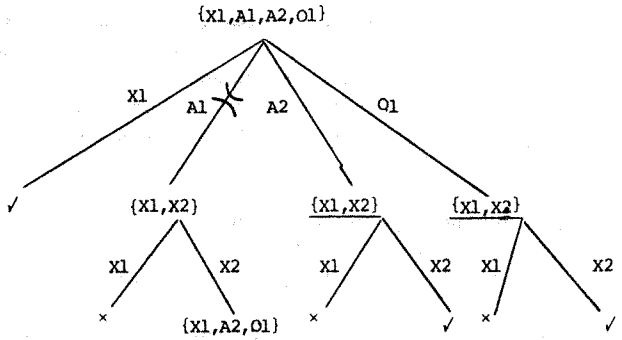

FIG. 7. A different computation of diagnoses for the full adder.

In particular applications TP might profitably be realized by special purpose theorem provers, e.g. constraint propagation techniques for solving systems of equations, as used by Davis [4] and de Kleer and Williams [6].

Whatever the theorem proving techniques used by TP, it should probably be implemented in such a way that intermediate computations obtained while computing a conflict set are cached for possible use in subsequent calls to TP.

(3) If the function TP can be realized so that it returns only minimal conflict sets, then in the generation of a pruned *HS*-tree, no edge will ever be marked redundant so that in this circumstance we can simplify the tree generation algorithm.

(4) Because pruned *HS*-trees are generated breadth-first, diagnoses are computed in order of increasing cardinality. Thus, all of the diagnoses involving just a single component are determined by those nodes labeled by $\surd$ at level 1[4] in the tree, and these are computed before the level-2 nodes which determine the diagnoses involving two components, etc. If, for some reason, we believe that diagnoses of cardinality greater than k are highly improbable, or if we are interested only in diagnoses of cardinality k or less, then DIAGNOSE can stop growing the *HS*-tree at level k.

Example 4.9. Figure 8 illustrates a device first introduced by Davis [4], and subsequently extensively analyzed by de Kleer and Williams [6]. The device has 5 components, M_1, M_2, M_3, A_1, and A_2. The observation is given by:

$$\text{in1}(M_1) = 3, \quad \text{in2}(M_1) = 2, \quad \text{in1}(M_2) = 3, \quad \text{in2}(M_2) = 2,$$
$$\text{in1}(M_3) = 3, \quad \text{in2}(M_3) = 2, \quad \text{out}(A_1) = 10, \quad \text{out}(A_2) = 12.$$

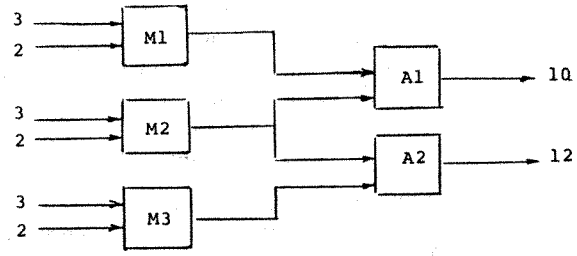

FIG. 8. A device with observed inputs and outputs. M_1, M_2, and M_3 are multipliers; A_1 and A_2 are adders.

[4] The root node is at level 0.

We omit a full specification of the system description; it will involve axioms specifying how the components normally behave, together with axioms about addition and multiplication of integers.

For example, the normal behaviour of a multiplier will be specified by:

$$\text{MULTIPLIER}(m) \wedge \neg \text{AB}(m) \supset \text{out}(m) = \text{in1}(m) * \text{in2}(m) .$$

The system description will also contain axioms describing how the components are interconnected, e.g.

$$\text{out}(M_1) = \text{in1}(A_1) , \quad \text{out}(M_2) = \text{in2}(A_1) ,$$
$$\text{out}(M_2) = \text{in1}(A_2) , \quad \text{etc.}$$

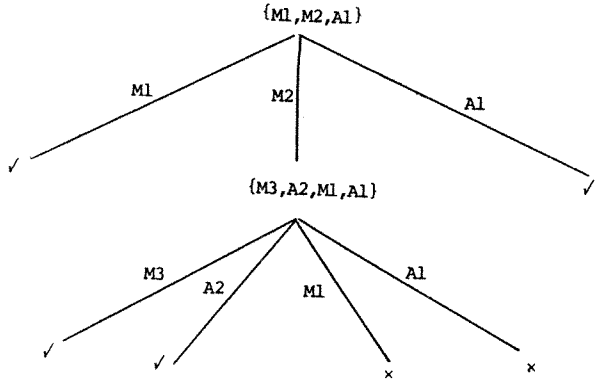

FIG. 9. A pruned *HS*-tree for Example 4.9.

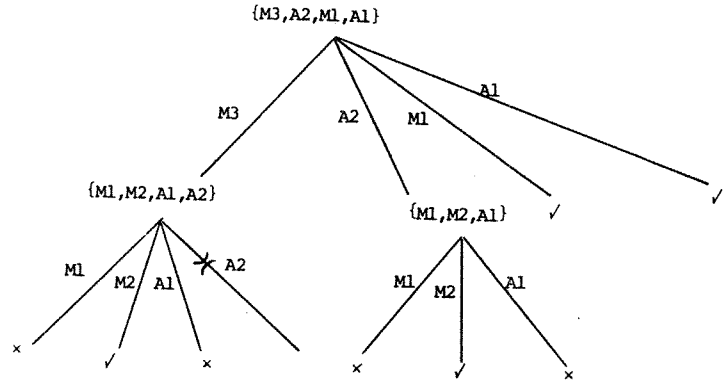

FIG. 10. A pruned *HS*-tree for Example 4.9.

