Linköping University Post Print

A Generalized Minimal Hitting-Set Algorithm to Handle Diagnosis With Behavioral Modes

Mattias Nyberg

N.B.: When citing this work, cite the original article.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Mattias Nyberg, A Generalized Minimal Hitting-Set Algorithm to Handle Diagnosis With Behavioral Modes, 2011, IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A-SYSTEMS AND HUMANS, (41), 1, 137-148.
<http://dx.doi.org/10.1109/TSMCA.2010.2048750>

Postprint available at: Linköping University Electronic Press
<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-63153>

A Generalized Minimal Hitting-Set Algorithm to Handle Diagnosis With Behavioral Modes

Mattias Nyberg

Abstract—To handle diagnosis with behavioral modes, a new generalized minimal hitting-set algorithm is presented. The key properties in comparison with that of the original minimal hitting-set algorithm given by de Kleer and Williams are that it can handle more than two modes per component and also nonpositive conflicts. The algorithm computes a logical formula that characterizes all diagnoses. Instead of minimal 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. One usage of the algorithm is fault isolation in the sense of fault detection and isolation (FDI). The algorithm is experimentally shown to provide significantly better performance compared to the fault isolation approach based on structured residuals, which is commonly used in FDI.

Index Terms—Fault detection and isolation (FDI), fault diagnosis, fault isolation.

I. INTRODUCTION

WITHIN THE field of fault diagnosis, it has often been assumed that each component has only two possible behavioral modes, e.g., see [1] and [2]. For this case, and given a set of conflict sets, it is well known that a minimal hitting set corresponds to a minimal diagnosis [1].¹ Algorithms for computing all minimal hitting sets have been presented in [1] and [2]. Improvements have later been given in, e.g., [3] and [4].

In [1] and [2], it is assumed that a conflict can only imply that some component is faulty. This is called a *positive conflict* [5]. If all conflicts are positive, it is also well known that the set of all minimal diagnoses characterizes all diagnoses [2]. The case of all conflicts being positive will occur if, for example, the faulty modes of the components have no fault models. However, if there are fault models, it is possible to have nonpositive conflicts.

If there is a desire to compute something that characterizes all diagnoses when there are nonpositive conflicts, the concept of minimal hitting sets and the algorithms in [1] and [2] cannot be used. To solve this, an alternative characterization based on

the so-called *kernel diagnoses* was proposed in [5], where also an algorithm to compute the kernel diagnoses was given. The kernel diagnoses characterize all diagnoses even in the case of nonpositive conflicts.

It has been noted in several papers that more than two possible behavioral modes are useful when designing diagnostic systems (see, e.g., [6] and [7]). For this case, neither minimal diagnoses nor kernel diagnoses can be used to characterize all diagnoses, and none of the algorithms in [1], [2], or [5] is applicable. However, Williams and Ragno [8] introduce *kernels* as a generalization of kernel diagnoses to more than two behavioral modes.

For the case of more than two behavioral modes and nonpositive conflicts, this paper proposes a new logical characterization of all diagnoses. Conflicts and diagnoses are represented by logical formulas, and instead of minimal diagnoses, kernel diagnoses, and kernels, we use more general conjunctions of 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.

The main contribution is a new *generalized hitting-set algorithm* computing the proposed logical characterization. The minimal hitting-set algorithm given in [2] is shown to be a special case of this new generalized algorithm. Note that even though the papers [6]–[8] consider more than two behavioral modes per component, they are not concerned with the characterization of and, in particular, the computation of a characterization 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 [1]). 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 introduced in [9]. In this paper, we will show how to obtain these preferred diagnoses by means of the aforementioned logical formulas and the new generalized minimal hitting-set algorithm.

The proposed generalized minimal hitting-set algorithm can be used in a traditional diagnosis problem formulation, as in [1] or [2], where a model and a set of observations are utilized to compute conflicts by the technique of “local propagation.” Another usage is in the case of precompiled potential conflicts [10]. This usage corresponds to the fault isolation problem, as defined within the control community (usually referred to as fault detection and isolation (FDI); e.g., see [11]–[18]). Precompiled potential conflicts are a common solution in

Manuscript received March 7, 2008; revised March 5, 2009; accepted December 6, 2010. Date of publication June 14, 2010; date of current version November 10, 2010. This paper was recommended by Associate Editor H. Pham.

The author is with the Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden, and also with Scania CV AB, 151 87 Södertälje, Sweden (e-mail: matny@isy.liu.se).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2010.2048750

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

embedded control systems where memory and computational limitations make it impossible to implement a full diagnostic inference engine that works directly on a model of the system. Section VIII contains an example of such an application: on-board diagnosis of the electrical driver for the fuel injection system of an automotive engine. The usage of the algorithm is demonstrated, as well as a short performance comparison with an alternative approach from the area of FDI. In the context of precompiled potential conflicts, and for the evaluation of real-world performance, the algorithm has also been tested in a fleet of real vehicles with promising results.

This paper is organized as follows. In Section II, the minimal hitting-set algorithm from [2] is restated as a reference. In Section III, the logical framework is presented. Then, the new generalized minimal hitting-set algorithm is given in Section IV. Sections V and VI discuss the relation to minimal and kernel diagnoses. Section VII describes how to compute the preferred diagnoses. Finally, Section VIII contains the aforementioned application study. All proofs of theorems have been placed in the Appendix.

II. GDE MINIMAL HITTING-SET ALGORITHM

Before presenting the new generalized minimal hitting-set algorithm, this section presents the general diagnostic engine (GDE) minimal hitting-set algorithm and its associated framework, as presented in [2]. 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. This means that 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 $\delta \subseteq \mathcal{C}$. Components contained in a diagnosis δ are assumed faulty, and components not contained in δ are assumed 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, then δ is a diagnosis if and only if, for all conflicts $C \in \mathbb{C}$, it holds that $\delta \cap C \neq \emptyset$. That is, δ is diagnosis if it is a so-called *hitting set* with respect to the collection of sets $\mathbb{C}$.

Given a set of diagnoses Δ and a new conflict C , the minimal hitting-set algorithm in [2] finds an updated set of minimal diagnoses. A version of the algorithm, as described in [2], is presented here as Algorithm 1.

Algorithm 1

input: a set of minimal diagnoses Δ , and a new conflict set C
output: the updated set of minimal diagnoses Θ

1. $\Delta_{\text{old}} := \Delta$
2. $\Delta_{\text{add}} := \emptyset$
3. **forall** $\delta_i \in \Delta$ **do**
4. **if** $\delta_i \cap C = \emptyset$ **then**

5. Remove δ_i from Δ_{old}
6. **forall** $c \in C$ **do**
7. $\delta_{\text{new}} := \delta_i \cup \{c\}$
8. **forall** $\delta_k \in \Delta$, $\delta_k \neq \delta_i$ **do**
9. **if** $\delta_k \subseteq \delta_{\text{new}}$ **then goto** LABEL1
10. **end**
11. $\Delta_{\text{add}} := \Delta_{\text{add}} \cup \{\delta_{\text{new}}\}$
12. LABEL1
13. **end**
14. **end**
15. **end**
16. $\Theta := \Delta_{\text{old}} \cup \Delta_{\text{add}}$

The algorithm has the property that if Δ is the set of all minimal diagnoses, the algorithm output Θ will contain all minimal diagnoses with respect also to the new conflict C . Furthermore, 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 [3]. It can also be noted that the loop over $\delta_k \in \Delta$ could be modified to $\delta_k \in \Delta_{\text{old}}$, which would be more efficient since Δ_{old} is smaller than Δ .

