Model-Based Diagnosis using Causal Networks

Adnan Darwiche
Rockwell Science Center
1049 Camino Dos Rios
Thousand Oaks, CA 91360
darwiche@rpal rockwell com

Abstract

This paper rests on several contributions. First, we introduce the notion of a *consequence,* which is a boolean expression that characterizes consistency-based diagnoses. Second, we introduce a basic algorithm for computing consequences when the system description is structured using a causal network. We show that if the causal network has no undirected cycles, then a consequence has a linear size and can be computed in linear time. Finally, we show that diagnoses characterized by a consequence and meeting some preference criterion can be extracted from the consequence in time linear in its size. A dual set of results is provided for abductive diagnosis.

1 Introduction

This paper presents an approach for computing diagnoses [Reiter, 1987; de Kleer *et a*/., 1992] when the system description is structured using a causal network — Figures 1 and 2 depict examples of structured system descriptions.

The most common approach for computing diagnoses has been the use of Assumption-Based Truth Maintenance Systems (ATMSs) [de Kleer, 1986; Reiter and de Kleer, 1987]. We will first explain the difficulties with such an approach and then describe the elements of our approach that address these difficulties.

An ATMS assigns a "label" to each proposition. The label of proposition *o* characterizes all consistency-based diagnoses of the observation -o. Once the label of a proposition is computed, one can immediately check whether the proposition is logically true. Therefore, computing labels is no easier than deciding satisfiability, which is one source of difficulty with this approach. What makes the ATMS approach especially difficult, however, is that labels can grow exponentially in size, even on very simple diagnostic problems. This difficulty has led to a body of research on "focusing" the ATMS, which attempts to control the size of ATMS labels. Focusing is based on the following intuition. The label of proposition *o* characterizes all diagnoses of observation -o. But one is rarely interested in all diagnoses, therefore, one rarely needs a "complete" label. Most often, one is interested in diagnoses that satisfy some preference criterion (for example, most probable diagnoses). Therefore, one can use such a criterion to compute "focused" labels that are of reasonable size, yet are good enough to characterize the diagnoses of interest.

Although a standard framework exists for computing ATMS labels [Forbus and de Kleer, 1993], no such framework seems to exist for focusing.

The approach we present in this paper is based on three main ideas:

Characterizing diagnoses using consequences: We introduce the notion of a *consequence* for characterizing all consistency-based diagnoses. The size of a consequence (which is a boolean expression) is always less than the size of a label. In fact, there are diagnostic problems that have exponential-size labels and linear-size consequences.

Utilizing system structure in computing consequences: We introduce a basic algorithm for computing consequences, the complexity of which is parameterized by the topology of the system's causal structure. We show that for singly-connected structures (no undirected cycles), the consequence is always linear in size and can be computed in linear time. For some of these structures, a label can be exponential in size.

A principled mechanism for focusing on preferred diagnoses: We show that if a consequence has a particular syntax (and-or tree where no symbols are shared between and-branches), then one can extract the diagnoses it characterizes and that meet a specific preference criterion in time linear in the size of the consequence. Diagnoses with the highest order-of magnitude probability is an example of such a preference criterion.

Therefore, we are providing a paradigm for diagnostic reasoning with causal structures, consequences, and preference criteria as the key components. By using this paradigm, one is guaranteed some complexity results that are parameterized by the topology of the system's causal structure. As we shall see, this approach is based on the causal-network paradigm in the probabilistic and constraint satisfaction literatures [Dechter and Dechter, 1994; Geffner and Pearl, 1987; Dechter and Dechter, 1988]. In both cases, the system structure is the key aspect that decides the difficulty of a reasoning problem. This (conceptually meaningful) parameter is what diagnostic practitioners need to control

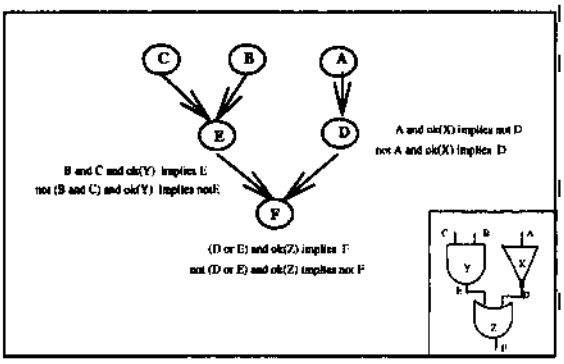

Figure 1: A structured system description (causal network) of a digital circuit.

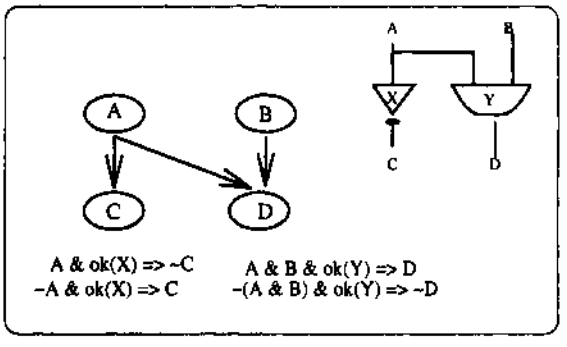

Figure 2: A structured system description (causal network) of a digital circuit.

in order to ensure an appropriate response time for their applications. The probabilistic literature contains many techniques for tweaking this parameter to ensure certain response times, most of which can be adopted by our proposed framework.