Since the observation $\text{out}(A_1) = 10$ conflicts with the predicted value $\text{out}(A_1) = 12$ the device is faulty. Figures 9 and 10 give two possible pruned *HS*-trees which algorithm DIAGNOSE might compute. From either of these we obtain the four diagnoses for this device: $\{M_1\}$, $\{A_1\}$, $\{M_2, M_3\}$, $\{A_2, M_2\}$.

4.4. Single fault diagnoses

A diagnosis is a *single fault* diagnosis iff it is a singleton. If it contains two or more components, it is a *multiple fault* diagnosis. For the full adder example, there is one single fault diagnosis, $\{X_1\}$, and two multiple fault diagnoses, $\{X_2, A_2\}$ and $\{X_2, O_1\}$.

Single fault diagnoses are of particular interest, primarily because one normally expects components to fail independently of each another. As a result, single fault diagnoses are judged more likely to be correct than any of their companion multiple fault diagnoses. Thus, in the case of the full adder, the single fault diagnosis $\{X_1\}$ is to be preferred over the other two multiple fault diagnoses.

Theorem 4.4 (Corollary 4.5) provides the following characterization of single fault diagnoses:

Corollary 4.10. *$\{c\}$ is a single fault diagnosis for* (SD, COMPONENTS, OBS) *iff c is an element of every (minimal) conflict set for* (SD, COMPONENTS, OBS).

If our concern is only to compute all single fault diagnoses, we can do so by allowing algorithm DIAGNOSE to generate a pruned *HS*-tree only to level 1 of the tree, returning $H(n)$ for each level-1 node labeled by $\surd$. In fact, the following result is a simple consequence of the correctness of algorithm DIAGNOSE.

Theorem 4.11 (Determining all single fault diagnoses from one conflict set). *Suppose C is a conflict set for* (SD, COMPONENTS, OBS). *Then $\{c\}$ is a diagnosis for* (SD, COMPONENTS, OBS) *iff $c \in C$ and* SD $\cup$ OBS $\cup \{\neg \text{AB}(k) \mid k \in \text{COMPONENTS} - \{c\}\}$ *is consistent.*

Theorem 4.11 generalizes in the natural way to the case where we have several, but not necessarily all, conflict sets.

Theorem 4.12 (Determining all single fault diagnoses from several conflict sets). *Suppose for $n \geq 1$ that $C_1, C_2, \ldots, C_n$ are conflict sets for* (SD, COMPONENTS, OBS), *and that $C = \bigcap_{i=1}^{n} C_i$. Then $\{c\}$ is a diagnosis for* (SD, COMPONENTS, OBS) *iff $c \in C$ and* SD $\cup$ OBS $\cup \{\neg \text{AB}(k) \mid k \in \text{COMPONENTS} - \{c\}\}$ *is consistent.*

Proof. ($\Rightarrow$) By Corollary 4.10, $c \in C$. The rest follows by Proposition 3.4.

($\Leftarrow$) Since (SD, COMPONENTS, OBS) has a conflict set, then SD $\cup$ OBS $\cup$ $\{\neg AB(k) \mid k \in$ COMPONENTS$\}$ is inconsistent. Since SD $\cup$ OBS $\cup$ $\{\neg AB(k) \mid k \in$ COMPONENTS $- \{c\}\}$ is consistent, then by Proposition 3.4, $\{c\}$ is a diagnosis for (SD, COMPONENTS, OBS). □

Theorem 4.12 is a generalization of the candidate generation procedure of Davis [4], and provides a formal justification for Davis' procedure. Davis' concern was the determination of single fault diagnoses for digital circuits represented by constraint networks. His candidate generation procedure computes some (not necessarily all) conflict sets, intersects these to obtain a set C of possible single fault candidates, then for each $c \in C$ performs a candidate consistency test by suspending (turning off) the constraint modeling c's behaviour. This consistency test via the suspension of c's constraint corresponds to the consistency test, called for by Theorem 4.12, of SD $\cup$ OBS $\cup$ $\{\neg AB(k) \mid k \in$ COMPONENTS $- \{c\}\}$. The exclusion of c from COMPONENTS amounts to "turning off" component c while performing the consistency test.

5. Measurements

Suppose that (SD, COMPONENTS, OBS) has more than one diagnosis. Without further information about the system, one cannot conjecture a unique diagnosis. One way to obtain further information about a system is to perform measurements of some kind e.g. insert a probe into a circuit, or perform a laboratory test on a patient. In this section, we study how such measurements can affect diagnoses. Specifically, we shall define a *measurement* MEAS to be an additional observation (and therefore a finite set of first-order sentences), and we shall consider the following question: What is the relationship between the diagnoses for (SD, COMPONENTS, OBS) and (SD, COMPONENTS, OBS $\cup$ MEAS)?

Typically, OBS will be an initial observation of the system (SD, COMPONENTS), leading to multiple diagnoses, and MEAS will be a measurement (an additional observation) of the system taken in an attempt to discriminate among the original multiple diagnoses.