III. 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 (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 in $\mathcal{C}$, 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 $M \subseteq \mathbf{R}_c$, the expression $c \in M$ is a formula. For example, consider a sensor that we model as the component s_1 . The formula $s_1 \in \{NF, G, UF\}$ means that the sensor is in mode NF , G , or UF . If M is a singleton, e.g., $M = \{NF\}$, we will also sometimes write $c = NF$. Furthermore, the constant $\perp$ 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 the set of assignments of the variables $\mathcal{C}$ that make γ true is a subset of the assignments that 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, then Φ and Γ are said to be equivalent, and we write $\Gamma \simeq \Phi$.

We will devote special interest to conjunctions on the form

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

where all components are unique, i.e., $c_i \neq c_j$ if $j \neq i$, 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

$$D_1 \vee D_2 \vee \dots \vee D_m. \quad (2)$$

Note that the different conjunctions D_i can contain different numbers 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 components s_1 , s_2 , and s_3 , where all have the behavioral-mode domain $\mathbf{R}_{s_i} = \{NF, G, B, UF\}$:

$$\begin{aligned} s_1 \in \{UF\} \wedge s_2 \in \{B, UF\} \wedge s_3 \in \{UF\} \\ s_1 \in \{UF\} \wedge s_2 \in \{B, UF\} \wedge s_1 \in \{G, UF\}. \end{aligned}$$

The first formula is in MNF but not the second since $s_1 \in \{UF\} \wedge s_2 \in \{B, UF\} \models s_1 \in \{G, UF\}$. The interpretation of the first formula is that sensor s_1 is in mode UF and sensor s_2 is in one of the modes B or UF , or sensor s_3 is in mode UF .

A. Conflicts and Diagnoses

A conflict is assumed to be written using the logical language defined earlier. For example, if it has been found that the component s_1 cannot be in mode NF at the same time as s_2 is in mode B or NF , then this gives the conflict

$$s_1 \in \{NF\} \wedge s_2 \in \{B, NF\}. \quad (3)$$

Note that, in a real system, the behavior of a sensor in mode NF cannot be distinguished from a very small bias that is a behavior belonging to mode B . Thus, $s_1 \in \{NF\} \wedge s_2 \in \{B\}$ can never be a conflict.

To relate this definition of conflict to the one used in Section II, consider the conflict $C = \{s_1, s_2, s_3\}$. With the logical language, we can write this conflict as $s_1 \in \{NF\} \wedge s_2 \in \{NF\} \wedge s_3 \in \{NF\}$.

Instead of conflicts, we will mostly use negated conflicts. In particular, we will use negated conflicts written in MNF. For an example, if the conflict (3) is negated and written in MNF, we obtain

$$s_1 \in \{G, B, UF\} \vee s_2 \in \{G, UF\}. \quad (4)$$

Without loss of generality, we will assume from now on that all negated conflicts are written on the form

$$c_1 \in M_1 \vee c_2 \in M_2 \vee \dots \vee c_n \in M_n \quad (5)$$

where $c_j \neq c_k$ if $j \neq k$, and $\emptyset \neq M_i \subset \mathbf{R}_{c_i}$. This means that (5) is in MNF.

A *system behavioral mode* is a conjunction containing a unique assignment of all components in $\mathcal{C}$. For example, if $\mathcal{C} = \{s_1, s_2, s_3\}$, a system behavioral mode could be

$$s_1 = UF \wedge s_2 = B \wedge s_3 = NF.$$

We consider the term *diagnosis* to refer to a system behavioral mode consistent with all negated conflicts.

Definition 1: Let $\mathbb{P}$ be the set of all negated conflicts. A system behavioral mode d is a *diagnosis* if $\{d\} \cup \mathbb{P} \not\models \perp$ or, equivalently, $d \models \mathbb{P}$.

To relate this definition of diagnosis to the one used in Section II, assume that $\mathcal{C} = \{s_1, s_2, s_3, s_4\}$ and consider the diagnosis $\delta = \{s_1, s_2\}$. With the logical language, we can write this diagnosis as $s_1 = F \wedge s_2 = F \wedge s_3 = NF \wedge s_4 = NF$.

B. Example

To illustrate how the logical language can be used to reason and perform diagnostic inference, consider the following example. Assume again that $\mathcal{C} = \{s_1, s_2, s_3\}$, where all have the behavioral-mode domain $\mathbf{R}_{s_i} = \{NF, G, B, UF\}$. Assume also that two conflicts have been detected

$$\begin{aligned} s_1 \in \{NF\} \wedge s_2 \in \{NF\} \\ s_2 \in \{NF, B\}. \end{aligned}$$

This corresponds to the negated conflicts

$$\begin{aligned} s_1 \in \{G, B, UF\} \vee s_2 \in \{G, B, UF\} \\ s_2 \in \{G, UF\}. \end{aligned}$$

To identify the set of diagnoses, we take the conjunction of the two negated conflicts and translate it to MNF. That is

$$\begin{aligned} (s_1 \in \{G, B, UF\} \vee s_2 \in \{G, B, UF\}) \wedge s_2 \in \{G, UF\} \\ \simeq s_1 \in \{G, B, UF\} \wedge s_2 \in \{G, UF\} \vee s_2 \in \{G, UF\} \\ \simeq s_2 \in \{G, UF\}. \end{aligned}$$

In the last equivalence, the first conjunction is removed since the second is a consequence of the first, i.e., $s_1 \in \{G, B, UF\} \wedge s_2 \in \{G, UF\} \models s_2 \in \{G, UF\}$. This removal results in that the last formula is in MNF. From the last formula, it is easy to read out that the diagnoses are all system behavioral modes such that $s_2 = G$ or $s_2 = UF$, e.g., $s_1 = NF \wedge s_2 = G \wedge s_3 = NF$ and $s_1 = G \wedge s_2 = UF \wedge s_3 = NF$.

In this small example, there were two conflicts, and we could easily, by hand, derive a formula in MNF equivalent to the conjunction of all negated conflicts. The algorithm presented in the next section derives this MNF formula in the general case.

IV. GENERALIZED MINIMAL HITTING-SET ALGORITHM

This section presents the new generalized minimal hitting-set algorithm. It handles more than two behavioral modes per component and also nonpositive conflicts. The algorithm takes 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 $\mathcal{Q} \simeq \mathcal{D} \wedge \mathcal{P}$.

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: $\mathcal{Q}$

1. $\mathcal{D}_{\text{old}} := \mathcal{D}$
 2. $\mathcal{D}_{\text{add}} := \text{empty formula}$
 3. **forall** $D_i \in \mathcal{D}$ **do**
 4. **if** $D_i \not\models \mathcal{P}$ **then**
 5. Remove D_i from $\mathcal{D}_{\text{old}}$
 6. **forall** $P_j \in \mathcal{P}$ **do**
 7. Let D_{new} be a conjunction in MNF such
 that $D_{\text{new}} \simeq D_i \wedge P_j$
 8. **forall** $D_k \in \mathcal{D}, D_k \neq D_i$ **do**
 9. **if** $D_{\text{new}} \models D_k$ **then goto** LABEL1
 10. **end**
 11. $\mathcal{D}_{\text{add}} := \mathcal{D}_{\text{add}} \vee D_{\text{new}}$
 12. LABEL1
 13. **end**
 14. **end**
 15. **end**
 16. $\mathcal{Q} := \mathcal{D}_{\text{old}} \vee \mathcal{D}_{\text{add}}$
-

To keep the algorithm description “clean,” some operations have been written in a simplified form. More details are discussed in Section IV-C in the following. 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 negated conflict $\mathcal{P}_1$ is considered, we already have a formula in MNF, and 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, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Then, the following hold.

- 1) $\mathcal{Q} \simeq \mathbb{P}$.
- 2) $\mathcal{Q}$ is in MNF.

□

The proof for this theorem can be found in the Appendix.

