A Generalization of the GDE Minimal Hitting-Set Algorithm to Handle Behavioral Modes

Mattias Nyberg
Department of Electrical Engineering, Linköping University,
SE-581 83 Linköping, Sweden
Phone: +46-13285714, Fax: +46-13282035,
Email: matny@isy.liu.se

Abstract

A generalization of the minimal hitting-set algorithm given by deKleer and Williams is presented. The original algorithm handles only one faulty mode per component and only positive conflicts. In contrast, the new algorithm presented here handles more than two modes per component and also non-positive conflicts. The algorithm computes a logical formula that characterizes all diagnoses. Instead of minimal diagnoses, or kernel diagnoses, some specific conjunctions in the logical formula are used to characterize the diagnoses. These conjunctions are a generalization of both minimal and kernel diagnoses. From the logical formulas, it is also easy to derive the set of preferred diagnoses.

1 Introduction

Within the field of fault diagnosis, it has often been assumed that each component has only two possible behavioral modes, e.g. see [Reiter, 1987; deKleer and Williams, 1987]. For this case, and given a set of conflict sets, it is well known that a minimal hitting set corresponds to a minimal diagnosis [Reiter, 1987][1]. Algorithms for computing all minimal hitting sets have been presented in [Reiter, 1987; deKleer and Williams, 1987]. Improvements have later been given in e.g. [Greiner *et al.*, 1989; Wotawa, 2001].

In [Reiter, 1987; deKleer and Williams, 1987] it is assumed that a conflict can only imply that some component is faulty. We call this a *positive conflict* [deKleer *et al.*, 1992]. If all conflicts are positive, it is also well known that the set of all minimal diagnoses characterizes all diagnoses [deKleer and Williams, 1987]. This will for example be the case if the faulty modes of the components have no fault models. However, if there are fault models, it is possible to have non-positive conflicts implying that some component is fault-free.

If there is a desire to compute something that characterizes all diagnoses when there are non-positive conflicts, the concept of minimal hitting sets and the algorithms in [Reiter, 1987; deKleer and Williams, 1987] can not be used. To solve this, an alternative characterization based on so called *kernel diagnoses* was proposed in [deKleer *et al.*, 1992], where also an algorithm to compute the kernel diagnoses was given. The kernel diagnoses characterize all diagnoses even in the case of non-positive conflicts.

It has been noted in several papers that more than two possible behavioral modes are useful for improving the performance of the diagnostic system, see e.g. [Struss and Dressler, 1989; deKleer and Williams, 1989]. For this case, neither minimal diagnoses or kernel diagnoses can be used to characterize all diagnoses. Further, none of the algorithms in [Reiter, 1987; deKleer and Williams, 1987; deKleer *et al.*, 1992] are applicable.

To be able to handle both more than two behavioral modes and non-positive conflicts, the present paper proposes a new characterization of all diagnoses. Conflicts and diagnoses are represented by logical formulas, and instead of minimal diagnoses and kernel diagnoses, we use more general conjunctions on a specific form. In the special case of two behavioral modes per component, these conjunctions become equivalent to kernel diagnoses, and in the case of only positive conflicts, they become equivalent to minimal diagnoses. Thus, the here proposed framework can be seen as a generalization of both minimal diagnoses and kernel diagnoses.

Another contribution is that we show that the minimal hitting set algorithm given in [deKleer and Williams, 1987] can in fact be generalized to compute the here proposed characterization. Note that, even though the papers [Struss and Dressler, 1989; deKleer and Williams, 1989] consider more than two behavioral modes per component, they are, in contrast to the present paper, not concerned with the characterization or computation of all diagnoses.

Under the assumption of only two behavioral modes per component, the minimal diagnoses can be argued to be the most desired diagnoses. This has been called the parsimony principle, e.g. see [Reiter, 1987]. In the generalized case of more than two behavioral modes, the minimal diagnoses are no longer necessarily the most desired diagnoses. Instead the concept of *preferred diagnoses* has been defined in [Dressler and Struss, 1992]. We will in this paper show how to obtain these preferred diagnoses by means of the above mentioned logical formulas.

[1] Reiter used the word diagnosis for what in this paper is called minimal diagnosis.

The paper is organized as follows. In Section 2, the algorithm from [deKleer and Williams, 1987] is restated as a reference. In Section 3, the logical framework is presented. Then the generalized version of the algorithm from [deKleer and Williams, 1987] is given in Section 4. Sections 5 and 6 discuss the relation to minimal and kernel diagnoses. Finally, Section 7 describes how to compute the preferred diagnoses. All proofs of theorems have been placed in an appendix.

2 The Original Algorithm

This section presents the original algorithm and its associated framework as presented in [deKleer and Williams, 1987]. However, since we have a different objective than in the original paper, we will not always use the same notation and naming convention.

The system to be diagnosed is assumed to consist of a number of components represented by a set $\mathcal{C}$. A *conflict* is represented as a set $C \subseteq \mathcal{C}$. The meaning of a conflict C is that not all components in C can be in the normal fault-free mode. Thus only positive conflicts can be handled. A conflict C_1 is said to be *minimal* if there is no other conflict C_2 such that $C_2 \subset C_1$.

A *diagnosis* δ is also represented as a set $\delta \subseteq \mathcal{C}$. The meaning of a diagnosis δ is that the components contained in δ are faulty and the components not contained in δ are fault free. A diagnosis δ_1 is said to be *minimal* if there is no other diagnosis δ_2 such that $\delta_2 \subset \delta_1$.

One fundamental relation between conflicts and diagnoses is that if $\mathbb{C}$ is the set of all minimal conflicts, δ is a diagnosis if and only if for all conflicts $C \in \mathbb{C}$ it holds that $\delta \cap C \neq \emptyset$.

Given a set of diagnoses Δ and a conflict C the minimal hitting set algorithm in [deKleer and Williams, 1987] finds an updated set of minimal diagnoses. A version of the algorithm, as described in the text of [deKleer and Williams, 1987], can be written as follows.

Algorithm 1

```
 Input: a set of minimal diagnoses $\Delta$, and a conflict set $C$
Output: the updated set of minimal diagnoses $\Theta$
$\Delta_{old} = \Delta$
forall $\delta_i \in \Delta$ do
    if $\delta_i \cap C = \emptyset$ then
        Remove $\delta_i$ from $\Delta_{old}$
        forall $c \in C$ do
            $\delta_{new} := \delta_i \cup \{c\}$
            forall $\delta_k \in \Delta$, $\delta_k \neq \delta_i$ do
                if $\delta_k \subseteq \delta_{new}$ then goto LABEL1
            end
            $\Delta_{add} := \Delta_{add} \cup \{\delta_{new}\}$
            LABEL1
        end
    end
end
$\Theta := \Delta_{old} \cup \Delta_{add}$
```

The algorithm has the properties that if Δ is the set of all minimal diagnoses, the algorithm output Θ will contain all minimal diagnoses with respect to also the new conflict C. Further, it also holds that Θ will contain only minimal diagnoses. Note that this algorithm does not require the conflict C to be minimal, contrary to what has been stated in [Greiner *et al.*, 1989]. It can also be noted that the loop over $\delta_k \in \Delta$ could be modified to $\delta_k \in \Delta_{old}$, which would be more efficient since Δ_{old} is smaller than Δ.