Definition 5.1. A diagnosis Δ for (SD, COMPONENTS, OBS) *predicts* Π (a first-order sentence) iff

$$\text{SD} \cup \text{OBS} \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in \text{COMPONENTS} - \Delta\} \models \Pi,$$

i.e. on the assumption that the components of Δ are all faulty, and the remaining components are all functioning normally, system behaviour Π must hold.

Example 5.2. For the device of Fig. 8 (Example 4.9), diagnosis $\{M_1\}$ predicts $\text{out}(M_2) = 6$ and $\text{out}(M_1) = 4$; diagnosis $\{M_2, M_3\}$ predicts $\text{out}(M_2) = 4$ and $\text{out}(M_3) = 8$.

Proposition 3.3 immediately provides a simpler version of Definition 5.1.

Proposition 5.3. *A diagnosis* Δ *for* (SD, COMPONENTS, OBS) *predicts* Π *iff*

$$\text{SD} \cup \text{OBS} \cup \{\neg AB(c) \mid c \in \text{COMPONENTS} - \Delta\} \models \Pi .$$

The following two results are immediate consequences of Definition 2.4.

Proposition 5.4. *If no diagnosis for* (SD, COMPONENTS, OBS) *predicts* $\neg\Pi$, *then* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$) *has the same diagnoses as* (SD, COMPONENTS, OBS).

In other words, a measurement which disconfirms no diagnosis provides no new information.

Proposition 5.5.

(1) *Every diagnosis for* (SD, COMPONENTS, OBS) *which predicts* Π *is a diagnosis for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$), *i.e. diagnoses are preserved under confirming measurements.*

(2) *No diagnosis for* (SD, COMPONENTS, OBS) *which predicts* $\neg\Pi$ *is a diagnosis for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$), *i.e. a measurement rejects the diagnoses which it disconfirms.*

A simple consequence of Proposition 5.5 is that whenever each diagnosis for (SD, COMPONENTS, OBS) predicts one of Π, $\neg\Pi$, then measuring Π retains all diagnoses predicting Π, and rejects all diagnoses predicting $\neg\Pi$. It is therefore tempting to conjecture that whenever every diagnosis predicts Π or $\neg\Pi$, then the diagnoses which remain after measuring Π are precisely those which predicted Π i.e. that the diagnoses for (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$) are precisely those for (SD, COMPONENTS, OBS) which predict Π. Unfortunately, as the next example shows, this conjecture is false.

Example 5.6. Consider the device of Fig. 8, with the indicated inputs and outputs. Recall that this had four diagnoses: $\{M_1\}$, $\{A_1\}$, $\{M_2, M_3\}$, $\{A_2, M_3\}$.

$\{M_1\}$ predicts $\text{out}(M_2) = 6$,

$\{A_1\}$ predicts $\text{out}(M_2) = 6$,

$\{M_2, M_3\}$ predicts $\text{out}(M_2) = 4$,

$\{M_2, A_2\}$ predicts $\text{out}(M_2) = 4$.

Suppose we now measure out(M_2) and obtain out(M_2) = 5. All four diagnoses predict out(M_2) $\neq$ 5, so that if the above conjecture were correct, this measurement should reject all four diagnoses, and *no new diagnoses should arise*. But in fact the four old diagnoses are replaced by four new ones: $\{M_1, M_2, M_3\}$, $\{M_1, M_2, A_2\}$, $\{M_2, M_3, A_1\}$, $\{M_2, A_1, A_2\}$.

Notice that each new diagnosis resulting from the measurement out(M_2) = 5 is a strict superset of some old diagnosis predicting out(M_2) $\neq$ 5. This is no accident, as the following result shows:

Theorem 5.7. *Suppose every diagnosis for* (SD, COMPONENTS, OBS) *predicts one of* Π, $\neg\Pi$. *Then*:

(1) *Every diagnosis for* (SD, COMPONENTS, OBS) *which predicts* Π *is a diagnosis for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$).

(2) *No diagnosis for* (SD, COMPONENTS, OBS) *which predicts* $\neg\Pi$ *is a diagnosis for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$).

(3) *Any diagnosis for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$) *which is not a diagnosis for* (SD, COMPONENTS, OBS) *is a strict superset of some diagnosis for* (SD, COMPONENTS, OBS) *which predicts* $\neg\Pi$. *In other words, any new diagnosis resulting from the new measurement* Π *will be a strict superset of some old diagnosis which predicted* $\neg\Pi$.

Proof. Claims (1) and (2) are simply Proposition 5.5. To prove claim (3) suppose that Δ_Π is a diagnosis satisfying the hypothesis of this claim. Because Δ_Π is a diagnosis for (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$),

$$\text{SD} \cup \text{OBS} \cup \{\Pi\} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta_\Pi\}$$

is consistent. Therefore

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta_\Pi\}$$

is consistent. Let Δ be a minimal subset of Δ_Π such that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent. By Proposition 3.4, Δ is a diagnosis for (SD, COMPONENTS, OBS). Since, by the hypothesis of claim (3), Δ_Π is not a diagnosis for (SD, COMPONENTS, OBS), Δ must be a strict subset of Δ_Π. It remains only to prove that Δ predicts $\neg\Pi$. By hypothesis of the theorem, Δ predicts one of Π, $\neg\Pi$, so assume to the contrary that Δ predicts Π, i.e. that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \models \Pi .$$

Therefore

$$\text{SD} \cup \text{OBS} \cup \{\Pi\} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent. But by Proposition 3.4, this together with the fact that Δ is a proper subset of Δ_Π implies that Δ_Π cannot be a diagnosis for (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$), contradiction. □

In general, Theorem 5.7 precludes a divide-and-conquer approach to discriminating among competing diagnoses on the basis of additional system measurements. While it is true that measuring Π preserves the old diagnoses predicting Π, and rejects the old diagnoses predicting $\neg\Pi$, new diagnoses can arise.