Remark: The importance of Theorem 1 is, according to item 1) and Definition 1, that the formula $\mathcal{Q}$ represents all diagnoses in the sense that d is a diagnosis if and only if it holds that $d \models \mathcal{Q}$, and according to item 2), that $\mathcal{Q}$ has the nice property of compactness, as explained in Section III.

A. Relation to the GDE Minimal Hitting-Set Algorithm

The original GDE minimal hitting-set algorithm stated in Section II represents conflicts and diagnoses as sets of compo-

nents. The new generalized minimal hitting-set algorithm can, in fact, be obtained by modifying this original algorithm. The principal difference is that all set operations are replaced with operations on MNF formulas.

The modifications are the following.

- 1) 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 a disjunction of conjunctions on the form (1).
- 2) Instead of using a conflict set C as input, use a negated conflict $\mathcal{P}$ on the form (5).
- 3) Instead of checking the condition $\delta_i \cap C = \emptyset$, check the condition $D_i \not\models \mathcal{P}$.
- 4) Instead of the assignment $\delta_{\text{new}} := \delta_i \cup \{c\}$, find a conjunction D_{new} in MNF such that $D_{\text{new}} \simeq D_i \wedge P_j$.
- 5) Instead of checking the condition $\delta_k \subseteq \delta_{\text{new}}$, check the condition $D_{\text{new}} \models D_k$.

B. Example

To illustrate the generalized minimal hitting-set algorithm, consider again an example where $\mathcal{C} = \{s_1, s_2, s_3\}$ and the domain of behavioral modes for each component is $\mathbb{R}_{s_i} = \{NF, G, B, UF\}$. We use the algorithm with the following inputs:

$$\mathcal{D} = D_1 \vee D_2 = s_1 \in \{G, B, UF\} \vee s_3 \in \{G, UF\}$$

$$\mathcal{P} = P_1 \vee P_2 = s_2 \in \{B, UF\} \vee s_3 \in \{G, B, UF\}.$$

In the execution of the algorithm, we enter line 4 where the condition $D_1 \not\models \mathcal{P}$ is fulfilled, which means that D_1 is removed from $\mathcal{D}_{\text{old}}$ and the second loop of the algorithm is entered. There, in line 7, a D_{new} is created such that $D_{\text{new}} \simeq D_1 \wedge P_1 = s_1 \in \{G, B, UF\} \wedge s_2 \in \{B, UF\}$. This D_{new} is then, in line 9, compared to D_2 in the condition $D_{\text{new}} \models D_2$. The condition is not fulfilled, which means that D_{new} is added to $\mathcal{D}_{\text{add}}$ in line 11. In the next iteration of the second loop, a D_{new} is created such that $D_{\text{new}} \simeq D_1 \wedge P_2 = s_1 \in \{G, B, UF\} \wedge s_3 \in \{G, B, UF\}$. Also, this time, the condition $D_{\text{new}} \models D_2$ is not fulfilled, implying that D_{new} is added to $\mathcal{D}_{\text{add}}$. Next, the conjunction D_2 is investigated, but since the condition $D_2 \models \mathcal{P}$ in line 4 holds, D_2 is not removed from $\mathcal{D}_{\text{old}}$, and the second loop is not entered. The algorithm output is finally formed as

$$\mathcal{Q} := \mathcal{D}_{\text{old}} \vee \mathcal{D}_{\text{add}} = D_2 \vee (D_1 \wedge P_1 \vee D_1 \wedge P_2)$$

$$= s_3 \in \{G, UF\} \vee s_1 \in \{G, B, UF\} \wedge s_2 \in \{B, UF\}$$

$$\vee s_1 \in \{G, B, UF\} \wedge s_3 \in \{G, B, UF\}.$$

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

C. Algorithm Details

To implement the algorithm, some more details need to be considered. The first is how to check the condition $D_i \not\models \mathcal{P}$ in line 4. To illustrate this, consider an example where D_i contains components c_1, c_2 , and c_3 and where $\mathcal{P}$ has components c_2, c_3 ,

and c_4 . Since $\mathcal{D}$ is in MNF, and $\mathcal{P}$ in the form (5), D_i and $\mathcal{P}$ will have the form

$$D_i = c_1 \in M_1^D \wedge c_2 \in M_2^D \wedge c_3 \in M_3^D \quad (6)$$

$$\mathcal{P} = c_2 \in M_2^P \vee c_3 \in M_3^P \vee c_4 \in M_4^P. \quad (7)$$

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, in line 7, find an expression Q_{new} in MNF such that $Q_{\text{new}} \simeq D_i \wedge P_j$. To illustrate this, consider an example where D_i contains components c_1 and c_2 and where P_j has component c_2 . Since $\mathcal{D}$ is in MNF, and $\mathcal{P}$ in the form (5), D_i and P_j will have the form

$$D_i = c_1 \in M_1^D \wedge c_2 \in M_2^D \quad (8a)$$

$$P_j = c_2 \in M_2^P. \quad (8b)$$

Then, Q_{new} will be formed as

$$D_{\text{new}} = c_1 \in M_1^D \wedge c_2 \in M_2^D \cap M_2^P$$

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

The third detail is how to check the condition $D_{\text{new}} \models D_k$ in line 9. To illustrate this, consider an example where D_{new} contains components c_1 and c_2 and where D_k has 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

$$D_{\text{new}} = c_1 \in M_1^n \wedge c_2 \in M_2^n \quad (9a)$$

$$D_k = c_2 \in M_2^D \wedge c_3 \in M_3^D. \quad (9b)$$

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

$$D'_{\text{new}} = c_1 \in M_1^n \wedge c_2 \in M_2^n \wedge c_3 \in \mathbf{R}_{c_3} \quad (10)$$

$$D'_k = c_1 \in \mathbf{R}_{c_1} \wedge c_2 \in M_2^D \wedge c_3 \in M_3^D. \quad (11)$$

Now, we see that the condition $D_{\text{new}} \models D_k$ 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 $D_{\text{new}} \models D_k$ 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 $M_i^n \subseteq M_i^D$.

D. Complexity

The complexity of Algorithm 2 mimics that of the original Algorithm 1. If $|\mathcal{D}|$ and $|\mathcal{P}|$ denote the number of conjunctions in $\mathcal{D}$ and $\mathcal{P}$, respectively, the worst case complexity of Algorithm 2 is on the order of $|\mathcal{D}|^2|\mathcal{P}|$. When the algorithm is used in an iterative fashion to process a set of n negated

conflicts, the total worst case complexity becomes $|\mathcal{P}|^{2n+1}$, i.e., exponential. In spite of this worst case performance, the algorithm can perform well in a real-world setting, as will be described in Section VIII.

V. RELATION TO MINIMAL DIAGNOSES

The concept of minimal diagnoses was originally proposed in [1] and [2] for systems where each component has only two possible behavioral modes, i.e., a normal fault-free mode and a faulty mode. Minimal diagnoses have two attractive properties. First, they represent the “simplest” diagnoses, in the sense that all other diagnoses contain additional faulty components, and are therefore often desired when prioritizing among diagnoses according to the principle of parsimony. Second, 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 nonpositive conflicts.

A. “Simplest” Property

For the case of more than two modes per component, the concept of *preferred diagnoses* was defined in [9] 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 others. For example, *NF* is usually preferred over any other mode, and a simple electrical fault, such as short or open circuit, may be preferred over other more complex behavioral modes. Furthermore, 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. Furthermore, these relations induce a partial order on the system behavioral modes. Let d_1 and d_2 be two system behavioral modes, i.e., $d_i = \bigwedge_{c \in \mathcal{C}} (c = b_c^i)$. 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 be formally defined as a diagnosis d_i such that there is no other diagnosis d_j where $d_j > d_i$. In Section VII, 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, the preferred diagnoses are exactly the minimal diagnoses.