2 Characterizing Diagnoses

In the diagnostic literature [de Kleer *et al.*, 1992], a *system* is typically characterized by a tuple $(\Delta, \mathbf{P}, \mathbf{A}, \phi)$ where Δ is a database constructed from atomic propositions $\mathbf{P} \cup \mathbf{A}$ and ϕ is a sentence constructed from P. The atoms in A are called assumables and those in P are called non-assumables. The intention is that the database describes the system behavior, the assumables represent the modes of system components and the sentence ϕ represents the observed system behavior.

A *diagnosis* is defined as a conjunction of literals that is consistent with $\Delta \cup \{\phi\}$ and that includes one literal for each assumable. Therefore, a diagnosis is an assignment of modes to components that is consistent with the system description and its observed behavior. In Figure 1, okX, okY and okZ are the assumables and $okX \wedge \neg okY \wedge okZ$ is a potential diagnosis.

An ultimate goal of diagnostic reasoning is to compute the most preferred diagnoses (according to some criterion) for a given system $(\Delta, \mathbf{P}, \mathbf{A}, \phi)$. The approach we propose in this paper achieves this objective in two steps. First, we compute "the consequence" of observation ϕ, which is a boolean expression that characterizes all the diagnoses of ϕ. Second, we extract the most preferred diagnoses from the computed consequence.

The consequence of an observation is defined formally below:[1]

Definition 1 *The consequence of observation ϕ, written Cons(), is the logically strongest sentence α constructed from atoms* A *such that* $\Delta \cup \{\phi\} \models \alpha$.[2]

In Figure 2, for example, the consequence of observation $C \wedge D$ is $\neg okX \vee \neg okY$ because it is the logically strongest sentence (constructed from assumables) that can be concluded from the given observation and system description.

A consequence characterizes all diagnoses in the following way:

Theorem 1 *d is a diagnosis for system $(\Delta, \mathbf{P}, \mathbf{A}, \phi)$ iff* $d \models Cons(\phi)$.

The consequence $\neg okX \vee \neg okY$ characterizes three diagnoses: $\neg okX \wedge \neg okY$, $okX \wedge \neg okY$ and $\neg okX \wedge okY$. Using "the most probable diagnosis" as the preference criterion, the most preferred diagnoses would be $okX \wedge \neg okY$ and $\neg okX \wedge okY$. We are assuming here that components are unlikely to break, they break independently and their probabilities of failures are equal.

3 The Role of System Structure

We will refer to the triple $(\Delta, \mathbf{P}, \mathbf{A})$ as a *system description* and keep it implicit whenever possible. We will also assume that any satisfiable sentence constructed from assumables is consistent with database Δ. This means that the system description does not fix the mode of any component.

Given some observation ϕ and some preference criterion, our ultimate goal is to compute all preferred diagnoses of ϕ according to this criterion. We will do this in two steps. First, we will compute the consequence of ϕ, which characterizes all its diagnoses. Second, we will extract from $Cons(\phi)$ the preferred diagnoses. The second step will be addressed in Section 5. In this and the following section, we focus on the first step.

We start with the following properties of consequences:

(C1) $Cons(true) \equiv true$,

(C2) $Cons(false) \equiv false$,

(C3) $Cons(\phi \vee \psi) \equiv Cons(\phi) \vee Cons(\psi)$, and

(C4) $Cons(\phi) \equiv Cons(\psi)$ when $\phi \equiv \psi$.

Note, however, that we do not have

[1] We use a capital letter (such as Y) to denote an atomic proposition, a small letter (such as y) to denote a literal, and a boldface letter such as Y or y to denote a set of atomic propositions or a set of literals, respectively.

[2] The consequence of a sentence is unique up to logical equivalence.

(C5) $Cons(\phi \wedge \psi) \equiv Cons(\phi) \wedge Cons(\psi)$.

Consider the system in Figure 2 for an example. The consequence of *C* is *true,* the consequence of *D* is *true,* but the consequence of $C \wedge D$ is $\neg okX \vee \neg okY$.

If Property C5 were true, then computing consequences would be very easy. To compute the consequence of ζ, we keep rewriting the expression $Cons(\zeta)$ using C1-C5 until we reach a boolean expression that involves only the connectives $\wedge$ and $\vee$ and consequences *Cons(n)),* where n is an observation that is local to an individual component. Such consequences, called *local consequences,* can be computed easily since they can be inferred from the component description.