3 A Logical Framework

Each component is assumed to be in exactly one out of several behavioral modes. A behavioral mode can be for example no-fault, abbreviated NF, gain-fault G, bias B, open circuit OC, short circuit SC, unknown fault UF, or just faulty F. For our purposes, each component is abstracted to a variable specifying the behavioral mode of that component. Let $\mathcal{C}$ denote the set of such variables. For each component variable c let $\mathbf{R}_c$ denote the *domain* of possible behavioral modes, i.e. $c \in \mathbf{R}_c$.

We will now define a set of formulas to be used to express that certain components are in certain behavioral modes. If c is a component variable in the set $\mathcal{C}$ and $M \subseteq \mathbb{R}_c$, the expression $c \in M$ is a formula. For example, if p is a pressure sensor, the formula $p \in \{NF, G, UF\}$ means that the pressure sensor is in mode NF, G, or UF. If M is a singleton, e.g. $M = \{NF\}$, we will sometimes write also $p = NF$. Further, the constant $\bot$ with value *false*, is a formula. If ϕ and γ are formulas then $\phi \wedge \gamma$, $\phi \vee \gamma$, and $\neg\phi$ are formulas.

In accordance with the theory of first order logic we say that a formula ϕ is a semantic consequence of another formula γ, and write $\gamma \models \phi$, if all assignments of the variables $\mathcal{C}$ that make γ true also make ϕ true. This can be generalized to sets of formulas, i.e. $\{\gamma_1, \dots, \gamma_n\} \models \{\phi_1, \dots, \phi_m\}$ if and only if $\gamma_1 \wedge \dots \wedge \gamma_n \models \phi_1 \wedge \dots \wedge \phi_m$. If it holds that $\Gamma \models \Phi$ and $\Phi \models \Gamma$, where Φ and Γ are formulas or sets of formulas, Φ and Γ are said to be equivalent and we write $\Gamma \simeq \Phi$.

We will devote special interest to conjunctions on the form

$$c_1 \in M_1 \wedge c_2 \in M_2 \wedge \dots \wedge c_n \in M_n \qquad (1)$$

where all components are unique, i.e. $c_i \not\equiv c_j$ if $j \neq k$, and each M_i is a nonempty proper subset of $\mathbf{R}_{c_i}$, i.e. $\emptyset \neq M_i \subset \mathbf{R}_{c_i}$. Let D_i denote a conjunction on the form (1). From a set of such conjunctions we can then form a disjunction

$$D_1 \vee D_2 \vee \dots D_m \qquad (2)$$

Note that the different conjunctions D_i can contain different number of components. We will say that a formula is in *maximal normal form* MNF if it is on the form (2) and has the additional property that no conjunction is a consequence of another conjunction, i.e. for each conjunction D_i, there is no conjunction D_j, $j \neq i$, for which it holds that $D_j \models D_i$. Note that the purpose of using formulas in MNF is that they are relatively compact in the sense that an MNF-formula does not contain redundant conjunctions and that each conjunction does not contain redundant assignments.

For an example consider the following two formulas containing pressure sensors p_1, p_2, and p_3, where all have the behavioral modes $\mathbb{R}_{p_i} = \{NF, G, B, UF\}$.

$$p_1 \in \{UF\} \wedge p_2 \in \{B, UF\} \vee p_3 \in \{UF\}$$
$$p_1 \in \{UF\} \wedge p_2 \in \{B, UF\} \vee p_1 \in \{G, UF\}$$

The first formula is in MNF but not the second since $p_1 \in \{UF\} \wedge p_2 \in \{B, UF\} \models p_1 \in \{G, UF\}$.

3.1 Conflicts and Diagnoses

A conflict is assumed to be written using the logical language defined above. For example, if has been found that the pressure sensor p_1 can not be in the mode NF at the same time as p_2 is in the mode B or NF, this gives the conflict

$$H = p_1 \in \{NF\} \wedge p_2 \in \{B, NF\} \tag{3}$$

To relate this definition of conflict to the one used in Section 2, consider the conflict $C = \{a, b, c\}$. With the logical language, we can write this conflict as $a \in \{NF\} \wedge b \in \{NF\} \wedge c \in \{NF\}$.

Instead of conflicts, we will mostly use negated conflicts, so instead of H we consider $\neg H$. In particular we will use negated conflicts written in MNF. For an example, the negated conflict $\neg H$, where H is defined as in (3), can be written in MNF as $p_1 \in \{G, B, UF\} \vee p_2 \in \{G, UF\}$. Without loss of generality, we will from now on assume that all negated conflicts are written on the form

$$c_1 \in M_1 \vee c_2 \in M_2 \vee \cdots \vee c_n \in M_n \tag{4}$$

where $c_j \not\equiv c_k$ if $j \neq k$, and $\emptyset \neq M_i \subset \mathbf{R}_{c_i}$. This means that (4) is in MNF.

A *system behavioral mode* is a conjunction containing a unique assignment of all components in $\mathcal{C}$. For example if $\mathcal{C} = \{p_1, p_2, p_3\}$, a system behavioral mode could be

$$p_1 = UF \wedge p_2 = B \wedge p_3 = NF$$

We consider the term *diagnosis* to refer to a system behavioral mode consistent with all negated conflicts. More formally, if $\mathbb{P}$ is the set of all negated conflicts, a system behavioral mode d is a *diagnosis* if $\{d\} \cup \mathbb{P} \not\models \bot$ or equivalently $d \models \mathbb{P}$.

To relate this definition of diagnosis to the one used in Section 2, assume that $\mathcal{C} = \{a, b, c, d\}$ and consider the diagnosis $\delta = \{a, b\}$. With the logical language, we can write this diagnosis as $a = F \wedge b = F \wedge c = NF \wedge d = NF$.

4 The Generalized Algorithm

With only small modifications, the original algorithm stated in Section 2 can be made to work with logical MNF-formulas instead of sets. The result is an algorithm that handles more than two behavioral modes per component and also non-positive conflicts. With the modification, the algorithm will take as inputs, a formula $\mathcal{D}$ and a negated conflict $\mathcal{P}$, both written in MNF. The purpose of the algorithm is then to derive a new formula $\mathcal{Q}$ in MNF such that $\mathcal{Q} \simeq \mathcal{D} \wedge \mathcal{P}$.

The modifications are the following:

- Instead of using a set of minimal diagnoses Δ as input, use a formula $\mathcal{D}$ in MNF. Note that $\mathcal{D}$ is not restricted to be a disjunction of system behavioral modes, but instead can be a disjunction of conjunctions on the form (1).
- Instead of using a conflict set C as input, use a negated conflict $\mathcal{P}$ on the form (4).
- Instead of checking the condition $\delta_i \cap C = \emptyset$, check the condition $D_i \not\models \mathcal{P}$.
- Instead of the assignment $\delta_{new} := \delta_i \cup \{c\}$, find a conjunction D_{new} in MNF such that $D_{new} \simeq D_i \wedge P_j$.
- Instead of checking the condition $\delta_k \subseteq \delta_{new}$, check the condition $D_{new} \models D_k$.