A different approach, compared to the concept of preferred diagnoses, is to compute the most probable diagnoses as in [7] and [8]. For example, in [8], the diagnosis problem is formulated as a constraint satisfaction problem, and the most probable diagnoses are computed using A* search. When using most probable diagnoses as in [7] and [8], it is required that a probability is assigned to each behavioral mode. Note the contrast to the concept of preferred diagnoses, which only requires a preference relation in the form of a partial order. This is an advantage in applications where it is difficult to obtain probability values of each behavioral mode.

Remark: One may ask if what “preferred” or “simplest” diagnoses mean. One possible formal justification is the

following. If $\mathcal{Q}$ is a formula such that $\mathcal{Q} \simeq \mathbb{P}$, then it holds that $P(d_i|\mathbb{P}) = P(d_i \wedge \mathcal{Q})/P(\mathcal{Q})$. This means that $P(d_i|\mathbb{P}) = P(d_i)/P(\mathcal{Q})$ if $d_i \models \mathbb{P}$, i.e., if d_i is a diagnosis, and $P(d_i|\mathbb{P}) = 0$ if $d_i \not\models \mathbb{P}$, i.e., if d_i 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 $P(d_i|\mathbb{P})$ for different diagnoses, it is enough to consider the priors $P(d_i)$. We assume that faults occur independently of each other, which means that $P(d_i) = \prod_{c \in \mathcal{C}} P(c = b_c^i)$, where $P(c = b_c^i)$ is the prior probability that component c is in behavioral mode b_c^i . To know the exact value of a prior $P(c = b_c^i)$ 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 the ones with highest probability. It can be noted that, in contrast, the concept of most probable diagnoses (see [7] and [8]) requires exact values of the priors $P(c = b_c^i)$, something that can be hard to obtain in real applications.

B. 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 nonpositive conflicts. In some special cases, the preferred diagnoses characterize all diagnoses with the help of the partial order $\geq$, but 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 $\mathcal{C} = \{s_1, s_2, s_3, s_4\}$, $\mathbf{R}_{s_i} = \{NF, B, G, UF\}$ for all components, and $s_1 \in \{B, UF\} \wedge s_2 \in \{G, UF\}$ is one of the conjunctions in an MNF formula. By letting each diagnosis be represented as an ordered set corresponding to $\langle s_1, s_2, s_3, s_4 \rangle$, this single conjunction characterizes the diagnoses

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

which is 256 diagnoses.

For another example, assume that each of the components $\mathcal{C} = \{s_1, s_2, s_3, s_4\}$ has only two modes, i.e., $\mathbf{R}_{s_i} = \{NF, F\}$. A conjunction $s_1 \in \{F\} \wedge s_2 \in \{F\}$ would then characterize all diagnoses $\{F\} \times \{F\} \times \{NF, F\} \times \{NF, F\}$. In Section II, this conjunction would be represented by $\{s_1, s_2\}$. 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 II.

VI. RELATION TO KERNEL DIAGNOSES

The paper [5] defines *partial diagnosis* and *kernel diagnosis*. In this section, we will see that the output of Algorithm 2 can be seen as a set of kernel diagnoses. In [5], the concept kernel diagnoses were introduced in the context of 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 nonpositive conflicts. As noted in [5], 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 unique mode assignments such that $d \models \mathbb{P}$. Then, a kernel diagnosis is a 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 2: 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. $\square$

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:

$$s_1 = N \wedge s_2 = N \vee s_1 = N \wedge s_2 = F. \quad (12)$$

All diagnoses represented by (12) are characterized by the single kernel diagnosis $s_1 = N$. Therefore, none of the conjunctions in (12) is kernel diagnosis.

A previous algorithm for calculating kernel diagnoses is given in [5]. In the language of this paper, this previous algorithm first makes a full expansion of the conjunction of all negated conflicts by distributing $\wedge$ over $\vee$. Then, all conjunctions that are not kernel diagnoses are removed.

VII. EXTRACTING PREFERRED DIAGNOSES

In Section V, 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, then it will have only one preferred diagnosis. This preferred diagnosis is 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 an example, consider two components s_1 and s_2 where $\mathbf{R}_{s_i} = \{NF, E, F\}$ and $NF >_{s_i} E >_{s_i} F$, and a third component s_3 where $\mathbf{R}_{s_3} = \{NF, B, G\}$ with the only relations $NF >_{s_3} B$ and $NF >_{s_3} G$. Then, consider the MNF formula

$$\mathcal{Q} = s_1 \in \{E\} \wedge s_3 \in \{B, G\} \\ \vee s_1 \in \{E, F\} \wedge s_2 \in \{E, F\} \wedge s_3 \in \{B, G\}. \quad (13)$$

The preferred diagnoses consistent with the first conjunction are $s_1 = E \wedge s_2 = NF \wedge s_3 = B$ and $s_1 = E \wedge s_2 = NF \wedge s_3 = G$. The preferred diagnoses consistent with the second conjunction are $s_1 = E \wedge s_2 = E \wedge s_3 = B$ and $s_1 = E \wedge s_2 = E \wedge s_3 = G$. As seen, the two diagnoses $s_1 = E \wedge s_2 = E \wedge s_3 = B$ and $s_1 = E \wedge s_2 = E \wedge s_3 = G$ are not preferred diagnoses of the whole formula $\mathcal{Q}$.

Fig. 1. Plot shows the total time needed to compute the preferred diagnoses on the Y-axis, and the time needed to compute $\mathcal{Q}$ on the X-axis (the straight line is included as reference). The histogram shows the distribution of additional computation time to compute the preferred diagnoses relative to the time needed to compute $\mathcal{Q}$.

The example shows that the preferred diagnoses cannot 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 the nonpreferred diagnoses. These can be removed by a simple pairwise comparison. Note that the set Ψ need not be calculated for every new negated conflict that is processed, i.e., instead, only at the time the preferred diagnoses are really needed (for example, before a service task is to be carried out).

One may ask how much extra time is needed for the computation of the preferred diagnoses as 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 total of 132 test cases were randomly generated. The test cases represent systems with between four and seven components, where each component has four possible behavioral modes. The number of negated conflicts varies between 2 and 12.

In Fig. 1, the results for the 132 test cases are shown. Each X-mark in the upper plot represents one test run. Moreover, the total time needed to compute the preferred diagnoses is on the Y-axis, and the time needed to compute $\mathcal{Q}$ is on the X-axis. The histogram shows the distribution of additional computation time needed to compute the preferred diagnoses from $\mathcal{Q}$, relative to the time needed to compute $\mathcal{Q}$. As seen, the extra time is mostly small compared to the total time needed to compute the preferred diagnoses.

VIII. APPLICATION EXAMPLE

We will now illustrate how the new generalized minimal hitting-set algorithm can be used in a practical diagnosis application. One area where model-based diagnosis is important is automotive applications, e.g., see [19]. Therefore, we choose, as an application example, an electrical driver for the fuel injectors of a six-cylinder automotive engine. This system has six components, namely, one driver for each of the six injectors. Each driver has eight behavioral modes: *NF*, *SBB* (short between banks), *SC* (stuck closed), *SCG* (short circuit to ground), *SLB* (short circuit on the low side to the ground), *OL* (open load), *SHB* (short circuit on the high side to the battery), and *UF*. The complexity of this example is illustrated by the fact that, in total, there are $8^6 = 262\,144$ system behavioral modes.

For onboard diagnosis of the system, there are 52 diagnostic tests corresponding to precompiled potential conflicts [10]. These are implemented in both hardware and software of the embedded system. Each diagnostic test tests the functionality of a subset of the system. The outcome of each diagnostic test is either pass or fail. If the outcome is fail, a negated conflict is created. The response of the diagnostic tests with respect to the different single faults is shown in the table in Fig. 2. An X in row i and column j means that the i th diagnostic test may respond to the fault of column j .