Property C5 does not hold, however, and this makes the computation of consequences more subtle. Property C5 may hold in certain cases. When it does, we say that ϕ is "independent" of ψ. For example, *C* and *D* are not independent in Figure 2 because $Cons(C \wedge D) \not\equiv Cons(C) \wedge Cons(D)$. More generally:

Definition 2 *Let* I, *3, and* K *be disjoint subsets of P. The sets* I *and 3 are conditionally* <u>independent</u> *given* K *precisely when*

(C6) $Cons(\alpha \wedge \beta \wedge \gamma) \equiv Cons(\alpha \wedge \gamma) \wedge Cons(\beta \wedge \gamma)$

for all conjunctive clau:α, β and γ over I, J, and K, *respectively.3*

When K is empty, we say that I and J are marginally independent. Note that Property C5 is a special case of Property C6 when K is empty since *true* is the only conjunctive clause over the empty set of atoms.

In Figure 2, for example, let $\mathbf{I} = \{C\}$, $\mathbf{J} = \{D\}$ and $\mathbf{K} = \{A\}$. Then $\mathbf{I}$ and $\mathbf{J}$ are not independent, since $Cons(C \wedge D) \not\equiv Cons(C) \wedge Cons(D)$ as we verified earlier. However, $\mathbf{I}$ and $\mathbf{J}$ are independent given $\mathbf{K}$ since

$$\begin{aligned} Cons(C \wedge D \wedge A) &\equiv Cons(C \wedge A) \wedge Cons(D \wedge A) \\ &\equiv \neg okX \wedge true; \\ Cons(C \wedge D \wedge \neg A) &\equiv Cons(C \wedge \neg A) \wedge Cons(D \wedge \neg A) \\ &\equiv true \wedge \neg okY; \end{aligned}$$

and similarly for $Cons(C \wedge \neg D \wedge A)$, $Cons(\neg C \wedge D \wedge A)$, etc.

Therefore, the key to computing consequences is the ability to detect system independences, which would be used to invoke Property C6. As we shall see next, the causal structure of a system is a very rich source of system independences. Explicating such a structure when describing systems, and detecting system independences from such a structure, is the topic of the next section.

3.1 Structured System Descriptions

When a system is described as in Figures 1 and 2, the result is called a *structured system description.*

A structured system description has two components: A *causal structure* and a set of *component descriptions.* The causal structure depicts the interconnections between system components, and component descriptions describe the functionality of system components.4 Formally, a causal structure is a directed acyclic graph, the nodes of which are the non-assumables P. A component description is a set of material implications. There is one component description (possibly empty) for each node in the causal structure. The component description associated with node N contains only two types of material implications: $\psi \wedge \alpha \supset N$ and $\phi \wedge \beta \supset \neg N$, where (1) ψ and ϕ are constructed from the parents of *N* in the causal structure; (2) α and β are constructed from assumable atoms A; and (3) $\psi \wedge \alpha \wedge \phi \wedge \beta$ must be inconsistent. These conditions hold iff a component description is local to a single component (1 and 2) and does not constrain the inputs of that component (3).

We will $(\mathcal{G}, \Delta, \mathbf{P}, \mathbf{A})$ to denote a structured system description, where G is the causal structure, Δ is the union of component descriptions, P are the atoms in G, and A are the atoms appearing in Δ but not in G.

3.2 System Independences from System Structure

A most important property of a structured system description is that its topology explicates many system independences:

Theorem 2 ([Darwiche, 1993]) *Let* $(\mathcal{G}, \Delta, \mathbf{P}, \mathbf{A})$ *be a structured system description and let* I, J, *and K be disjoint sets of atoms in G. If I and* J *are d-separated by* K *in G, then* I *and 3 are conditionally independent given* K *wrt to* (Δ, P, A).

d-separation is a topological test that can be performed in polynomial time and is discussed in detail elsewhere [Pearl, 1988].

In Figure 2, $\{C\}$ a r $\{D\}$ r e not d-separated by the empty set, which means that $\{C\}$ and $\{D\}$ may not be marginally independent (this was confirmed in the previous section). But $\{C\}$ and $\{D\}$ are d-separated by $\{A\}$, which means that they are conditionally independent given $\{A\}$

This independence is useful for computing the consequence of observation $\underline{C} \wedge \underline{D}$. We first use $C \wedge D \equiv (C \wedge D \wedge A) \vee (C \wedge D \wedge \neg A)$ to apply Property C3:

$$Cons(C \wedge D) \equiv Cons(C \wedge D \wedge A) \vee Cons(C \wedge D \wedge \neg A).$$

The independence of $\{C\}$ and $\{D\}$ given $\{A\}$ validates Property C6:

$$\begin{aligned} Cons(C \wedge D \wedge A) &\equiv Cons(C \wedge A) \wedge Cons(D \wedge A) \\ &\equiv \neg okX \wedge true, \\ Cons(C \wedge D \wedge \neg A) &\equiv Cons(C \wedge \neg A) \wedge Cons(D \wedge \neg A) \\ &\equiv true \wedge \neg okY. \end{aligned}$$

Therefore, $Cons(C \wedge D) \equiv \neg okX \vee \neg okY$.