In the algorithm we will use the notation $D_i \in \mathcal{D}$ to denote the fact that D_i is a conjunction in $\mathcal{D}$. The algorithm can now be stated as follows:

Algorithm 2

```
Input: a formula $\mathcal{D}$ in MNF, and a negated conflict $\mathcal{P}$
Output: $\mathcal{Q}$
$\mathcal{D}_{old} = \mathcal{D}$
forall $D_i \in \mathcal{D}$ do
    if $D_i \not\models \mathcal{P}$ then
        Remove $D_i$ from $\mathcal{D}_{old}$
        forall $P_j \in \mathcal{P}$ do
            Let $D_{new}$ be a conjunction in MNF such
                that $D_{new} \simeq D_i \wedge P_j$
            forall $D_k \in \mathcal{D}$, $D_k \neq D_i$ do
                if $D_{new} \models D_k$ then goto LABEL1
            end
            $\mathcal{D}_{add} := \mathcal{D}_{add} \vee D_{new}$
            LABEL1
        end
    end
end
$\mathcal{Q} := \mathcal{D}_{old} \vee \mathcal{D}_{add}$
```

To keep the algorithm description "clean", some operations have been written in a simplified form. More details are discussed in Section 4.2 below. Note that an improvement corresponding to the change of Δ to Δ_{old} in Algorithm 1 is not possible for the generalized algorithm.

The algorithm is assumed to be used in an iterative manner as follows. First when only one conflict $\mathcal{P}_1$ is considered, the diagnoses are already described by $\mathcal{P}_1$. Thus, the algorithm is not needed. When a second conflict $\mathcal{P}_2$ is considered, the algorithm is fed with $\mathcal{D} = \mathcal{P}_1$ and $\mathcal{P} = \mathcal{P}_2$, and produces the output $\mathcal{Q}$ such that $\mathcal{Q} \simeq \mathcal{P}_1 \wedge \mathcal{P}_2$. Then, for each additional conflict $\mathcal{P}_n$ that is considered, the input $\mathcal{D}$ is the old output $\mathcal{Q}$.

When the algorithm is used in this way, the following results can be guaranteed.

Theorem 1 *Let $\mathbb{P}$ be a set of negated conflicts that is not inconsistent, i.e. $\mathbb{P} \not\models \bot$, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Then it holds that $\mathcal{Q} \simeq \mathbb{P}$.*

Theorem 2 *The output $\mathcal{Q}$ from Algorithm 2 is in MNF.*

The proofs for these results can be found in the appendix.

4.1 Example

To illustrate the algorithm, consider the following small example where $\mathcal{C} = \{p_1, p_2, p_3\}$ and the domain of behavioral modes for each component is $\mathbb{R}_{p_i} = \{NF, G, B, UF\}$:

$$\mathcal{D} = D_1 \vee D_2 = p_1 \in \{G, B, UF\} \vee p_3 \in \{G, UF\}$$
$$\mathcal{P} = P_1 \vee P_2 = p_2 \in \{B, UF\} \vee p_3 \in \{G, B, UF\}$$

First the condition $D_1 \not\models \mathcal{P}$ is fulfilled which means that D_1 is removed from $\mathcal{D}_{old}$ and the inner loop of the algorithm is entered. There a D_{new} is created such that $D_{new} \simeq D_1 \wedge$

$P_1 = p_1 \in \{G, B, UF\} \wedge p_2 \in \{B, UF\}$. This D_{new} is then compared to D_2 in the condition $D_{new} \models D_2$. The condition is not fulfilled which means that D_{new} is added to $\mathcal{D}_{add}$. Next a D_{new} is created such that $D_{new} \simeq D_1 \wedge P_2 = p_1 \in \{G, B, UF\} \wedge p_3 \in \{G, B, UF\}$. Also this time the condition $D_{new} \models D_2$ is not fulfilled, implying that D_{new} is added to $\mathcal{D}_{add}$. Next, the conjunction D_2 is investigated but since $D_2 \models \mathcal{P}$ holds, D_2 is not removed from $\mathcal{D}_{old}$ and the inner loop is not entered. The algorithm output is finally formed as

$$\begin{aligned}\mathcal{Q} &:= \mathcal{D}_{old} \vee \mathcal{D}_{add} = D_2 \vee (D_1 \wedge P_1 \vee D_1 \wedge P_2) = \\ &= p_3 \in \{G, UF\} \vee p_1 \in \{G, B, UF\} \wedge p_2 \in \{B, UF\} \vee \\ &\quad \vee p_1 \in \{G, B, UF\} \wedge p_3 \in \{G, B, UF\}\end{aligned}$$

It can be verified that $\mathcal{Q} \simeq \mathcal{D} \wedge \mathcal{P}$. Also, it can be seen that $\mathcal{Q}$ is in MNF.

4.2 Algorithm Details

To implement the algorithm, some more details need to be known. The first is how to check the condition $D_i \models \mathcal{P}$. To illustrate this, consider an example where D_i contains components c_1, c_2, and c_3 and $\mathcal{P}$ components c_2, c_3, and c_4. Since $\mathcal{D}$ is in MNF, and $\mathcal{P}$ in the form (4), D_i and $\mathcal{P}$ will have the form

$$D_i = c_1 \in M_1^D \wedge c_2 \in M_2^D \wedge c_3 \in M_2^D \quad (5)$$

$$\mathcal{P} = c_2 \in M_2^P \vee c_3 \in M_3^P \vee c_4 \in M_4^P \quad (6)$$

We realize that the condition $D_i \models \mathcal{P}$ holds if and only if $M_2^D \subseteq M_2^P$ or $M_3^D \subseteq M_3^P$. Thus, this example shows that in general, $D_i \models \mathcal{P}$ holds if and only if D_i and $\mathcal{P}$ contain at least one common component c_i where $M_i^D \subseteq M_i^P$.

The second detail is how to find an expression Q_{new} in MNF such that $Q_{new} \simeq D_i \wedge P_j$. To illustrate this, consider an example where D_i contains components c_1 and c_2, and P_j the component c_2. Since $\mathcal{D}$ is in MNF, and $\mathcal{P}$ in the form (4), D_i and P_j will have the form

$$D_i = c_1 \in M_1^D \wedge c_2 \in M_2^D \quad (7a)$$

$$P_j = c_2 \in M_2^P \quad (7b)$$

Then Q_{new} will be formed as $D_{new} = c_1 \in M_1^D \wedge c_2 \in M_2^D \cap M_2^P$ which means that $D_{new} \simeq D_i \wedge P_j$. If it holds that $M_2^D \cap M_2^P \neq \emptyset$, D_{new} will be in MNF. Otherwise let $D_{new} = \perp$. The check $D_{new} \models D_k$ will then immediately make the algorithm jump to *LABEL1* meaning that D_{new} will not be added to $\mathcal{D}_{add}$.