Corollary 5.8. *Suppose that* { } *is not a diagnosis for* (SD, COMPONENTS, OBS). *Then under the assumptions of Theorem* 5.7, *any new diagnosis arising from the new measurement* Π *will be a multiple fault diagnosis.*

Thus, in the nontrivial case where the system is truly faulty, the new diagnoses (if any) resulting from a new measurement will be multiple fault diagnoses. This provides the following characterization of the single fault diagnoses which survive a new measurement:

Corollary 5.9. *Suppose that* { } *is not a diagnosis for* (SD, COMPONENT, OBS). *Then under the assumptions of Theorem* 5.7, *the single fault diagnoses for* (SD, COMPONENTS, OBS $\cup$ $\{\Pi\}$) *are precisely those of* (SD, COMPONENTS, OBS) *which predict* Π.

Corollary 5.9 justifies a divide-and-conquer strategy for discriminating among competing single fault diagnoses on the basis of system measurements.

Example 5.10. Consider the device of Fig. 8, with the indicated inputs and outputs. Recall that this had two single fault diagnoses: $\{M_1\}$ and $\{A_1\}$. We can discriminate between these single fault diagnoses by measuring out(M_1) because $\{M_1\}$ predicts out(M_1) = 4 while $\{A_1\}$ predicts out(M_1) = 6. Suppose we measure out(M_1) and obtain out(M_1) = 6. Then by Corollary 5.9, we now know that $\{A_1\}$ is the only possible single fault diagnosis. Of course, there may still be other remaining multiple fault diagnoses, including new multiple fault diagnoses. In fact, no new diagnoses arise, and the remaining diagnoses after the measurement are: $\{A_1\}$, $\{M_2, M_3\}$, and $\{M_2, A_2\}$.

Many interesting problems remain to be explored for a theory of measure-

ment. Can we characterize situations in which measurements do not lead to new diagnoses but simply filter old ones? When new diagnoses do arise as a result of system measurements, can we determine these new diagnoses in a reasonable way from the pruned *HS*-tree already computed in determining the old diagnoses? Genesereth [8] describes a method for automatically generating certain system measurements. Are there other approaches to this test generation problem?

6. Generalizations and Relationship to Nonmonotonic Reasoning

Thus far our development of a theory of diagnosis has relied upon first-order logic as the underlying representation language for system descriptions. A close inspection of the preceding definitions, theorems, and proofs reveals that very few special features of first-order logic were actually required in developing the theory, so that the logical representation language may be generalized. In this section, we shall consider such generalizations. We shall also observe that diagnostic reasoning is nonmonotonic, and relate the theory of this paper to default logic [17].

6.1. Beyond first-order logic

In order to provide a concrete development of a theory of diagnoses, we have been assuming first-order logic with equality as the underlying representation language. In actual fact, Definition 2.4 of a diagnosis, the subsequent results of Section 3 leading to the algorithm DIAGNOSE of Section 4, and the results of Section 5 on measurements require very weak assumptions on the nature of the logic used. Specifically, suppose that L is any logic with the following properties:

(1) Its semantics is binary i.e. every sentence of L has value true or false in a given structure.

(2) L has $\{\wedge, \vee, \neg\}$ among its logical connectives, and these have their usual interpretations.

Then we can generalize the concept of a system so that a system description and its observation can be any set of sentences of L. The definition of a diagnosis remains the same in this generalized setting as in Definition 2.4. It is a simple matter to inspect the proofs of all results in Section 3 to see that they continue to hold, provided $\models$ is understood to denote the semantic entailment relation for the logic L. The "algorithm" DIAGNOSE of Section 4.3 for computing all diagnoses for a system remains the same and is correct for L. Of course, for DIAGNOSE really to be an algorithm, we need a sound, complete and decidable theorem prover for L at the core of the function TP which DIAGNOSE calls. It is also easy to see that our results of Section 4.4 on single fault diagnoses remain the same when L is the underlying logic. Finally, inspection of the proofs of Section 5 reveals that all of our results on measurements continue to hold for L.

Since our theory of diagnosis imposes such weak constraints on the system representation logic, the theory can accomodate a wide range of diagnostic tasks. For example, time varying digital hardware have natural representations in a temporal logic [12] and this might form the basis for a diagnostic reasoning system for such devices. Similarly, time varying physiological properties are central to certain kinds of medical diagnosis tasks [18]. Database logic has been proposed for representing many forms of databases [9] so that violation of database integrity constraints might profitably be viewed as a diagnostic reasoning problem with database logic providing the system description language.

These examples, and others like them, require a proper investigation with respect to problems of representation and computation. The fact that they all conform to a common theory of diagnosis is an encouraging and unifying observation.