The technique of applying Property C3 to generate consequences that can be decomposed using Property C6 is very powerful. In fact, the algorithm to be given in the following section for computing consequences is based on making (optimal) use of this technique.

3A *conjunctive clause* over atoms X is a conjunction of literals that includes one literal for each atom in X.

4 A structured system description is a special case of a *symbolic causal network* [Darwiche and Pearl, 1994].

4 Computing Consequences

Before we discuss the algorithm, we will consider a more elaborate example to provide more intuition on the computational value of system independences.

Consider Figure 3, an example from [Freitag and Friedrich, 1992], which depicts part of an audio switching matrix typically used in broadcasting stations for the flexible connection of studios, recording devices, etc. The given system consists of one input amplifier, 1000 output amplifiers and 1000 switches. For the sake of simplicity, an audio matrix is represented by and-gates and buffers which logically produce the same behavior. The following is observed about the system: the input signal is ON, the first and-gate gets an OFF signal and all other and-gates get ON signals. The output of buffer C_5 is OFF, while outputs of all other buffers are ON. We would like to compute the consequence of this system behavior, therefore, characterizing all diagnoses.

As it turns out, diagnosing this system is easy because its causal structure (shown in Figure 3) explicates independences that can be used to decompose the global consequence into local consequences that can be evaluated locally. The systems independences are:

SI1: $\{I_1\}$ is independent of all other atoms given $\{C_1\}$.

SI2: $\{I_i, C_i, C_{i+1}\}$ are independent of $\{I_j, C_j, C_{j+1}\}$ given $\{C_1\}$, where $i, j = 2, 4, \ldots, 2000$ and $i \neq j$. That is, the atoms corresponding to each connected amplifier and switch are independent of those corresponding to other amplifiers and switches given $\{C_1\}$.

We will now show how these independences can be used to decompose the consequence of interest, $Cons(I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \ldots I_{2000} \wedge C_{2001})$, which characterizes all diagnoses. The assumables here are $abC_1, \ldots, abC_{2001}$ and the non-assumables are $I_1, I_2, I_4, \ldots, I_{2000}$ and $C_1, C_3, C_5, \ldots, C_{2001}$.

We first perform a case analysis on C_1 by applying Property C3:

$$Cons(I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001}) \equiv$$
$$Cons(C_1 \wedge I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001}) \vee$$
$$Cons(\neg C_1 \wedge I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001})$$

Given SI1, further decomposition is possible using Property C6:

$$Cons(I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001})$$
$$\equiv [Cons(C_1 \wedge I_1) \wedge Cons(C_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001})] \vee$$
$$[Cons(\neg C_1 \wedge I_1) \wedge Cons(\neg C_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001})].$$

Given SI2, we can decompose the above consequences using Property C6:

$$Cons(C_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \ldots I_{2000} \wedge C_{2001})$$
$$\equiv Cons(C_1 \wedge \neg I_2 \wedge C_3) \wedge Cons(C_1 \wedge I_4 \wedge \neg C_5) \ldots Cons(C_1 \wedge I_{2000} \wedge C_{2001})$$
$$Cons(\neg C_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \ldots I_{2000} \wedge C_{2001})$$
$$\equiv Cons(\neg C_1 \wedge \neg I_2 \wedge C_3) \wedge Cons(\neg C_1 \wedge I_4 \wedge \neg C_5) \ldots Cons(\neg C_1 \wedge I_{2000} \wedge C_{2001}).$$

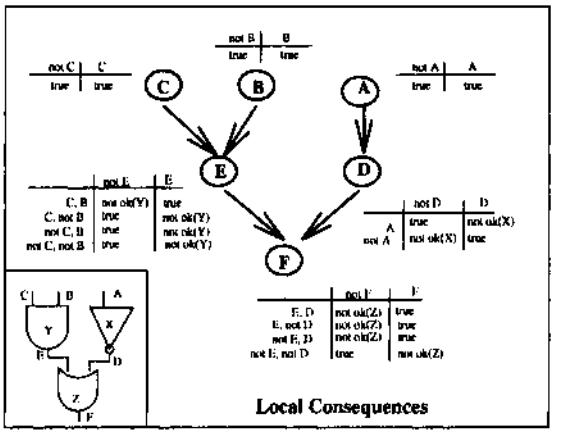

Figure 4: A structured system description expressed in terms of local consequences. The entry in row ψ and column ϕ represents the consequence of $\psi \wedge \phi$.