The third detail is how to check the condition $D_{new} \models D_k$. To illustrate this, consider an example where D_{new} contains components c_1 and c_2, and D_k the components c_2 and c_3. Since D_{new} and $\mathcal{D}$ are both in MNF, D_{new} and D_k will have the form

$$D_{new} = c_1 \in M_1^n \wedge c_2 \in M_2^n \quad (8a)$$

$$D_k = c_2 \in M_2^D \wedge c_3 \in M_3^D \quad (8b)$$

Without changing their meanings, these expressions can be expanded so that they contain the same set of components:

$$D'_{new} = c_1 \in M_1^n \wedge c_2 \in M_2^n \wedge c_3 \in \mathbf{R}_{c_3} \quad (9)$$

$$D'_k = c_1 \in \mathbf{R}_{c_1} \wedge c_2 \in M_2^D \wedge c_3 \in M_3^D \quad (10)$$

Now we see that the condition $D_{new} \models D_k$ holds if and only if $M_1^n \subseteq \mathbf{R}_{c_1}$, $M_2^n \subseteq M_2^D$, and $\mathbf{R}_{c_3} \subseteq M_3^D$. The first of these three conditions is always fulfilled and the third can never be fulfilled since, by definition of MNF, $M_3^D \subset \mathbf{R}_{c_3}$. Thus, this example shows that $D_{new} \models D_k$ holds if and only if (1), D_k contains only components that are also contained in D_{new}, and (2), for all components c_i contained in both D_{new} and D_k it holds that $M_i^n \subseteq M_i^D$.

The fourth detail to be considered is the expression $\mathcal{D}_{add} := \mathcal{D}_{add} \vee D_{new}$. Since $\mathcal{D}_{add}$ is not assigned from the beginning, this expression is to be read as $\mathcal{D}_{add} := D_{new}$ when $\mathcal{D}_{add}$ is unassigned.

Finally, note that $\mathcal{D}_{old}$ or $\mathcal{D}_{add}$ may be unassigned or empty at some places in the algorithm. In that case, e.g. in $\mathcal{Q} := \mathcal{D}_{old} \vee \mathcal{D}_{add}$, the missing term can just be neglected.

5 Relation to Minimal Diagnoses

The concept of minimal diagnoses was originally proposed in [Reiter, 1987; deKleer and Williams, 1987] for systems where each component has only two possible behavioral modes, i.e. the normal fault-free mode and a faulty mode. Minimal diagnoses have two attractive properties. Firstly, they represent the "simplest" diagnoses and are therefore often desired when prioritizing among diagnoses. Secondly, in case there are only positive conflicts, the minimal diagnoses characterize the set of all diagnoses. These two properties will now be investigated for the generalized case of more than two modes per component and non-positive conflicts.

5.1 "Simplest" Property

For the case of more than two modes per component, the concept of *preferred diagnoses* was defined in [Dressler and Struss, 1992] as a generalization of minimal diagnoses. The basic idea is that the behavioral modes for each component are ordered in a partial order defining that some behavioral modes are more preferred than other. For example, NF is usually preferred over any other mode, and a simple electrical fault, such as short-cut or open circuit, may be preferred over other more complex behavioral modes. Further, an unknown fault UF may be the least preferred mode.

For a formal definition let $b_c^1 \geq_c b_c^2$ denote the fact that for component c, the behavioral mode b_c^1 is equally or more preferred than b_c^2. For each component, this relation forms a partial order on the behavioral modes. Further, these relations induce a partial order on the system behavioral modes. Let d_1 and d_2 be two system behavioral modes $d_i = \wedge_{c \in \mathcal{C}}(c = b_c^i)$. Then we write $d_1 \geq d_2$ if for all $c \in \mathcal{C}$ it holds that $b_c^1 \geq_c b_c^2$. A preferred diagnosis can then formally be defined as a diagnosis d such that there is no other diagnosis d' where $d' > d$. In Section 7 we will discuss how the preferred diagnoses can be obtained from an MNF formula representing all diagnoses. Note that in the case of only two modes, preferred diagnoses are exactly the minimal diagnoses.

Remark: One may ask what "preferred" or "simplest" diagnoses means. One possible formal justification is the following. Let $P(d)$ denote the prior probability of the system behavioral mode $d = \wedge_{c \in \mathcal{C}} c = b_c$. We assume that faults occur independently of each other which means that $P(d) =$

$\prod_{c \in \mathcal{C}} P(c = b_c)$ where $P(c = b_c)$ is the prior probability that component c is in behavioral mode b_c. If $\mathcal{Q}$ is a formula such that $\mathcal{Q} \simeq \mathbb{P}$, it holds that $P(d|\mathbb{P}) = P(d \wedge \mathcal{Q})/P(\mathcal{Q})$. This means that $P(d|\mathbb{P}) = P(d)/P(\mathcal{Q})$ if $d \models \mathbb{P}$, i.e. if d is a diagnosis, and $P(d|\mathbb{P}) = 0$ if $d \not\models \mathbb{P}$, i.e. if d is not a diagnosis. For a given set $\mathbb{P}$, the term $P(\mathcal{Q})$ is only a normalization constant, which means that to compare $P(d|\mathbb{P})$ for different diagnoses it is enough to consider the priors $P(d)$. To know the exact value of a prior $P(c = b_c)$ may be very difficult or even impossible. Therefore one may assume that for each component, the priors are unknown but at least partially ordered. Under this assumption, and given the set of negated conflicts, the preferred diagnoses are then the most probable ones.

5.2 Characterizing Property

Now we investigate how the characterizing property of minimal diagnoses can be generalized to the case of more than two modes and the presence of non-positive conflicts. In some special cases, the preferred diagnoses characterize all diagnoses with the help of the partial order $\geq$. That is, if d_1 is a diagnosis and if $d_2 < d_1$, we know that also d_2 is a diagnosis. This is always true when there are only two modes per component and only positive conflicts, which in turn is guaranteed when there are no fault models. Note that it may also be true in a case with more than two modes, even in the presence of fault models. However this does not hold generally.

In an MNF-formula, the conjunctions have the property that they characterize all diagnoses. For example consider the case when the components are $=\{a, b, c, d, e\}$, $\mathbf{R} = \{NF, B, G, UF\}$ for all components, and $a \in \{B, UF\} \wedge b \in \{G, UF\}$ is one of the conjunctions in an MNF formula. By letting each diagnosis be represented as an ordered set corresponding to $\langle a, b, c, d, e \rangle$, this single conjunction characterizes the diagnoses

$$\{B, UF\} \times \{G, UF\} \times \{NF, B, G, UF\} \times \\ \times \{NF, B, G, UF\} \times \{NF, B, G, UF\}$$

which is 256 diagnoses.

For another example assume that each of the components $\mathcal{C} = \{a, b, c, d\}$ has only two modes, i.e. $\mathbf{R} = \{NF, F\}$. A conjunction $a \in \{F\} \wedge b \in \{F\}$ would then characterize all diagnoses $\{F\} \times \{F\} \times \{NF, F\} \times \{NF, F\}$. In Section 2 this conjunction would be represented by $\{a, b\}$. If all conflicts are positive, all conjunctions would be on this form, and there is a one-to-one correspondence between the conjunctions in an MNF-formula and the minimal diagnoses in the original framework described in Section 2.