For example, we can see that the diagnostic test T7 may respond to behavioral mode *SCG* or *UF* in any of injectors 2, 3, 4, or 5. If the outcome of the test T7 is fail, we obtain the negated conflict $inj_2 \in \{SCG, UF\} \vee inj_3 \in \{SCG, UF\} \vee inj_4 \in \{SCG, UF\} \vee inj_5 \in \{SCG, UF\}$.

We now assume that tests 10, 30, 38, 44, and 45 have the outcome fail. Then, the set of all preferred diagnoses is to be computed with Algorithm 2, together with the principles described in Section VII. For comparison, we also use a commonly used FDI approach to fault isolation, namely, *structured residuals* [11]. In this approach, the actual response of the diagnostic tests is matched to the expected responses of the diagnostic tests for different faults, the so-called *fault signatures*. In the experiment, we have used the table of fault signatures, as shown in Fig. 2, but extended to all multiple faults. Since the X:s in the table corresponds to the case of an uncertain response, we say that a fault (i.e., a system behavioral mode) matches the actual response if each 0 corresponds to a diagnostic test with outcome pass, and each X to a test with outcome pass or fail. To make the comparison between the structured residuals and approach based on Algorithm 2 fair, we extend the structured-residual approach so that it computes the preferred diagnoses, which is also a more relevant problem. This is done by traversing the table from left to right, and the system behavioral mode b of each column is compared to a set Ω of already computed preferred diagnoses. If it is concluded that $b < d$ for some diagnosis $d \in \Omega$, then b is neglected; otherwise, it is added to Ω if the diagnostic test response matches the column. Furthermore, if it is concluded that $d < b$, then d is removed from Ω .

When calculating the preferred diagnoses, we use a partial order defined by the relations $NF > b$ for all behavioral modes $b \neq NF$ and $b > UF$ for all $b \neq UF$. The total number of diagnoses is computed to be 31 960. Furthermore,

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 modes per component, the minimal diagnoses do not necessarily correspond to the most desired diagnoses, it was instead shown how the preferred diagnoses could be obtained from the conjunctions with a reasonable amount of computational effort.

Finally, one possible application for the proposed algorithm was demonstrated, namely, onboard fault isolation in automotive embedded systems. In this application study, it was seen that the proposed algorithm provides a significant performance improvement compared to an approach based on structured residuals, which is the standard fault isolation method within FDI. Furthermore, in a real-world test involving a fleet of vehicles, the new algorithm has been shown to perform well.

APPENDIX PROOFS OF THE THEOREMS

The Appendix contains proofs for the two theorems presented in this paper. In the proofs, we will assume that the set of negated conflicts $\mathbb{P}$ is ordered. We will then use the notation $\mathbb{P}_n$ to denote the subset of the n th first elements in $\mathbb{P}_n$. For a given n , the notation $\mathcal{Q}^*$, or $\mathcal{D}^*$, will be used to denote the full expansion of $\bigwedge_{P \in \mathbb{P}_n} P$ obtained by distributing $\wedge$ over $\vee$. For example, if $\mathbb{P}_2 = \{a \in \{A, B\} \vee b \in \{A\}, a \in \{B, C\} \vee c \in \{B\}\}$, then the full expansion of $\bigwedge_{P \in \mathbb{P}_2} P$ will be

$$\mathcal{Q}^* = a \in \{B\} \vee a \in \{A, B\} \wedge c \in \{B\} \vee a \in \{B, C\} \wedge b \in \{A\} \vee b \in \{A\} \wedge c \in \{B\}. \quad (14)$$

Furthermore, the notation $\mathcal{Q}_{\min}^*$ is used to denote an expression obtained by removing, from $\mathcal{Q}^*$, one by one, each conjunction Q_i^* as long as there is still another conjunction Q_j^* left in $\mathcal{Q}^*$ such that $Q_i^* \models Q_j^*$.

Proof of Theorem 1

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 $Q_2 \models Q_1$. Note first that $Q_1 \in \mathcal{D}_{\text{old}}$ and $Q_2 \in \mathcal{D}_{\text{old}}$ cannot hold since lines 1 and 5 imply that $\mathcal{D}_{\text{old}} \subseteq \mathcal{D}$ and $\mathcal{D}$ is in the input required to be in MNF. There are therefore three cases that need to be investigated: 1) $Q_1 \in \mathcal{D}_{\text{old}}$, $Q_2 \in \mathcal{D}_{\text{add}}$; 2) $Q_2 \in \mathcal{D}_{\text{old}}$, $Q_1 \in \mathcal{D}_{\text{add}}$; and 3) $Q_1 \in \mathcal{D}_{\text{add}}$, $Q_2 \in \mathcal{D}_{\text{add}}$.

- 1) Since $Q_1 \in \mathcal{D}_{\text{old}}$, it holds, from line 1, that $Q_1 \in \mathcal{D}$. Note that $\mathcal{D}_{\text{add}}$ is assigned in line 11, and the fact $Q_2 \in \mathcal{D}_{\text{add}}$ then means that $D_{\text{new}} = Q_2$ in some iteration of the second loop. During this iteration, it could not be the case that $D_i = Q_1$ since Q_1 would have then been removed from $\mathcal{D}_{\text{old}}$ in line 5. Therefore, D_{new} must have been compared to Q_1 in line 9. Since Q_2 has really been added, and line 11 executed, it cannot have been the case that $Q_2 \models Q_1$.
- 2) Since $Q_1 \in \mathcal{D}_{\text{add}}$, it holds from line 7 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) From the way D_{new} is formed in line 7, there are three cases: a) $Q_2 = D_i \wedge P_{j2}$, $Q_1 = D_i \wedge P_{j1}$; b) $Q_2 = D_{i2} \wedge P_j$, $Q_1 = D_{i1} \wedge P_j$; and c) $Q_2 = D_{i2} \wedge P_{j2}$, $Q_1 = D_{i1} \wedge P_{j1}$, where, in all cases, $P_{j1} \neq P_{j2}$ and $D_{i1} \neq D_{i2}$.
 - a) Let us say that $P_{j1} = a \in A_p$. Note that, according to (5), $A_p \subset \mathbf{R}_a$. For the relation $Q_2 = D_i \wedge P_{j2} \models D_i \wedge P_{j1} = Q_1$ to hold, it must therefore be the case that the component of P_{j1} is contained in D_i or P_{j2} . The latter is not possible because of the assumed form (5) of $\mathcal{P}$. Hence, let us say that $D_i = a \in A \wedge \dots$. The relation $Q_2 \models Q_1$ implies that $A \subseteq A \cap A_p$, which further means that $A \subseteq A_p$. This implies that $D_i \models a \in A_p \models \mathcal{P}$. Thus, Q_1 and Q_2 are, because of the condition in line 4, never subject to be added to $\mathcal{D}_{\text{add}}$, which is a contradiction.
 - b) Since $Q_2 \in \mathcal{D}_{\text{add}}$, $D_{\text{new}} = Q_2$ in some iteration of the second loop. In this iteration, D_i in the algorithm equals D_{i2} . Thus, D_k in the third loop can take the value D_{i1} . We have that $D_{\text{new}} = Q_2 = D_{i2} \wedge P_j \models D_{i1} \wedge P_j \models D_{i1}$. This means, according to the condition in line 9, that Q_2 cannot have been added to $\mathcal{D}_{\text{add}}$, which is a contradiction.
 - c) We have that $Q_2 = D_{i2} \wedge P_{j2} \models D_{i1} \wedge P_{j1} \models D_{i1} \in \mathcal{D}$. By reasoning as in case b), this means that Q_2 cannot have been added to $\mathcal{D}_{\text{add}}$.