Each of the resulting consequences can be evaluated locally. For example, the consequence of $\neg C_1 \wedge I_1$ is abC_1 and the consequence of $C_1 \wedge I_1$ is *true* (normal behavior). Moreover, each of these consequences depends only on the behavior of buffer C_1. Similarly, the consequences of $C_1 \wedge \neg I_2 \wedge C_3$, $C_1 \wedge I_4 \wedge \neg C_5$ and $C_1 \wedge I_6 \wedge C_7$ are $abC_2 \vee abC_3$, $abC_4 \vee abC_5$ and *true*, respectively, and these consequences depend only on the behavior of components $\{C_2, C_3\}$, $\{C_4, C_5\}$ and $\{C_6, C_7\}$, respectively. Substituting for the above consequences, we get

$$Cons(I_1 \wedge \neg I_2 \wedge C_3 \wedge I_4 \wedge \neg C_5 \wedge \ldots I_{2000} \wedge C_{2001})$$
$$\equiv [true \wedge (abC_2 \vee abC_3) \wedge (abC_4 \vee abC_5) \wedge true \wedge \ldots \wedge true] \vee$$
$$[abC_1 \wedge (abC_2 \vee abC_3) \wedge true \wedge (abC_6 \vee abC_7) \wedge \ldots \wedge (abC_{2000} \vee abC_{2001})],$$

which simplifies to:

$$(abC_2 \vee abC_3) \wedge [abC_4 \vee abC_5 \vee (abC_1 \wedge (abC_6 \vee abC_7) \wedge \ldots \wedge (abC_{2000} \vee abC_{2001}))].$$

This is the final answer, which is a boolean expression in the form of an and-or tree. This tree will be the input to the focusing phase discussed in Section 5, where the most preferred diagnoses are computed. In the remainder of this section, we present the general algorithm for computing consequences.

The algorithm computes the consequence of observation o, which is a conjunctive clause over a subset **O** of the non-assumables **P**. The algorithm is symmetric to a well-known one in the probabilistic literature [Pearl, 1988]. It applies to singly-connected networks in which only one undirected path exists between any two nodes. The algorithm can be extended to multiply-connected networks in a straightforward manner, leading to an algorithm that is exponential in the size of a network cutset at worse [Pearl, 1988].[5]

[5] Multiply-connected networks are not necessarily harder

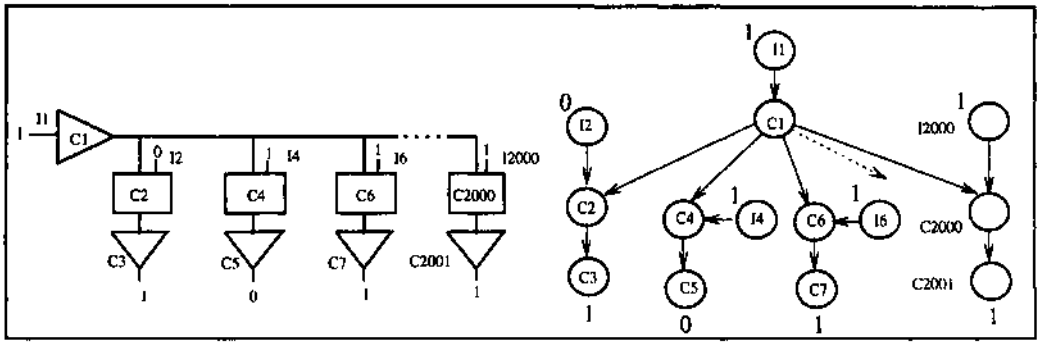

Figure 3: A digital system and its corresponding causal structure. The square boxes are and-gates and the triangles are buffers. System observations are shown on corresponding wires. I_i and C_{i+1} are ON for $i = 8, 10, \ldots, 1998$.

The algorithm works by rewriting the expression $Cons(\mathbf{o})$ using Properties C1-C6 and stops when every term in the expression is of the form $Cons(x \wedge \mathbf{u})$, where x is a literal of atom X and $\mathbf{u}$ is a conjunctive clause over its parents $\mathbf{U}$. The term $Cons(x \wedge \mathbf{u})$ is called a *local consequence* and it involves a complete observation of all inputs and output of a single component. Figure 4 shows local consequences for the system in Figure 1.

The algorithm can be described as a recursive and deterministic[6] application of the following rewrite rules:

$$
\begin{aligned}
Cons(\mathbf{o}) &\rightarrow Cons(x \wedge \mathbf{o}) \vee Cons(\neg x \wedge \mathbf{o}) \\
Cons(x \wedge \mathbf{o}) &\rightarrow \pi(x) \wedge \lambda(x) \\
\pi(x) &\rightarrow \bigvee_{\mathbf{u}} Cons(x \wedge \mathbf{u}) \bigwedge_{\mathbf{u} \models u} \pi_X(u) \\
\lambda(x) &\rightarrow Obs(x) \bigwedge_{Y} \lambda_Y(x) \\
\pi_Y(x) &\rightarrow \pi(x) \wedge Obs(x) \bigwedge_{Y'} \lambda_{Y'}(x) \\
\lambda_X(u) &\rightarrow \bigvee_{x} \lambda(x) \wedge \bigvee_{\mathbf{u} \models u} Cons(x \wedge \mathbf{u}) \bigwedge_{\mathbf{u} \models u'} \pi_X(u') \\
Obs(x) &\rightarrow \begin{cases} false & \text{if } \mathbf{o} \models \neg x; \\ true & \text{otherwise.} \end{cases}
\end{aligned}
$$

Here, X is an arbitrary node in the network; U and U' are distinct parents of X; Y and Y' are distinct children of X; and $\mathbf{U}$ contains all parents of X. Lowercase letters represent conjunctive clauses over their corresponding atoms.