If there is a fault model for the mode F of a component a, the non-positive conflict $a \in \{F\}$ may appear. Assume also that a conflict $b = \{NF\}$ appears. This has the consequence that a formula in MNF, describing all diagnoses, may for example contain a conjunction $a \in \{NF\} \wedge b \in \{F\}$. This conjunction characterizes all diagnoses $\{NF\} \times \{F\} \times \{NF, F\} \times \{NF, F\}$, and this is a so called *kernel diagnosis* (see the next section). Note that to represent this conjunction is not possible using sets as described in Section 2. Note also that there is one minimal diagnosis in this example, namely $a = NF \wedge b = F \wedge c = NF \wedge d = NF$, and this minimal diagnosis does not characterize all diagnoses.

6 Relation to Kernel Diagnoses

The paper [deKleer *et al.*, 1992] defines *partial diagnosis* and *kernel diagnosis*. This was done assuming only two modes per component. The purpose of kernel diagnoses is that the set of all kernel diagnoses characterizes all diagnoses even in the case when there are non-positive conflicts. As noted in [deKleer *et al.*, 1992], also a subset of kernel diagnoses is sometimes sufficient to characterize all diagnoses.

In the context of this paper we can define partial diagnosis as a conjunction d of mode assignments such that $d \models \mathbb{P}$. Then, a kernel diagnosis is partial diagnosis d such that there is no other partial diagnosis d' where $d \models d'$.

According to the following theorem, the output $\mathcal{Q}$ from Algorithm 2 is, in the two-mode case, a disjunction of kernel diagnoses.

Theorem 3 *Let each component have only two possible behavioral modes, let $\mathbb{P}$ be a set of negated conflicts, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Then it holds that each conjunction of $\mathcal{Q}$ is a kernel diagnosis.*

Note that the MNF property alone does not guarantee that all conjunctions are kernel diagnoses. This can be seen in the following formula which is in MNF.

$$c_1 = N \wedge c_2 = N \vee c_1 = N \wedge c_2 = F \qquad (11)$$

All diagnoses represented by (11) are characterized by the single kernel diagnosis $c_1 = N$. Therefore none of the conjunctions in (11) are kernel diagnoses.

Even though the paper [deKleer *et al.*, 1992] defines partial and kernel diagnoses for the case of only two modes per component, the definition of partial and kernel diagnoses given above is applicable also to the case of more than two modes per component. However, the conjunctions in the output $\mathcal{Q}$ from Algorithm 2 will for this case not be kernel diagnoses. Instead each conjunction represents a set of partial diagnoses, e.g. the first conjunction of (12) represents the two partial diagnoses $c_1 = E \wedge c_3 = B$ and $c_1 = E \wedge c_3 = G$. Since the second conjunction of (12) represents e.g. $c_1 = E \wedge c_2 = E \wedge c_3 = B$, it is also obvious that the partial diagnoses represented by each conjunction are not necessarily kernel diagnoses.

7 Extracting Preferred Diagnoses

In Section 5 it was concluded that the conjunctions in the output $\mathcal{Q}$ from Algorithm 2 characterize all diagnoses, and in the special case of two modes per component and only positive conflicts, there is a one-to-one correspondence between MNF-conjunctions and the minimal diagnoses. This special case has also the property that if we study each conjunction in an MNF formula $\mathcal{Q}$ separately, it will have only one preferred diagnosis. This preferred diagnosis is a also a preferred diagnosis when considering the whole formula $\mathcal{Q}$. The consequence is that it is straightforward to extract the preferred diagnosis from a formula $\mathcal{Q}$. In the general case, there is no such guarantee. For example, in the two-mode case and when some conflicts are non-positive, which means that the negated conflict will contain some assignment $c = NF$, there may be a conjunction not corresponding to a preferred diagnosis.

For an example with more than two modes, consider two components c_1 and c_2 where $\mathbf{R}_{c_i} = \{NF, E, F\}$ and $NF >_{c_i} E >_{c_i} F$, and a third component c_3 where $\mathbf{R}_{c_i} = \{NF, B, G\}$ with the only relations $NF >_{c_3} B$ and $NF >_{c_3} G$. Then consider the MNF-formula

$$\begin{aligned} \mathcal{Q} = c_1 \in \{E\} \wedge c_3 \in \{B, G\} \vee \\ c_1 \in \{E, F\} \wedge c_2 \in \{E, F\} \wedge c_3 \in \{B, G\} \end{aligned} \quad (12)$$

The preferred diagnoses consistent with the first conjunction are $c_1 = E \wedge c_2 = NF \wedge c_3 = B$ and $c_1 = E \wedge c_2 = NF \wedge c_3 = G$. The preferred diagnoses consistent with the second are $c_1 = E \wedge c_2 = E \wedge c_3 = B$ and $c_1 = E \wedge c_2 = E \wedge c_3 = G$. As seen, the two diagnoses $c_1 = E \wedge c_2 = E \wedge c_3 = B$ and $c_1 = E \wedge c_2 = E \wedge c_3 = G$ are not preferred diagnoses of the whole formula $\mathcal{Q}$.

The example shows that preferred diagnoses can not be extracted simply by considering one conjunction at a time. Instead the following procedure can be used. For each conjunction in $\mathcal{Q}$, find the preferred diagnoses consistent with that conjunction, and collect all diagnoses found in a set Ψ. The set Ψ may contain non-preferred diagnoses. These can be removed by a simple pairwise comparison. Note that the set Ψ need not to be calculated for every new negated conflict that is processed. Instead only at the time the preferred diagnoses are really needed, for example before a service task is to be carried out, the set Ψ needs to be calculated.

One may ask how much extra time that is needed for the computation of the preferred diagnoses, compared to the time needed to process all negated conflicts and compute $\mathcal{Q}$. To give an indication of this, the following empirical experiment was set up. A number of 132 test cases were randomly generated. The test cases represent systems with between 4 and 7 components, where each component has 4 possible behavioral modes. The number of negated conflicts varies between 2 and 12.

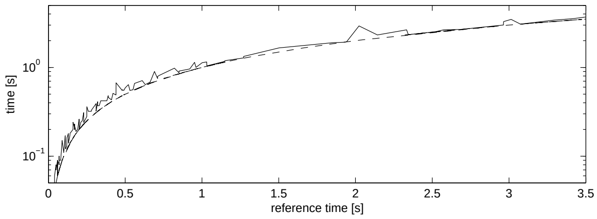

Figure 1: The total execution times for computing $\mathcal{Q}$ (dashed line) and preferred diagnoses (solid line).

In Figure 1, the results for the 132 test cases are shown. The reference time on the x-axis is chosen to be the computation time needed to compute $\mathcal{Q}$. As seen, the figure indicates that the extra time needed to compute preferred diagnoses from the MNF formula $\mathcal{Q}$, is almost negligible compared to the time needed to compute only the MNF formula.