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

Lemma 2: Let $\mathcal{D}^*$ be the full expansion of $\bigwedge_{P \in \mathbb{P}_{n-1}} P$. For no two conjunctions D_1^* and D_2^* in $\mathcal{D}_{\min}^*$, there are a component c , sets M_1 and M_2 , and a conjunction $\bar{D}$, not containing c , such that $D_1^* \simeq \bar{D} \wedge c \in M_1$ and $D_2^* \simeq \bar{D} \wedge c \in M_2$.

Proof: Assume that $\mathcal{D}_{\min}^*$ has two conjunctions D_1^* and D_2^* such that $D_1^* \simeq \bar{D} \wedge c \in M_1$ and $D_2^* \simeq \bar{D} \wedge c \in M_2$, where the conjunction $\bar{D}$ does not contain c . Note that each conjunction in $\mathcal{D}^*$, and therefore also in $\mathcal{D}_{\min}^*$, is the conjunction of one P_i from each negated conflict in $\mathbb{P}$. Let the negated conflicts in $\mathbb{P}$ be indexed from 1 to $|\mathbb{P}|$. Let I_1 be the index set of exactly those negated conflicts that have an assignment P_i such that P_i is a part of D_1^* and P_i contains the component c .

To illustrate the notation introduced, consider the following example:

$$\begin{aligned} \mathbb{P}_3 &= \{P_1, P_2, P_3\} \\ &= \{P_{11} \vee P_{12} \vee P_{13}, P_{21} \vee P_{22}, P_{31} \vee P_{32} \vee P_{33}, P_{41} \vee P_{42}\}. \end{aligned}$$

Note that all negated conflicts P_j have the form (5). Let the assignments P_{11} , P_{21} , and P_{31} contain the component c , and for clarity, these have been marked with gray. Let $D_1^* = P_{11} \wedge P_{21} \wedge P_{32} \wedge P_{41}$. This means that $c \in M_1 \simeq P_{11} \wedge P_{21}$ and $\bar{D} \simeq P_{32} \wedge P_{41}$. The index set I_1 is uniquely determined to be $I_1 = \{1, 2\}$.

Now, to continue with the proof, let I_2 be the index set of exactly those negated conflicts that have an assignment P_i such that P_i is a part of D_2^* and that P_i contains the component c . Note that since $D_1^* \not\simeq D_2^*$, it holds that the sets M_1 and M_2 are distinct, and therefore, the sets I_1 and I_2 are also distinct.

Since each conjunction in $\mathcal{D}_{\min}^*$ is the conjunction of one P_i from each negated conflict in $\mathbb{P}$, it holds that, in D_1^* , $\bar{D}$ is formed by P_i 's from the negated conflicts I_1^C . Similarly, in D_2^* , $\bar{D}$ is formed by P_i 's from the negated conflicts I_2^C . Now, let D'_2 be the conjunction of those P_i 's in D_2^* that belong to the negated conflicts in the set $I_2^C \cap I_1$. Let D' be the conjunction of D'_2 and those P_i 's in D_1^* , not containing c . Note that $D' \simeq \bar{D}$.

To illustrate the notation, continue with the previous example, and let $D_2^* = P_{12} \wedge P_{21} \wedge P_{31} \wedge P_{42}$. Then, it holds that $I_2 = \{2, 3\}$, $I_2^C \cap I_1 = \{1\}$, $D'_2 = P_{12}$, and $D' = P_{12} \wedge P_{32} \wedge P_{41}$.

Next, let D_c be the conjunction of the P_i 's that belong to the negated conflicts $I_1 \cap I_2$ and are present in D_1^* . In the example, $I_1 \cap I_2 = \{2\}$ and $D_c = P_{21}$. Note that $c \in M_1 \models D_c$ and $c \in M_2 \models D_c$ always hold.

Let $D_3^* = D' \wedge D_c$, with D' and D_c being formed as described earlier, and note that D_3^* must be in $\mathcal{D}^*$. Also note that $D_1^* \simeq \bar{D} \wedge c \in M_1 \models D' \wedge D_c = D_3^*$ and, similarly, $D_2^* \models D_3^*$.

If $D_1^* \simeq D_3^*$, this would imply that $D_2^* \models D_3^* \simeq D_1^*$, which contradicts the starting assumption that $\mathcal{D}_{\min}^*$ contains both D_1^* and D_2^* . Therefore, $D_1^* \not\models D_3^*$ must hold. However, together with $D_1^* \models D_3^*$, this implies that D_1^* cannot be in $\mathcal{D}_{\min}^*$, which is a contradiction. ■

Lemma 3: Let $\mathcal{D}^*$ be the full expansion of $\bigwedge_{P' \in \mathbb{P}_{n-1}} P'$. Let $\mathcal{Q} = \mathcal{D}_{\text{old}} \vee \mathcal{D}_{\text{add}}$ be the output from Algorithm 2, given $\mathcal{D}_{\min}^*$ and $\mathcal{P}$ as inputs. If there is a $D_{i_m} \in \mathcal{D}_{\min}^*$ and a $P_j \in \mathcal{P}$, such that D_{i_m} is not contained in $\mathcal{D}_{\text{old}}$, and there is no conjunction $Q_l \simeq D_{i_m} \wedge P_j$ contained in $\mathcal{D}_{\text{add}}$ after running the algorithm, then there is a $D_{i_{m+1}}$ in $\mathcal{D}_{\min}^*$ 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 $\mathcal{D}_{\text{old}}$ means that the second loop of the algorithm must have been entered when $D_i = D_{i_m}$. Then, the fact that no $Q_l \simeq D_{i_m} \wedge P_j$ is contained in $\mathcal{D}_{\text{add}}$ means, according to line 9, that

$$D_{i_m} \wedge P_j \models D_k \quad (15)$$

for some $D_k \neq D_{i_m}$. By choosing $i_{m+1} = k$, this gives $D_{i_m} \wedge P_j \models D_{i_{m+1}}$.

Next, we will prove that $D_{i_{m+1}} \wedge P_j \not\models D_{i_m} \wedge P_j$. This is equivalent to proving $D_k \wedge P_j \not\models D_{i_m} \wedge P_j$. Let the single assignment in P_j be $a \in A_p$, and let comps D_i denote the set of components in D_i . We will divide the proof into three cases: 1) $a \notin \text{comps } D_i$; 2) $a \in \text{comps } D_i$, $a \notin \text{comps } D_k$; and 3) $a \in \text{comps } D_i$, $a \in \text{comps } D_k$.

- 1) The fact (15), or equivalently $D_i \wedge P_j \models D_k$, together with the fact that $a \notin \text{comps } D_i$, would imply that $D_i \models D_k$. This is a contradiction since $D_i \in \mathcal{D}$, $D_k \in \mathcal{D}$, and $\mathcal{D}$ is in the input required to be in MNF.
- 2) This case means that D_i can be written as $D_i = D' \wedge a \in A_i$, where $a \notin \text{comps } D'$, and the fact (15) becomes $D' \wedge a \in A_i \cap A_p \models D_k$. This, together with the fact that $a \notin \text{comps } D_k$, implies that $D' \models D_k$ and, consequently, that $D_i \models D_k$, which is a contradiction since $\mathcal{D}$ is in MNF.
- 3) Assume that $D_k \wedge P_j \models D_i \wedge P_j$. This relation can be written as $D'_k \wedge a \in A_p \cap A_k \models D'_i \wedge a \in A_p \cap A_i$, where D'_k and D'_i are conjunctions not containing component a . This relation would imply that $D'_k \models D'_i$. Furthermore, the fact (15) becomes $D'_i \wedge a \in A_p \cap A_i \models$

$D'_k \wedge a \in A_k$, 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 would be the assignment of component a . Lemma 2 says that 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 full expansion of $\bigwedge_{P \in \mathbb{P}_{n-1}} P$. Let $\mathcal{Q}$ be the output from Algorithm 2, given $\mathcal{D}_{\min}^*$ and $\mathcal{P}$ as inputs. For each conjunction D_i in $\mathcal{D}_{\min}^*$ 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 $\mathcal{D}_{\text{old}}$, then the lemma is trivially fulfilled. If a $Q_l \simeq D_i \wedge P_j$ is instead contained in $\mathcal{D}_{\text{add}}$, then the lemma is also trivially fulfilled. Study now the case where D_i is not contained in $\mathcal{D}_{\text{old}}$ and no $Q_l \simeq D_i \wedge P_j$ is contained in $\mathcal{D}_{\text{add}}$. We can then apply Lemma 3 with $i_m = i$. This gives us a $D_{i_{m+1}}$ in $\mathcal{D}_{\min}^*$ such that $D_{i_m} \wedge P_j \models D_{i_{m+1}}$.