The algorithm starts by rewriting $Cons(\mathbf{o})$ using Property C3 into $Cons(x \wedge \mathbf{o}) \vee Cons(\neg x \wedge \mathbf{o})$ for some atom X in the network. To compute the consequence of $x \wedge \mathbf{o}$, the algorithm partitions the observation $\mathbf{o}$ into two parts, $\mathbf{o}_X^-$ about the descendants of X and $\mathbf{o}_X^+$ about the non-descendants of X; see Figure 5. This validates Property C6:

$$
\begin{aligned}
Cons(x \wedge \mathbf{o}) &\equiv Cons(x \wedge \mathbf{o}_X^+ \wedge \mathbf{o}_X^-) \\
&\equiv \underbrace{Cons(x \wedge \mathbf{o}_X^+)}_{\pi(x)} \wedge \underbrace{Cons(x \wedge \mathbf{o}_X^-)}_{\lambda(x)}
\end{aligned}
$$

because X d-separates any of its descendants from any of its non-descendants.

To compute the consequence $Cons(x \wedge \mathbf{o}_X^-)$, the algorithm partitions the observation $\mathbf{o}_X^-$ into a number of observations $\mathbf{o}_{XY}^-$, each about the nodes connected to one child Y of X; see Figure 5. This validates Property C6:

$$
\begin{aligned}
Cons(x \wedge \mathbf{o}_X^-) &\equiv Cons(x \wedge \bigwedge_{Y} \mathbf{o}_{XY}^-) \\
&\equiv \bigwedge_{Y} \underbrace{Cons(x \wedge \mathbf{o}_{XY}^-)}_{\lambda_Y(x)}
\end{aligned}
$$

because X d-separates any nodes connected to one of its children from any nodes connected to other children.

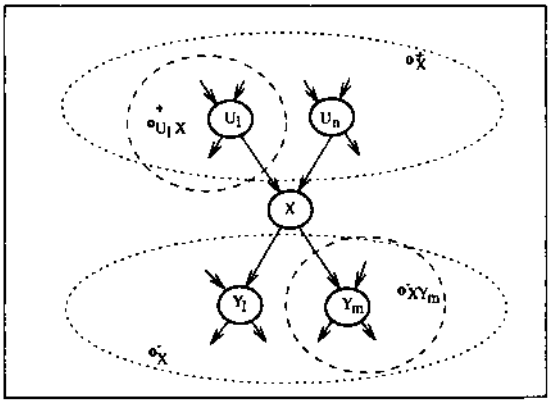

Figure 5: Decomposing observation nodes in a singly-connected networks.

than singly-connected ones. For example, n-bit adders lead to multiply-connected networks that behave computationally like singly-connected ones.

[6]Except for the first rewrite rule, which applies only once.

To compute $Cons(x \wedge o_X^+)$, the algorithm applies Property C3:

$$Cons(x \wedge o_X^+) \equiv \bigvee_{u} Cons(x \wedge u \wedge o_X^+),$$

where u is a conjunctive clause over U (the parents of

To compute $Cons(x \wedge u \wedge o_X^+)$, the algorithm partitions the observation o_X^+ into a number of observations o_{UX}^+, each about the nodes connected to X through its parent U; see Figure 5. This allows the application of Property C6:

$$Cons(x \wedge u \wedge o_X^+) \equiv Cons(x \wedge u) \bigwedge_{u \models u} \underbrace{Cons(u \wedge o_{UX}^+)}_{\pi_X(u)},$$

which can also be justified using d-separation and Theorem 2. A detailed derivation of this algorithm can be found in [Darwiche, 1992].

The complexity of the algorithm is similar to its probabilistic counterpart: linear in the number of arcs but exponential in the number of parents per node. We can verify this by counting the number of conjoin and disjoin operations.

5 Extracting Preferred Diagnoses

The algorithm presented in the previous section computes consequences that have the form of an and-or tree. If component descriptions do not share assumables, then no assumables will be shared by the branches of any and-node in the tree. In this section, we show that if a consequence satisfies the previous two properties, then one can extract from it the most preferred diagnosis in time linear in the size of the consequence, as long as the preference criterion meets some conditions.