8 Conclusions

In this paper the minimal hitting-set algorithm from [deKleer and Williams, 1987] has been generalized to handle more than two modes per component and also non-positive conflicts. This has been done by first establishing a framework where all conflicts and diagnoses are represented with special logical formulas. Then the original minimal hitting-set algorithm needed only small modifications to obtain the desired results. It has been formally proven that $\mathcal{Q} \simeq \mathbb{P}$, i.e. the algorithm output is equivalent to the set of all diagnoses. Further it was proven that the algorithm output $\mathcal{Q}$ is in the MNF-form that guarantees that $\mathcal{Q}$ does not contain redundant conjunctions.

In a comparison with the original framework where conflicts and diagnoses are represented by sets, it was concluded that the conjunctions in the output $\mathcal{Q}$, from the generalized algorithm, are a true generalization of the minimal diagnoses obtained from the minimal hitting-set algorithm. It has also been concluded that the conjunctions are a true generalization of kernel diagnoses. Since, for the case of more than two mode per component, minimal diagnoses do not necessarily correspond to the most desired diagnoses, it was instead shown how preferred diagnoses could be obtained from the conjunctions with a reasonable amount of effort.

References

[deKleer and Williams, 1987] J. deKleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, Issue 1, Volume 32:pp. 97–130, 1987.

[deKleer and Williams, 1989] J. deKleer and B.C. Williams. Diagnosis with behavioral modes. IJCAI, pages 1324–1330, 1989.

[deKleer *et al.*, 1992] J. deKleer, A.K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, Issue 2-3, Volume 56:pp. 197–222, 1992.

[Dressler and Struss, 1992] O. Dressler and P. Struss. Back to defaults: Characterizing and computing diagnoses as coherent assumption sets. ECAI, pages 719–723, 1992.

[Greiner *et al.*, 1989] R. Greiner, B.A. Smith, and R.W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41(1):79–88, 1989.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

[Struss and Dressler, 1989] P. Struss and O. Dressler. 'physical negation' - integrating fault models into the general diagnosis engine. IJCAI, pages 1318–1323, 1989.

[Wotawa, 2001] F. Wotawa. A variant of reiter's hitting-set algorithm. *Information Processing Letters*, 79(1):45–51, 2001.

Appendix

Lemma 1 *The output $\mathcal{Q}$ from Algorithm 2 contains no two conjunctions such that $Q_2 \models Q_1$.*

PROOF. Assume the contrary, that Q_1 and Q_2 are two conjunctions in $\mathcal{Q}$ and $Q_2 \models Q_1$. There are three cases that need to be investigated: (1) $Q_1 \in \mathcal{D}_{old}$, $Q_2 \in \mathcal{D}_{add}$, (2) $Q_2 \in \mathcal{D}_{old}$, $Q_1 \in \mathcal{D}_{add}$, (3) $Q_1 \in \mathcal{D}_{add}$, $Q_2 \in \mathcal{D}_{add}$.

1) The fact $Q_2 \in \mathcal{D}_{add}$ means that $D_{new} = Q_2$ at some point. Since $Q_1 \in \mathcal{D}_{old}$, D_{new} must then have been compared to Q_1. Since Q_2 has really been added, it cannot have been the case that $Q_2 \models Q_1$.

2) Since $Q_1 \in \mathcal{D}_{add}$, it holds that $Q_1 = D_i \wedge P_j$ for some $D_i \in \mathcal{D}$. The fact $Q_2 \models Q_1$ implies that $Q_2 \models D_i \wedge P_j \models D_i$. This is a contradiction since $Q_2 \in \mathcal{D}$, and $\mathcal{D}$ is in MNF.

3) There are three cases: (a) $Q_2 = D_i \wedge P_{j2} \models D_i \wedge P_{j1} = Q_1$, (b) $Q_2 = D_{i2} \wedge P_j \models D_{i1} \wedge P_j = Q_1$, (c) $Q_2 = D_{i2} \wedge P_{j2} \models D_{i1} \wedge P_{j1} = Q_1$, where in all cases, $P_{j1} \neq P_{j2}$ and $D_{i1} \neq D_{i2}$.

   a) We know that D_i and $\mathcal{P}$ are formulas on forms like $D_i = a \in A \wedge b \in B \wedge c \in C$ and $\mathcal{P} = a \in A_p \vee b \in B_p$ respectively. This means that $Q_1 = a \in A \cap A_p \wedge b \in B \wedge c \in C$ and $Q_2 = a \in A \wedge b \in B \cap B_p \wedge c \in C$. The fact $Q_2 \models Q_1$ implies that $A \subseteq A \cap A_p$ which further means that $A \subseteq A_p$. This implies $D_i = a \in A \wedge b \in B \wedge c \in C \models a \in A_p \models \mathcal{P}$. Thus, Q_1 and Q_2 are never subject to be added to $\mathcal{D}_{add}$.

   b) We have that $Q_2 = D_{i2} \wedge P_j \models D_{i1} \wedge P_j \models D_{i1} \in \mathcal{D}$. This means that $Q_2 = D_{i2} \wedge P_j$ can not have been added to $\mathcal{D}_{add}$.

   c) We have that $Q_2 = D_{i2} \wedge P_{j2} \models D_{i1} \wedge P_{j1} \models D_{i1} \in \mathcal{D}$. This means that $Q_2 = D_{i2} \wedge P_{j2}$ can not have been added to $\mathcal{D}_{add}$.

All these investigations show that it impossible that $Q_2 \models Q_1$. □

Theorem 2 *The output $\mathcal{Q}$ from Algorithm 2 is in MNF.*

PROOF. From Lemma 1 it follows that $\mathcal{Q}$ contains no two conjunctions such that $Q_2 \models Q_1$. All conjunctions in $\mathcal{D}_{old}$ are trivially on the form specified by (1). All conjunctions in $\mathcal{D}_{add}$ are also on the form (1) because of the requirement on D_{new}. Thus $\mathcal{Q}$ is in MNF. □

Lemma 2 *Let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. For any two conjunctions Q_1 and Q_2 in $\mathcal{Q}$, there is no component c and conjunction $\bar{D}$ such that $Q_1 \simeq \bar{D} \wedge c \in A_1$ and $Q_2 \simeq \bar{D} \wedge c \in A_2$ where $A_1 \subseteq \mathbf{R}_c$ and $A_2 \subseteq \mathbf{R}_c$.*

PROOF. Assume that there is a component c and conjunction $\bar{D}$ such that $Q_1 \simeq \bar{D} \wedge c \in A_1$ and $Q_2 \simeq \bar{D} \wedge c \in A_2$. We can write Q_1 as $c \in A^{\phi_1} \wedge \bar{D}_1$ where A^{ϕ_1} is the intersection of the sets M_j obtained from all $P \in \phi_1 \subseteq \mathbb{P}$, and $\bar{D}_1$ is the conjunction of one P_j obtained from every $P \in \mathbb{P} \setminus \phi_1$. Similarly we write Q_2 as $c \in A^{\phi_2} \wedge \bar{D}_2$.