If $D_{i_{m+1}}$ is contained in $\mathcal{D}_{\text{old}}$, then the lemma is fulfilled with $Q_k = D_{i_{m+1}}$. If a $Q_v \simeq D_{i_{m+1}} \wedge P_j$ is instead contained in $\mathcal{D}_{\text{add}}$, then note that $D_{i_m} \wedge P_j \models D_{i_{m+1}}$ implies that $D_{i_m} \wedge P_j \models D_{i_{m+1}} \wedge P_j \simeq Q_v$. This means that the lemma is fulfilled with $Q_k = Q_v$. In this way, we can repeatedly apply Lemma 3 as long as the new $D_{i_{m+1}}$ obtained is not contained in $\mathcal{D}_{\text{old}}$ and there is no $Q_v \simeq D_{i_{m+1}} \wedge P_j$ contained in $\mathcal{D}_{\text{add}}$.

We will now prove that, after a finite number of applications of Lemma 3, we obtain a $D_{i_{m+1}}$ such that $D_{i_{m+1}}$ is contained in $\mathcal{D}_{\text{old}}$ or there is a $Q_v \simeq D_{i_{m+1}} \wedge P_j$ contained in $\mathcal{D}_{\text{add}}$. Note that each application of Lemma 3 guarantees that $D_{i_m} \wedge P_j \models D_{i_{m+1}} \wedge P_j$ and $D_{i_{m+1}} \wedge P_j \not\models D_{i_m} \wedge P_j$. These two properties imply that, in the series of applications of Lemma 3, all conjunctions obtained are unique, i.e., all conjunctions $D_{i_m}, D_{i_{m+1}}, D_{i_{m+2}}, \dots$ are unique. This means that the maximum number of times that Lemma 3 can be applied in this way is limited by the number of conjunctions in $\mathcal{D}$.

Assume now that Lemma 3 has been applied by a maximum number of times (which equals the number of conjunctions in $\mathcal{D}$ minus 1), and we have not obtained any $D_{i_{m+1}}$, where $D_{i_{m+1}}$ is contained in $\mathcal{D}_{\text{old}}$, or there is a $Q_v \simeq D_{i_{m+1}} \wedge P_j$ contained in $\mathcal{D}_{\text{add}}$. Then, Lemma 3 actually says that we can apply it once more and obtain a new set $D_{i_{m+1}}$. Since all conjunctions obtained from Lemma 3 are unique, we cannot obtain a previous conjunction, but there are also no conjunctions left. This is therefore a contradiction that proves that when Lemma 3 has been applied by a maximum number of times, we must obtain a conjunction $D_{i_{m+1}}$, where $D_{i_{m+1}}$ is contained in $\mathcal{D}_{\text{old}}$, or there is a $Q_v \simeq D_{i_{m+1}} \wedge P_j$ contained in $\mathcal{D}_{\text{add}}$. ■

Lemma 5: Let $\mathcal{Q}$ be the output from Algorithm 2 after processing all negated conflicts in $\mathbb{P}$. Let $\mathcal{Q}^*$ be the full expansion of $\bigwedge_{P \in \mathbb{P}} P$. Then, there is a one-to-one correspondence between the conjunctions in $\mathcal{Q}$ and $\mathcal{Q}_{\min}^*$ such that, for each conjunction Q_i in $\mathcal{Q}$, there is a unique conjunction Q_i^* in $\mathcal{Q}_{\min}^*$, where $Q_i \simeq Q_i^*$ and vice versa.

Proof: The proof is constructed by induction over n . For a given n , let $\mathcal{Q}^*$ be a full expansion of $\bigwedge_{P \in \mathbb{P}_n} P$. For the induction start, let $n = 1$, which means that $\mathbb{P}_n$ consists of only one negated conflict $\mathcal{P}$. As stated in Section IV, the algorithm is not needed in this case since $\mathcal{P}$ is already in MNF. That is, the output after processing this single conflict is $\mathcal{Q} = \mathcal{P}$. Since

$n = 1$, it also holds that $\mathcal{Q}^* = \mathcal{P}$. Then, trivially, it holds that, for each conjunction Q_i in $\mathcal{Q}$, there is a unique conjunction Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i \simeq Q_i^*$, and that, for each Q_i^* in $\mathcal{Q}_{\min}^*$, there is a unique Q_i in $\mathcal{Q}$ such that $Q_i \simeq Q_i^*$.

For the induction step, consider an arbitrary $n > 1$. Let $\mathcal{D}^*$ be a full expansion of $\bigwedge_{\mathcal{P} \in \mathbb{P}_{n-1}} \mathcal{P}$. Let $\mathcal{D}$ be the algorithm output after having processed all negated conflicts in $\mathbb{P}_{n-1}$. Assume that, for each conjunction D_i in $\mathcal{D}$, there is a unique conjunction D_i^* in $\mathcal{D}_{\min}^*$ such that $D_i \simeq D_i^*$, and that, for each D_i^* in $\mathcal{D}_{\min}^*$, there is a unique D_i in $\mathcal{D}$ such that $D_i \simeq D_i^*$. Without loss of generality, we can then assume that $\mathcal{D} = \mathcal{D}_{\min}^*$.

Let $\mathcal{Q}$ be the algorithm output when feeding it with $\mathcal{D} = \mathcal{D}_{\min}^*$ and a new negated conflict $\mathcal{P}$. Let $\mathcal{Q}_{\min}^*$ be constructed from $\mathbb{P}_n$. We will prove hereinafter that, for each conjunction Q_i in $\mathcal{Q}$, there is a conjunction Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i \simeq Q_i^*$, and that, for each Q_i^* in $\mathcal{Q}_{\min}^*$, there is a Q_i in $\mathcal{Q}$ such that $Q_i \simeq Q_i^*$.

Consider an arbitrary conjunction Q_1 in $\mathcal{Q}$. Because of line 16, Q_1 is in $\mathcal{D}_{\text{old}}$ or $\mathcal{D}_{\text{add}}$. First, we consider the case when Q_1 is in $\mathcal{D}_{\text{old}}$. Since Q_1 is in $\mathcal{D}_{\text{old}}$, then $Q_1 = D_i$ for a D_i in $\mathcal{D}$. Because of lines 4 and 5, it holds that $D_i \models \mathcal{P}$, and there is therefore, according to the discussion in Section IV-C, a conjunction P_j in $\mathcal{P}$ such that $D_i \models P_j$. Thus, $D_i \wedge P_j \simeq D_i$, and therefore, $Q_1 \simeq D_i \wedge P_j$. By definition, the conjunction $D_i \wedge P_j$ is in $\mathcal{Q}^*$, so we have shown, for the case Q_1 is in $\mathcal{D}_{\text{old}}$, that there is a $Q_1^* = D_i \wedge P_j \simeq Q_1$ in $\mathcal{Q}^*$.

Next, assume that there is no Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i^* \simeq Q_1^*$. This would mean that there is another $Q_2^* = D_k \wedge P_l$ in $\mathcal{Q}^*$ such that $Q_1^* \models Q_2^*$ and $Q_2^* \not\models Q_1^*$. Note that D_k is in $\mathcal{D}^*$. Now, there are two possible cases: 1) $k \neq i$ and 2) $k = i, j \neq l$.