The preference criterion is specified by a triple $(\Sigma, \oplus, \leq_\oplus)$ where Σ is a set of costs, $\oplus$ is some cost addition operation and $\leq_\oplus$ is a cost total ordering.7 The cost function should be such that each literal or its negation has a zero cost and the cost of a diagnosis is obtained by adding the costs of its individual literals. An example of such a preference criterion is $(\{0, 1, 2, \ldots\}, +, \leq)$, where the cost of a literal is the order-of-magnitude of its probability.8

Given a preference criterion $(\Sigma, \oplus, \leq_\oplus)$ and given an and-or tree r (with no assumables shared by branches of and-nodes), one can extract its most preferred diagnoses $pd(\tau)$ using the following recursive procedure:9

1. If $\tau = l$, where l is a leaf node, then $pd(\tau) = \{l\}$
2. If $\tau = \tau_1 \vee \tau_2$, then $pd(\tau)$ is the least costly members of $pd(\tau_1) \cup pd(\tau_2)$[10]
3. If $\tau = \tau_1 \wedge \tau_2$, then $pd(\tau)$ is $\{\alpha \wedge \beta\}$ where $\alpha \in pd(\tau_1)$ and $\beta \in pd(\tau_2)$.[11]

It is clear that the above procedure involves only a linear number of recursive calls, one for each node in the tree. What remains to be shown is some guarantee on the size of *pd(Ti)* during these recursive calls. As it turns out, each subtree on which a recursive call may apply represents the answer to a diagnostic problem that involves part of the observation o and some local observations involving a single component. In particular, each $pd(\tau_i)$ corresponds to one of $pd(x \wedge o)$, $pd(x \wedge o_X^+)$, $pd(x \wedge o_X^-)$, $pd(x \wedge o_{XY}^-)$, and $pd(x \wedge u \wedge o_{UX}^+)$, where X is an arbitrary node in the network, Y is one of its children, U is one of its parents, and u is a conjunctive clause over these parents.

We can summarize the guarantees for computing most-preferred diagnoses as follows. First, computing the consequence is linear in the number of nodes and exponential in the number of parents per node in a causal structure: The computed consequence has the same size. Second, extracting the most preferred diagnoses from the consequence involves a number of minimization and conjunction operations that is linear in the size of the consequence. Finally, each one of these operations is applied to a pair of sets, each containing the preferred diagnoses of an asymptotically simpler diagnostic problem.

6 Dual Results for Abduction

There is a dual to consequence calculus, called *argument calculus,* which associates *arguments* with sentences instead of consequences. The role that arguments play in abductive reasoning is similar to the role that consequences play in diagnostic reasoning. Following is the definition of an argument wrt a system description $(\Delta, \mathbf{P}, \mathbf{A})$ and observation ϕ:

Definition 3 *The argument for* ϕ, *written* $Arg(\phi)$, *is the logically weakest sentence a constructed from atoms* A *such that* $\Delta \cup \{\alpha\} \models \phi$.

The duality between arguments and consequences is given below:

Theorem 3 $Arg(\phi) \equiv \neg Cons(\neg\phi)$.

Intuitively, the most general argument supporting ϕ is that the most specific outcome of $\neg\phi$ does not hold.

Argument calculus can be viewed as a semantical ATMS since the prime implicants of $Arg(\phi)$ constitute the ATMS label of ϕ [Darwiche, 1993]. This result, together with Theorem 3, explains the influential role that ATMSs have been playing in diagnostic reasoning.

The following properties hold for arguments [Darwiche, 1993]: $Arg(true) \equiv true$, $Arg(false) \equiv false$, $Arg(\phi \wedge \psi) \equiv Arg(\phi) \wedge Arg(\psi)$, and $Arg(\phi) \equiv Arg(\psi)$ when $\phi \equiv \psi$. Again, we do not have the property: $Arg(\phi \vee \psi) \equiv Arg(\phi) \vee Arg(\psi)$. Consider the system in Figure 1 for an example. The argument for *A* is *false,*

$cost(\alpha) <_\oplus cost(\beta)$ only if $cost(\alpha \wedge \gamma) <_\oplus cost(\beta \wedge \delta)$ for some γ and all δ where $\alpha \wedge \gamma$ and $\beta \wedge \delta$ are conjunctive clauses over assumables.

Properties of the and-or tree ensure that $cost(\alpha \wedge \beta) = cost(\alpha) \oplus cost(\beta)$.

[7] $\oplus$ is commutative, associative and has a zero element; $i \leq_\oplus j$ iff $i \oplus k = j$ for some k.

[8] Note that $([0, 1], *, \geq)$ where the cost of a literal is its probability does not satisfy the above conditions since $Pr(l) = 1$ or $Pr(\neg l) = 1$ does not hold for all literals l!

[9] The procedure returns partial diagnoses, which have to be completed using zero cost literals.

[10] Properties of the cost function ensure that

the argument for D is false, but the argument for A V D is okX.