We can find a D' such that $D' \simeq \bar{D}_1 \simeq \bar{D}_2$ and where D' is the conjunction of one P_j obtained from every $P \in \mathbb{P} \setminus (\phi_1 \cap \phi_1)$. Then let $D^* = c \in A^{\phi_1 \cap \phi_2} \wedge D'$ which means that $Q_1 \models c \in A^{\phi_1 \cap \phi_2} \wedge \bar{D}_1 \simeq D^*$. Similarly we can obtain the relation $Q_2 \models c \in A^{\phi_1 \cap \phi_2} \wedge \bar{D}_2 \simeq D^*$. By construction of D^* it can be realized that $D^* \models Q_k$ for some conjunction Q_k in $\mathcal{Q}$. Because of this relation both Q_1 and Q_2 can not be contained in $\mathcal{Q}$ which is a contradiction. This means that there can not be a component c and conjunction $\bar{D}$ such that $Q_1 \simeq \bar{D} \wedge c \in A_1$ and $Q_2 \simeq \bar{D} \wedge c \in A_2$. □

Lemma 3 *Let $\mathcal{Q} = \mathcal{D}_{old} \wedge \mathcal{D}_{add}$ be the output from Algorithm 2 after processing all test negated conflicts in $\mathbb{P}$. If D_{i_m} is not contained in $\mathcal{D}_{old}$, and the set $D_{i_m} \wedge P_j$ is not contained in $\mathcal{D}_{add}$, after running the algorithm, then there is a $D_{i_{m+1}}$ such that $D_{i_m} \wedge P_j \models D_{i_{m+1}}$ and $D_{i_{m+1}} \wedge P_j \not\models D_{i_m} \wedge P_j$.*

PROOF. The fact that D_{i_m} is not contained in $\mathcal{D}_{old}$ means that the inner loop of the algorithm must have been entered when $D_i = D_{i_m}$. Then the fact that $D_{i_m} \wedge P_j$ is not contained in $\mathcal{D}_{add}$, means that $D_{i_m} \wedge P_j \models D_k$ for some D_k, $k \neq i_m$. By choosing $i_{m+1} = k$, this gives $D_{i_m} \wedge P_j \models D_{i_{m+1}}$.

Next we prove that $D_k \wedge P_j \not\models D_i \wedge P_j$. Let the single assignment in P_j be $a \in A_p$. We will divide the proof into four cases: (1) $a \notin \text{comps } D_i$, $a \notin \text{comps } D_k$, (2) $a \in \text{comps } D_i$, $a \notin \text{comps } D_k$, (3) $a \notin \text{comps } D_i$, $a \in \text{comps } D_k$, and (4) $a \in \text{comps } D_i$, $a \in \text{comps } D_k$.

1) The fact $D_i \wedge P_j \models D_k$ would imply $D_i \models D_k$ which is impossible because $\mathcal{D}$ is in MNF.

2) This means that D_i can be written as $D_i = D' \wedge a \in A_i$. The fact $D_i \wedge P_j \models D_k$ would then imply that $D' \models D_k$ and consequently that $D_i \models D_k$, which is impossible because $\mathcal{D}$ is in MNF.

3) First assume that D_i contains a component $c \notin D_k$. Note that this component is not component a. This would imply that c is not contained in P_j. Thus the components of $D_i \wedge P_j$ is a not a subset of the components of $D_k \wedge P_j$, which implies $D_k \wedge P_j \not\models D_i \wedge P_j$. The case left to investigate is when the components of D_i are a subset of the components of D_k.

   Assume that $D_k \wedge P_j \models D_i \wedge P_j$. This relation can be written $D'_k \wedge a \in A_p \cap A_k \models D_i \wedge a \in A_p$ where D'_k is a conjunction not containing component a. For this relation to hold it must hold that $D'_k \models D_i$. This means that $D_k = a \in A_k \wedge D'_k \models D_i$ which is impossible because $\mathcal{D}$ is in MNF.

4) Assume that $D_k \wedge P_j \models D_i \wedge P_j$. This relation can be written $D'_k \wedge a \in A_p \cap A_k \models D'_i \wedge a \in A_p \cap A_i$ where D'_k and D'_i are conjunctions not containing component a. This relation would imply $D'_k \models D'_i$. Further on, the fact $D_i \wedge P_j \models D_k$ can be written $a \in A_p \cap A_i \wedge D'_i \models a \in A_k \wedge D'_k$, which implies that $D'_i \models D'_k$. Thus we have $D'_i \simeq D'_k$ and the only possible difference between D_i and D_k is the assignment of component a. Lemma 2 says this is impossible.

With $i = i_m$ and $k = i_{m+1}$, these four cases have shown that $D_{i_{m+1}} \wedge P_j \not\models D_{i_m} \wedge P_j$. □

Lemma 4 *Let $\mathcal{D}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}_{n-1}$, and $\mathcal{Q}$ the output given $\mathcal{D}$ and $\mathcal{P}$ as inputs. For each conjunction D_i in $\mathcal{D}$ and P_j in $\mathcal{P}$ it holds that there is a conjunction Q_k in $\mathcal{Q}$ such that $D_i \wedge P_j \models Q_k$.*

PROOF. If, after running the algorithm, D_i is contained in $\mathcal{D}_{old}$, then the lemma is trivially fulfilled. If instead $D_i \wedge P_j$ is contained in $\mathcal{D}_{add}$, then the lemma is also trivially fulfilled. Study now the case where D_i is contained in $\mathcal{D}_{old}$ and $D_i \wedge P_j$ is not contained in $\mathcal{D}_{add}$. We can then apply Lemma 3 with $\mathbb{P} = \mathbb{P}_{n-1} \cup \{\mathcal{P}\}$. This gives us a $D_{i_{m+1}}$ such that $D_{i_m} \wedge P_j \models D_{i_{m+1}}$ and $D_{i_{m+1}} \wedge P_j \not\models D_{i_m} \wedge P_j$.

If $D_{i_{m+1}}$ is contained in $\mathcal{D}_{old}$, then the lemma is fulfilled. If instead $D_{i_{m+1}} \wedge P_j$ is contained in $\mathcal{D}_{add}$, note that $D_{i_m} \wedge P_j \models D_{i_{m+1}}$ implies $D_{i_m} \wedge P_j \models D_{i_{m+1}} \wedge P_j$. This means that the lemma is fulfilled. In this way we can repeatedly apply Lemma 3 as long as the new $D_{i_{m+1}}$ obtained is not contained in $\mathcal{D}_{old}$ and $D_{i_{m+1}} \wedge P_j$ not contained in $\mathcal{D}_{add}$.