6.2. Diagnosis and default logic[5]

As we have seen in Section 5, diagnostic reasoning is nonmonotonic in the sense that it can happen that none of a system's diagnoses survive a new observation of that system. In fact, as we now show, there is an intimate connection between diagnostic reasoning when the underlying logic is first-order and default logic [17].

To show the connection, we consider a system (SD, COMPONENTS) under observation OBS, and the corresponding default theory D whose first-order axioms are SD $\cup$ OBS, and whose default rules are

$$\left\{ \frac{:\neg \mathrm{AB}(c)}{\neg \mathrm{AB}(c)} \mid c \in \mathrm{COMPONENTS} \right\}.^{6}$$

The following theorem shows that there is a 1-1 correspondence between the diagnoses for (SD, COMPONENTS, OBS) and the extensions for the above default theory, and that these extensions are precisely the sentences predicted by the corresponding diagnoses.

Theorem 6.1. *Consider a system* (SD, COMPONENTS) *under observation* OBS *where* SD *and* OBS *are sets of first-order sentences. Then E is an extension for the*

[5] This section assumes that the reader is familiar with the literature on nonmonotonic reasoning (e.g. [1]), and specifically with default logic as described in [17].

[6] In [17] the notation $\alpha : M\beta / \gamma$ was used for default rules.The "M" was an unfortunate choice of notation meant to suggest "consistent" although it is in no way a sentential operator. As a piece of notation it was entirely spurious and we omit it from this paper.

default theory

$$\text{DT} = \left(\left\{\frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \mid c \in \text{COMPONENTS}\right\}, \text{SD} \cup \text{OBS}\right)$$

iff for some diagnosis Δ *for* (SD, COMPONENTS, OBS), $E = \{\Pi \mid \Delta \text{ predicts } \Pi\}$.

Proof. The proof relies upon the following proposition which follows easily from the results of [17]:

Proposition. *Suppose R is a set of default rules, each of the form* $:\alpha/\alpha$ *for* α *a first-order sentence. Then E is an extension for the default theory* (R, W) *iff* $E = \text{Th}(W \cup \{\beta \mid :\beta/\beta \in D\})$[7] *where D is a maximal subset of R such that* $W \cup \{\beta \mid :\beta/\beta \in D\}$ *is consistent.*

We now proceed with the proof of the main theorem.

($\Rightarrow$) Suppose E is an extension for DT. By the above proposition,

$$E = \text{Th}\left(\text{SD} \cup \text{OBS} \cup \left\{\neg \text{AB}(c) \mid \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \in D\right\}\right), \tag{6.1}$$

where D is a maximal subset of the default rules of DT such that

$$\text{SD} \cup \text{OBS} \cup \left\{\neg \text{AB}(c) \mid \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \in D\right\} \text{ is consistent}. \tag{6.2}$$

Let

$$\Delta = \left\{c \mid c \in \text{COMPONENTS and } \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \notin D\right\}.$$

Then

$$\left\{\neg \text{AB}(c) \mid \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \in D\right\} = \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}, \tag{6.3}$$

so that by (6.2), $\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$ is consistent. Moreover, Δ is a minimal subset of COMPONENTS with this property because D is a maximal subset of the default rules of DT with property (6.2). Hence, by Proposition 3.4, Δ is a diagnosis for (SD, COMPONENTS, OBS). Finally, by (6.1) and (6.3),

$$E = \text{Th}(\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}),$$

so that by Proposition 5.3, $E = \{\Pi \mid \Delta \text{ predicts } \Pi\}$.

[7] If S is a set of first-order sentences, Th(S) denotes the logical closure of S.

($\Leftarrow$) Suppose Δ is a diagnosis for (SD, COMPONENTS, OBS) and let $E = \{\Pi \mid \Delta \text{ predicts } \Pi\}$. By Proposition 5.3,

$$E = \text{Th}(\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}). \tag{6.4}$$

Since Δ is a diagnosis, then by Proposition 3.4 it is a minimal subset of COMPONENTS such that

$$\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\} \text{ is consistent}. \tag{6.5}$$

Let

$$D = \left\{\frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \mid c \notin \Delta\right\}.$$

Then (6.3) holds. Hence, by (6.4),

$$E = \text{Th}\left(\text{SD} \cup \text{OBS} \cup \left\{\neg \text{AB}(c) \mid \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \in D\right\}\right)$$

and by (6.5),

$$\text{SD} \cup \text{OBS} \cup \left\{\neg \text{AB}(c) \mid \frac{:\neg \text{AB}(c)}{\neg \text{AB}(c)} \in D\right\}$$

is consistent. Finally, D is a maximal subset of the default rules of DT with this consistency property since Δ is a minimal subset of COMPONENTS with property (6.5). Hence, by the above proposition, E is an extension for DT. □

7. Relationship to Other Research

The primary influences on this paper have been the work of de Kleer [5] and Genesereth [7].

De Kleer's research, while restricted to troubleshooting electronic circuits, appears to be among the earliest approaches in the literature to diagnosis from first principles. In his 1976 paper, de Kleer introduces the important concept of a conflict set, a concept which we have appropriated for our diagnostic algorithm. De Kleer's LOCAL system provided the diagnostic component for the SOPHIE III electronic computer aided instruction system [2]. In a recent paper de Kleer and Williams [6] have independently proposed a characterization of diagnoses, including multiple fault diagnoses, which corresponds to our Theorem 4.4. Their work differs from ours, however, in lacking a formalization of their diagnostic theory. On the other hand, de Kleer and Williams go beyond our theory of diagnosis by providing a method for computing the

probabilities of the different diagnoses that arise in a given setting, and for using these probabilities to identify appropriate system measurements to make next.