Theorem 4 The sets I and J are independent given K (according to Definition 2) precisely when

$$(A6)\ \ Arg(\alpha \vee \beta \vee \gamma) \equiv Arg(\alpha \vee \gamma) \vee Arg(\beta \vee \gamma)$$

for all disjunctive clauses α, β and γ over $\mathbf{I}$, $\mathbf{J}$, *and* $\mathbf{K}$, *respectively.*[12]

In Figure 1, {B,C, E] are independent of (d-separated from) {A,D}. Thus,

$$\begin{aligned} & Arg(A \wedge B \wedge C \supset (\neg D \vee E)) \\ \equiv\ & Arg(\neg A \vee \neg B \vee \neg C \vee \neg D \vee E) \\ \equiv\ & Arg((\neg A \vee \neg D) \vee (\neg B \vee \neg C \vee E)) \\ \equiv\ & Arg(\neg A \vee \neg D) \vee Arg(\neg B \vee \neg C \vee E)\ \ (using\ A6) \\ \equiv\ & Arg(A \supset \neg D) \vee Arg(B \wedge C \supset E) \\ \equiv\ & okX \vee okY. \end{aligned}$$

Given a system description (A, P, A) and an observation o), let us define an abductive diagnosis as a diagnosis a that (together with the system description) logically entails the observation, $\Delta \cup \{\alpha\} \models \phi$.[13] Then:

Theorem 5 d is an abductwe diagnosis of system $(\Delta, \mathbf{P}, \mathbf{A}, \phi)$ *iff* $d \models Arg(\phi)$.

That is, the argument for observation 0 characterizes all its abductive diagnoses. Therefore, Theorem 3 is the basis for a dual set of results for computing abductive diagnoses.

7 Conclusion and Related Work

We have presented an approach for computing the most preferred diagnoses. We formally defined the class of system descriptions and the class of preference criteria to which the approach is applicable. We also characterized the computational guarantees it offers, which we believe are among the sharpest guarantees provided so far.

What is most important about our approach is that it ties the computational complexity of diagnostic reasoning to a very meaningful parameter: the topology of a system structure. Thus, it provides diagnostic practitioners with more flexibility in engineering the response time of their applications. This emphasis on structure has been the central theme in probabilistic reasoning lately [Pearl, 1988]. There have been some several attempts to import this theme into model-based diagnosis [Dechter and Dechter, 1994; Geffner and Pearl, 1987; Dechter and Dechter, 1988]. A number of structure-based algorithms have been provided for computing the most likely diagnoses, which seem to have similar computational complexity and appeal to the same underlying principles. Previous algorithms, however, have rested on the language of constraints among multivalued variables. A major contribution of this paper is (the symbolic) consequence calculus, which allows computation directly on boolean syntax. This not only simplifies structured-based algorithms significantly, but also provides a method for humans to compute diagnoses of non-trivial problems (as illustrated in Section 4).

Another important feature of the presented approach is the very simple and general mechanism for focusing on preferred diagnoses, which comes with useful guarantees. We are unaware of similar guarantees on the computational complexity of focusing using a mechanism as general as the one we have proposed.

[12] A disjunctive dause over atoms X is a disjunction of literals that includes one literal for each atom in X.

[13] An abductive diagnosis of observation is also called an explanation of ϕ.

References

[Darwiche and Pearl, 1994] Adnan Darwiche and Judea Pearl. Symbolic causal networks. In Proceedings of AAAI, pages 238 244. AAAI, 1994.

[Darwiche, 1992] Adnan Darwiche. A Symbolic Generalization of Probability Theory. PhD thesis, Stanford University, 1992.

[Darwiche, 1993] Adnan Darwiche. Argument calculus and networks. In Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI), pages 420-427, 1993".

[de Kleer et al, 1992] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. Artificial Intelligence, 56(2-3): 197—222, 1992.

[de Kleer, 1986] Johan de Kleer. An assumption-based TMS. Artificial Intelligence, 28:127-162, 1986.

[Dechter and Dechter, 1988] Rina Dechter and Avi Dechter. Belief maintenance in dynamic constraint networks. In Proceedings of AAAI, pages 37-42. AAAI, 1988.

[Dechter and Dechter, 1994] Rina Dechter and Avi Dechter. Structure-driven algorithms for truth maintenance. Artificial Intelligence, 1994. To appear.

[Forbus and de Kleer, 1993] Kenneth D. Forbus and Johan de Kleer. Building Problem Solvers. MIT Press, 1993.

[Freitag and Friedrich, 1992] Hartmut Freitag and Gerhard Friedrich. Focusing on independent diagnosis problems."In Proceedings of KR, pages 521-531, 1992.

[Geffner and Pearl, 1987] Hector Geffner and Judea Pearl. An improved constraint-propagation algorithm for diagnosis. In Proceedings of IJCA1, pages 1105-1111, Milan, Italy, 1987.

[Pearl, 1988] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.

[Reiter and de Kleer, 1987] Ray Reiter and Johan de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In Proceedings of AAAI, pages 183-188. AAAI, 1987.

[Reiter, 1987] Raymond Reiter. A theory of diagnosis from first principles. Artificial Intelligence, 32:57-95, 1987.