We will now prove that after a finite number of applications of Lemma 3 we obtain a $D_{i_{m+1}}$ where $D_{i_{m+1}}$ is contained in $\mathcal{D}_{old}$ or $D_{i_{m+1}} \wedge P_j$ is contained in $\mathcal{D}_{add}$. Note that that each application of Lemma 3 guarantees that $D_{i_m} \wedge P_j \models D_{i_{m+1}} \wedge P_j$ and $D_{i_{m+1}} \wedge P_j \not\simeq D_{i_m} \wedge P_j$. This fact itself implies that there cannot be an infinite number of applications of Lemma 3. □

Theorem 1 *Let $\mathbb{P}$ be a set of negated conflicts that is not inconsistent, i.e. $\mathbb{P} \not\models \bot$, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Then it holds that $\mathcal{Q} \simeq \mathbb{P}$.*

PROOF. Let $\mathbb{P}_{n-1}$ denote the set all negated conflicts in $\mathbb{P}$ except $\mathcal{P}$. Then it holds that $\mathbb{P} \simeq \mathbb{P}_{n-1} \cup \{\mathcal{P}\} \simeq \mathcal{D} \wedge \mathcal{P}$. Lemma 4 implies that $\mathcal{D} \wedge \mathcal{P} \models \mathcal{Q}$. Left to prove is $\mathcal{Q} \models \mathcal{D} \wedge \mathcal{P}$. Take arbitrary conjunction Q_k in the output $\mathcal{Q}$. If Q_k is in $\mathcal{D}_{old}$, then it must be in also $\mathcal{D}$, i.e. $Q_k = D_i$ for some conjunction D_i in $\mathcal{D}$. The fact that D_i is in $\mathcal{D}_{old}$ means also that $D_i \models \mathcal{P}$. Thus $Q_k = D_i \models \mathcal{D} \wedge \mathcal{P}$. □

Lemma 5 *Let $\mathbb{P}_{n-1} \cup \mathcal{P}_n$ be a set of negated conflicts, and let each component have only two possible behavioral modes. If $\mathcal{D}$ is the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}_{n-1}$, then a new call to the algorithm with inputs $\mathcal{D}$ and $\mathcal{P}_n$ gives an output $\mathcal{Q}$ in which each conjunction is a kernel diagnosis.*

PROOF. Take an arbitrary conjunction Q_k in $\mathcal{Q}$. It holds that $Q_k \simeq D_i \wedge P_j$ for some conjunction D_i in $\mathcal{D}$ and some conjunction P_j in $\mathcal{P}_n$. If $Q_k \simeq D_i$, then Q_k is a kernel diagnosis since D_i is. Next we investigate the other case $Q_k \not\simeq D_i$.

Assume that Q_k is not a kernel diagnosis. The assignment P_j can be written as $c_p = M_p$. Thus, we can write Q_k as $Q_k = D_i \wedge (c_p = M_p)$. Since by assumption Q_k is not a kernel diagnosis, we can remove one assignment, either $c_p = M_p$ or some assignment $a = M_a$ in D_i, from Q_k and obtain a partial diagnosis. The partial diagnosis obtained is either D_i or $\bar{D} \wedge c_p = M_p$, where $D_i = \bar{D} \wedge a = M_a$. Study first the case where D_i is the partial diagnosis. By definition, this means that $D_i \models \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\}$, which implies $D_i \models \mathcal{P}_n$. This means that D_i would not be removed from D_{old} and thus become one conjunction in $\mathcal{Q}$. Since $Q_k = D_i \wedge (c_p = M_p) \models D_i$, both Q_k and D_i cannot be conjunctions in $\mathcal{Q}$ because $\mathcal{Q}$ is in MNF according to Theorem 2. This contradiction shows that D_i can not be a partial diagnosis.

Next, study the case where $\bar{D} \wedge c_p = M_p$ is the partial diagnosis, and let $\bar{M}_a$ denote the complementary element to M_a. This means that both $\bar{D} \wedge c_p = M_p \wedge a = M_a$ and $\bar{D} \wedge c_p = M_p \wedge a = \bar{M}_a$ are partial diagnoses. This means, by definition, that $\bar{D} \wedge c_p = M_p \wedge a = \bar{M}_a \models \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\} \simeq \mathcal{Q}$. Since $Q_k = \bar{D} \wedge a = M_a \wedge c_p = M_p$, and $\mathcal{Q}$ is in MNF, there must be another Q_m such that $\bar{D} \wedge c_p = M_p \wedge a = \bar{M}_a \models Q_m$. According to Lemma 2, it can not hold that $Q_m = \bar{D} \wedge c_p = M_p \wedge a = \bar{M}_a$. Therefore we can remove one assignment from $\bar{D} \wedge c_p = M_p \wedge a = \bar{M}_a$ and still obtain a conjunction d such that $d \models Q_m$. Note then that it can not hold that $d = \bar{D} \wedge c_p = M_p$ since this would imply that $Q_k \models Q_m$.

Now we investigate the case $d = \bar{D} \wedge a = \bar{M}_a$. Let Ω denote the set of assignments contained in $\bar{D}$. The fact that $Q_k = \bar{D} \wedge a = M_a \wedge c_p = M_p$ means that each negated conflict $\mathcal{P} \in \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\}$ contains an assignment in $\Omega \cup \{a = M_a\} \cup \{c_p = M_p\}$.

Next, $\bar{D} \wedge a = \bar{M}_a \models Q_m$ means that Q_m contains a subset of the assignments contained in $\bar{D} \wedge a = \bar{M}_a$. This further means that each negated conflict $\mathcal{P} \in \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\}$ contains an assignment from $\Omega_m \cup \{a = \bar{M}_a\}$. This means that a $\mathcal{P}'$ that does not contain any assignment from Ω_m must contain the assignment $a = \bar{M}_a$. The consequence of this is that $\mathcal{P}'$ cannot contain the assignment $a = M_a$. Since it was concluded above that each $\mathcal{P}$ contains an assignment in $\Omega \cup \{a = M_a\} \cup \{c_p = M_p\}$, $\mathcal{P}'$ must then contain the assignment $c_p = M_p$. Thus each negated conflict $\mathcal{P} \in \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\}$ contains an assignment from $\Omega_m \cup \{c_p = M_p\}$.

We can now select one assignment from each $\mathcal{P} \in \mathbb{P}_{n-1} \cup \{\mathcal{P}_n\}$ but with the requirement that the selected assignment must be $c_p = M_p$ or contained in Ω. By forming a conjunction Φ of these assignments, it will hold that $\bar{D} \wedge c_p = M_p \models \Phi$. Therefore $Q_k = \bar{D} \wedge a = M_a \wedge c_p = M_p \models \Phi$. If Φ is not one of the conjunctions in $\mathcal{Q}$, there will be another Q_v such that $\Phi \models Q_v$. This means that $Q_k \models Q_v$ and Q_i cannot be contained in $\mathcal{Q}$, which is a contradiction. Thus we have shown that it cannot hold that $d = \bar{D} \wedge a = \bar{M}_a$, and therefore that $\bar{D} \wedge c_p = M_p$ cannot be a partial diagnosis. This further means that Q_k must be a kernel diagnosis. □

Theorem 3 *Let each component have only two possible behavioral modes, let $\mathbb{P}$ be a set of negated conflicts, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Then it holds that each conjunction of $\mathcal{Q}$ is a kernel diagnosis.*

PROOF. It is not difficult to realize that, after processing the first two negated conflicts in $\mathbb{P}$, each conjunction of the output $\mathcal{Q}$ is a kernel diagnoses. For each further negated conflict that is processed, each conjunction of the new output will be a kernel diagnosis according to Lemma 5. □