One of the first uses of logic as a representation language for diagnosis from first principles is due to Genesereth [7], who uses first-order logic for representing systems, and a resolution style theorem prover for computing candidate faults as well as for generating tests to discriminate among competing diagnoses. Our work extends and generalizes some of his results in a number of ways, the most fundamental of which are:

(1) Our theory applies equally well to a wide variety of logics, not just first-order.

(2) We provide a formal analysis of multiple fault diagnoses.

(3) We give an algorithm for computing all diagnoses, including multiple fault diagnoses.

(4) We prove various results about the effects of system measurements on diagnoses.

Davis [4] has proposed an approach to diagnosis from first principles, but oriented towards devices simulatable by constraint propogation techniques. Moreover, he addresses single fault diagnoses only. Unfortunately, Davis does not formalize his approach, so that comparisons between our theory and his are difficult to make. Nevertheless, Davis does describe a "Candidate Generation Procedure" for computing potential single fault diagnoses. As we remarked in Section 4.4, our Theorem 4.12 can be interpreted as a formal justification for Davis' procedure. On the other hand, his analysis of bridge faults in digital circuits is beyond the capabilities of our theory because our theory requires a fixed, a priori enumeration of the system components which might fail.

There are two other approaches to diagnosis from first principles, similar in spirit to ours in that they are logically based. One, by David Poole and his colleagues [10, 13] has independently observed the connection of diagnostic reasoning with default logic. Both references describe how a default logic theorem prover can be used to compute diagnoses, but the focus of these papers is on mechanisms for such computations, and hence is quite different than ours. The other logically based approach to diagnosis is by Ginsberg [8], who adopts a logic of counterfactual implication as a foundation for diagnostic reasoning. Following Genesereth [7] Ginsberg assumes a first-order representation of the system being diagnosed. His departure from Genesereth is to define diagnoses in terms of counterfactual consequences of the system observation. As an illustration of Ginsberg's theory, consider the full adder of Example 2.2. Recall that the diagnoses of this adder under the observation that it outputs 1, 0 in response to inputs 1, 0, 1 are $\{X_1\}$, $\{X_2, O_1\}$, $\{X_2, A_2\}$. If we denote by SD and OBS the adder's system description and observation, then the following formula is a counterfactual consequence of OBS with respect to the theory $\text{SD} \cup \{\neg\text{AB}(X_1), \neg\text{AB}(X_2), \neg\text{AB}(A_1), \neg\text{AB}(A_2), \neg\text{AB}(O_1)\}$:[8]

$$\text{AB}(X_1) \vee \text{AB}(X_2) \wedge \text{AB}(O_1) \vee \text{AB}(X_2) \wedge \text{AB}(A_2)$$

This, of course, represents the above three diagnoses for the adder. Obviously there is a close relationship between our consistency based definition of a diagnosis, and Ginsberg's counterfactual based definition. In fact, it is a simple matter to show that in the case of first-order logic there is a 1-1 correspondence between the diagnoses, in our sense, of (SD, COMPONENTS, OBS), and Ginsberg's possible worlds for OBS in $\text{SD} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS}\}$, provided all formulae in SD are protected. From this it follows that our two definitions of a diagnosis are essentially equivalent in the first-order case.

7.1. The GSC diagnostic model

A recent theory of diagnosis is the GSC (generalized set covering) model of Reggia, Nau and Wang [15, 16]. This provides a formal model of what they call "abductive diagnostic inference" and has been applied to problems of medical diagnosis [15]. In this section we describe the GSC model, show how it may be represented within our formalism, and using our formalism derive a characterization of its diagnoses which conforms (almost) to that defined by Reggia et al. A nice side effect of our logical reconstruction of the GSC model is the scope for generalizing the model which the logical representation provides.

In the GSC model, a *diagnostic problem* (D, M, C, M^+) is defined by four sets:

D—a finite set of *disorders* (e.g. in a medical setting D might represent all the known diseases).

M—a finite set of *manifestations* (e.g. in a medical setting M might represent all possible symptoms, laboratory results, etc. that can be caused by diseases in D).

$C \subseteq D \times M$. The relation C is meant to capture the notion of causation: $(d, m) \in C$ means "d can cause m."

$M^+ \subseteq M$. M^+ is the set of manifestations which have been observed to occur in the current diagnostic setting.

Within our formalism, we interpret a GSC model's diagnostic problem (D, M, C, M^+) as follows:

(1) Define a system (SD, D) whose components are the disorders of D, and whose system description SD is given by the following:

(i) For each disorder $d \in D$, SD contains the axiom DISORDER(d).

(ii) For each $m \in M$, if $(d_1, m), \ldots, (d_n, m)$ are all the elements of C with second component m, then SD contains the axiom

[8] Assuming that all the formulae of SD are protected. See [8] for details.

$$\text{OBSERVED}(m) \supset \text{PRESENT}(d_1) \vee \cdots \vee \text{PRESENT}(d_n)\,. \tag{7.1}$$

This says that an observed manifestation m must be "caused" by the presence of at least one of the disorders $d_1, \ldots, d_n$.

(iii) SD contains the axiom

$$(\forall d).\text{DISORDER}(d) \wedge \neg\text{AB}(d) \supset \neg\text{PRESENT}(d)\,,$$

i.e. normally, a disorder is not present.

(2) The observation of the above system is given by an axiom OBSERVED(m) for each $m \in M^+$.

This completes our logical reconstruction of the GSC diagnostic problem. We consider next the definition of a diagnosis (called an explanation by Reggia et al.) in the GSC model. If (D, M, C, M^+) is a diagnostic problem, then $E \subseteq D$ is a *cover* of M^+ iff for each $m \in M^+$ there exists $d \in E$ such that $(d, m) \in C$. E is a *minimum cardinality* cover[9] of M^+ iff $|E| \leq |E'|$ for every cover E' of M^+. E is a *minimal* cover[10] of M^+ iff no proper subset of E is a cover of M^+. According to Reggia et al., an *explanation* for a diagnostic problem (D, M, C, M^+) is defined to be a minimum cardinality cover for M^+. As we shall now see, it is the minimum cardinality property of an explanation, as distinct from the property of being minimal with respect to set inclusion, which will distinguish the concept of an explanation in the GSC model from the concept of a diagnosis in our logical reconstruction of the GSC model.

Theorem 7.1. *Suppose* (D, M, C, M^+) *is a diagnostic problem in the* GSC *model, and* (SD, D, OBS) *is the logical representation of this diagnostic problem as described above. Then* Δ *is a diagnosis for* (SD, D, OBS) *iff* Δ *is a minimal cover of* M^+.

Proof. Suppose $M^+ = \{m_1, \ldots, m_k\}$, so that all the axioms of SD of the form (7.1) are:

$$\begin{aligned}
&\text{OBSERVED}(m_1) \supset \text{PRESENT}(d_1^{(1)}) \vee \cdots \vee \text{PRESENT}(d_{n_1}^{(1)})\,,\\
&\qquad\vdots\\
&\text{OBSERVED}(m_k) \supset \text{PRESENT}(d_1^{(k)}) \vee \cdots \vee \text{PRESENT}(d_{n_k}^{(k)})\,.
\end{aligned}$$

It is easy to see that each of $\{d_1^{(1)}, \ldots, d_{n_1}^{(1)}\}, \ldots, \{d_1^{(k)}, \ldots, d_{n_k}^{(k)}\}$ is a conflict set for (SD, D, OBS) since clearly, for $i = 1, \ldots, k$, SD $\cup$ OBS $\cup$ $\{\neg\text{AB}(d_1^{(i)}), \ldots, \neg\text{AB}(d_{n_i}^{(i)})\}$ is inconsistent. We shall prove that any minimal conflict set for (SD, D, OBS) is one of the sets $\{d_1^{(i)}, \ldots, d_{n_i}^{(i)}\}$. To that end let K be a minimal conflict set for (SD, D, OBS), and suppose on the contrary that K is not one of the sets $\{d_1^{(i)}, \ldots, d_{n_i}^{(i)}\}$. Since K is a minimal conflict set and since each $\{d_1^{(i)}, \ldots, d_{n_i}^{(i)}\}$ is a conflict set, K cannot be a superset of $\{d_1^{(i)}, \ldots, d_{n_i}^{(i)}\}$. Hence, for $i = 1, \ldots, k$, each set $\{d_1^{(i)}, \ldots, d_{n_i}^{(i)}\}$ contains an element, say $d_1^{(i)}$, not contained in K. We shall show how to construct a model M of SD $\cup$ OBS $\cup$ $\{\neg\text{AB}(d) \mid d \in K\}$. From this it will follow that K cannot be a conflict set for (SD, D, OBS)—a contradiction.

To construct M, take $M \cup D$ as its domain. Take DISORDER(d) to be true for each $d \in D$, false otherwise. Let OBSERVED(m) be true for each $m \in M^+$, false otherwise. Finally, let AB(d) and PRESENT(d) both be false for each $d \in K$, true otherwise. Notice that PRESENT($d_1^{(i)}$) is true in M for $i = 1, \ldots, k$, since $d_1^{(i)} \notin K$. It follows that M is a model of SD $\cup$ OBS $\cup$ $\{\neg\text{AB}(d) \mid d \in K\}$.

To sum up, we have proved that every set of $\{\{d_1^{(1)}, \ldots, d_{n_1}^{(1)}\}, \ldots, \{d_1^{(k)}, \ldots, d_{n_k}^{(k)}\}\}$ is a conflict set for (SD, D, OBS) and that every minimal conflict set for (SD, D, OBS) is contained in $\{d_1^{(1)}, \ldots, d_{n_1}^{(1)}\}, \ldots, \{d_1^{(k)}, \ldots, d_{n_k}^{(k)}\}\}$.

Now it is simple to prove that $\Delta \subseteq D$ is a minimal cover of M^+ iff Δ is a minimal hitting set for $\{\{d_1^{(1)}, \ldots, d_{n_1}^{(1)}\}, \ldots, \{d_1^{(k)}, \ldots, d_{n_k}^{(k)}\}\}$. Since every member of this last set is a conflict set for (SD, D, OBS) and since every minimal conflict set for (SD, D, OBS) is a member of this last set, the theorem now follows by Corollary 4.5. □

Theorem 7.1 allows us to compare our concept of a diagnosis for the GSC model with the concept of an explanation of Reggia et al. For us, a diagnosis is a minimal cover of M^+, while for Reggia et al. it is a minimum cardinality cover of M^+. Now every minimum cardinality cover of M^+ is also a minimal cover of M^+, but not conversely. Thus every explanation as defined by Reggia et al. will be a diagnosis according to our theory, but not conversely. In fact, it is easy to see that the explanations for a GSC diagnostic problem are precisely the minimum cardinality diagnoses for our logical version of the diagnostic problem. This suggests that the concept of an explanation as a minimum cardinality cover of M^+ is inappropriate, that the appropriate concept should be based upon minimal covers as we have done.

In general, a theory of diagnosis could be based on minimal cardinality principles, as in the GSC model, but in our opinion such a theory would lead to unintuitive results. For example, in the case of the full adder, it would correctly yield the single fault diagnosis $\{X_1\}$, but overlook the two intuitively plausible double faults $\{X_2, A_2\}$ and $\{X_2, O_1\}$. Recently, Reggia and his colleagues

[9] Note that what we here call a minimum cardinality cover, Reggia et al. call a minimal cover.
[10] Not to be confused with what Reggia et al. call a minimal cover. See Footnote 9.

have independently adopted this point of view, and their current investigations of the GSC model and its extensions are based upon minimal covers of M^+, rather than minimal cardinality covers [14].

8. Summary

We summarize what we take to be the main contributions of this paper to a theory of diagnosis from first principles:

(1) The definition of the concept of a diagnosis, including multiple fault diagnoses, based upon the preservation of the consistency of the system description and its observation.

(2) The explicit use of the AB predicate for representing faults and possible relationships between faults.

(3) The ability of the theory to accommodate a wide variety of logics.

(4) The algorithm DIAGNOSE for computing all diagnoses.

(5) Characterizations of single fault diagnoses and their computation.

(6) Various results about the affects of system measurements on diagnoses.

(7) The nonmonotonic character of diagnosis, specifically its relationship to default logic.

ACKNOWLEDGMENT

I owe a special debt to Johan de Kleer for his encouragement, and for his ready responses to my many questions about diagnosis, especially during the initial phases of this research. Many thanks to David Etherington for acting as an insightful sounding board during the development of these ideas. My thanks also to Teresa Miao for her careful and patient preparation of this manuscript.

This research was done with the financial support of the National Sciences and Engineering Research Council of Canada, under operating grant A7642.

REFERENCES

1. Bobrow, D.G. (Ed.), Special Issue on Non-Monotonic Logic, *Artificial Intelligence* **13** (1, 2) (1980).
2. Brown, J.S., Burton, D. and de Kleer, J., Pedagogical natural language and knowledge engineering techniques in SOPHIE I, II and III, in: D. Sleeman and J.S. Brown (Eds.), *Intelligent Tutoring Systems* (Academic Press, New York, 1982) 227–282.
3. Buchanan, B.G. and Shortliffe E.H. (Eds.), *Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (Addison-Wesley, Reading, MA, 1984).
4. Davis, R., Diagnostic reasoning based on structure and behavior, *Artificial Intelligence* **24** (1984) 347–410.
5. de Kleer, J., Local methods for localizing faults in electronic circuits, MIT AI Memo 394, Cambridge, MA, 1976.
6. de Kleer, J. and Williams, B.C., Diagnosing multiple faults, *Artificial Intelligence* **32** (1987) 97–130 (this issue).
7. Genesereth, M.R., The use of design descriptions in automated diagnosis, *Artificial Intelligence* **24** (1984) 411–436.
8. Ginsberg, M.L., Counterfactuals, *Artificial Intelligence* **30** (1986) 35–79.
9. Jacobs, B.E., On database logic, *J. ACM* **29** (2) (1982) 310–332.
10. Jones, M. and Poole, D., An expert system for educational diagnosis based on default logic, in: *Proceedings Fifth International Workshop on Expert Systems and their Applications* **II**, Avignon, France (1985) 673–683.
11. McCarthy, J., Applications of circumscription to formalizing common-sense knowledge, *Artificial Intelligence* **28** (1986) 89–116.
12. Moszkowski, B., A temporal logic for multilevel reasoning about hardware, *IEEE Computer* **18** (2) (1985) 10–19.
13. Poole, D., Aleliunas, R. and Goebel, R., Theorist: A logical reasoning system for defaults and diagnosis, Tech. Rept., Logic Programming and Artificial Intelligence Group, Department of Computer Science, University of Waterloo, Waterloo, Ont., 1985.
14. Reggia, J.A., Personal communication.
15. Reggia, J.A., Nau, D.S. and Wang, Y., Diagnostic expert systems based on a set covering model, *Int. J. Man-Mach. Stud.* **19** (1983) 437–460.
16. Reggia, J.A., Nau, D.S. and Wang, Y., A formal model of diagnostic inference I. Problem formulation and decomposition, *Inf. Sci.* **37** (1985) 227–256.
17. Reiter, R., A logic for default reasoning, *Artificial Intelligence* **13** (1, 2) (1980) 81–132.
18. Tsotsos, J.K., Knowledge organization and its role in representation and interpretation for time-varying data: The ALVEN system, *Comput. Intell.* **1** (1985) 16–32.

Received March 1986; revised version received June 1986