- 1) Since $D_i \simeq D_i \wedge P_j$ and $i \neq k$, we have the relation $D_i \simeq D_i \wedge P_j \simeq Q_1^* \models Q_2^* \simeq D_k \wedge P_l \models D_k$. Also, we have $D_k \wedge P_l \simeq Q_2^* \not\models Q_1^* \simeq D_i \wedge P_j \simeq D_i$. This implies that $D_k \not\models D_i$. However, since D_i is in $\mathcal{D}_{\min}^*$, there cannot be any D_k in $\mathcal{D}^*$ such that $D_i \models D_k$ and $D_k \not\models D_i$. Thus, we have a contradiction.
- 2) Since $D_i \simeq D_i \wedge P_j$, we have the relation $D_i \simeq D_i \wedge P_j \simeq Q_1^* \models Q_2^* \simeq D_i \wedge P_l \models P_l$. This means that $D_i \simeq D_i \wedge P_l$ and, furthermore, that $D_i \simeq D_i \wedge P_l \simeq Q_2^* \not\models Q_1^* \simeq D_i \wedge P_j \simeq D_i$, which is a contradiction.

In conclusion, these contradictions show, for the case Q_1 is in $\mathcal{D}_{\text{old}}$, that there is a Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i^* \simeq Q_1$.

Next, we consider the case when Q_1 is in $\mathcal{D}_{\text{add}}$. Since Q_1 is in $\mathcal{D}_{\text{add}}$, the second loop of the algorithm has been entered with a D_i in $\mathcal{D}$ and P_j in $\mathcal{P}$ such that $Q_1 \simeq D_i \wedge P_j$. Therefore, $Q_1 \simeq D_i \wedge P_j$, and by definition, we have that $D_i \wedge P_j$ is in $\mathcal{Q}^*$. Thus, we have shown that there is a $Q_1^* \simeq D_i \wedge P_j \simeq Q_1$ in $\mathcal{Q}^*$.

Next, assume that there is no Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i^* \simeq Q_1^*$. This would mean that there is another $Q_2^* \simeq D_k \wedge P_l$ in $\mathcal{Q}^*$ such that $Q_1^* \models Q_2^*$. Now, there are two possible cases: 1) $k \neq i$ and 2) $k = i, j \neq l$.

- 1) Since Q_1 is in $\mathcal{D}_{\text{add}}$, and according to lines 8 and 9, it must hold that $D_i \wedge P_j \not\models D_k$. At the same time, $Q_1^* \models Q_2^*$ implies that $D_i \wedge P_j \simeq Q_1^* \models Q_2^* \simeq D_k \wedge P_l \models D_k$, which is a contradiction.

- 2) We have that $D_i \wedge P_j \simeq Q_1^* \models Q_2^* \simeq D_k \wedge P_l \models P_l$. According to (5), P_j does not contain the same component as P_l . Then, $D_i \wedge P_j \models P_l$ implies that $D_i \models P_l$. This, in turn, implies that $D_i \models \mathcal{P}$ and, consequently, according to line 4, that the second loop is not entered, which is a contradiction.

We have shown here that, for the case Q_1 is also in $\mathcal{D}_{\text{add}}$, there is a Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i^* \simeq Q_1$.

In conclusion, when we feed the algorithm with $\mathcal{D} = \mathcal{D}_{\min}^*$ and $\mathcal{P}$, it holds that, for each conjunction Q_i in $\mathcal{Q}$, there is a conjunction Q_i^* in $\mathcal{Q}_{\min}^*$ such that $Q_i \simeq Q_i^*$. From the definition of $\mathcal{Q}_{\min}^*$, it also holds trivially that Q_i^* is unique, i.e., there is no other Q_{i2}^* in $\mathcal{Q}_{\min}^*$ such that $Q_i^* \simeq Q_{i2}^*$. Left to prove now is that, for each Q_i^* in $\mathcal{Q}_{\min}^*$, there is a unique Q_i in $\mathcal{Q}$ such that $Q_i \simeq Q_i^*$.

Take an arbitrary Q_i^* in $\mathcal{Q}_{\min}^*$. The conjunctions of $\mathcal{Q}_{\min}^*$ must be a subset of the conjunctions of the full expansion of $\mathcal{D}_{\min}^* \wedge \mathcal{P}$. Therefore, there is a D_i in $\mathcal{D}_{\min}^*$ and a P_j in $\mathcal{P}$ such that $Q_i^* = D_i \wedge P_j$. Since we feed the algorithm with $\mathcal{D}_{\min}^*$ and $\mathcal{P}$, we can apply Lemma 4, which tells us that there is a Q_k in $\mathcal{Q}$ such that $D_i \wedge P_j \models Q_k$.

We have previously concluded that since Q_k is in $\mathcal{Q}$, there is a conjunction Q_l^* in $\mathcal{Q}_{\min}^*$ such that $Q_k \simeq Q_l^*$. Thus, we have that $Q_i^* \models Q_k \simeq Q_l^*$, where both Q_i^* and Q_l^* are in $\mathcal{Q}_{\min}^*$. Due to the definition of $\mathcal{Q}_{\min}^*$, this must mean that $Q_i^* \equiv Q_l^*$. Thus, we have the relation $Q_i^* \models Q_k \simeq Q_l^* \equiv Q_i^*$, which implies that $Q_i^* \simeq Q_k$. In conclusion, with $Q_i = Q_k$, we have proven that, for each Q_i^* in $\mathcal{Q}_{\min}^*$, there is a Q_i in $\mathcal{Q}$ such that $Q_i \simeq Q_i^*$. Finally, a consequence of Lemma 1 is that Q_i , i.e., there is no other Q_{i2} in $\mathcal{Q}$ such that $Q_{i2} \simeq Q_i$. ■

Theorem 1: 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, the following hold.

- 1) $\mathcal{Q} \simeq \mathbb{P}$.
- 2) $\mathcal{Q}$ is in MNF.

Proof: For the 1) part of the theorem, consider $\mathcal{Q}_{\min}^*$ obtained from $\mathbb{P}$. By definition of $\mathcal{Q}_{\min}^*$, it holds that $\mathcal{Q}_{\min}^* \simeq \mathbb{P}$. Then, $\mathcal{Q} \simeq \mathbb{P}$ is a trivial consequence of Lemma 5. □

For the 2) part of the theorem, note first that Lemma 1 $\mathcal{Q}$ says that it contains no two conjunctions such that $Q_2 \models Q_1$. Also, we need to prove that each conjunction is in the form specified by (1).

All conjunctions in $\mathcal{D}_{\text{add}}$ are on the form (1) because of the requirement on D_{new} in line 7. Therefore, all conjunctions added in the process of forming $\mathcal{Q}$ from the set $\mathbb{P}$ are on the form (1). Possibly, there might also be conjunctions in $\mathcal{Q}$ not added via $\mathcal{D}_{\text{add}}$ but instead originating from the first negated conflict $\mathcal{P}$ in $\mathbb{P}$. However, since $\mathcal{P}$ is, by definition, on the form (1), it holds that all conjunctions in $\mathcal{Q}$ must be on the form (1). ■

Proof of Theorem 2

Lemma 6: Let each component have only two possible behavioral modes, let d be a partial diagnosis with respect to $\mathbb{P}$, and let $\mathcal{Q}$ be the output from Algorithm 2 after processing

all negated conflicts $\mathbb{P}$. Then, it holds that $d \models Q_v$ for some Q_v in $\mathcal{Q}$.

Proof: From the definition of partial diagnosis, it holds that $d \models \mathbb{P}$. This means that, for each negated conflict $\mathcal{P}$ in $\mathbb{P}$, it holds that $d \models \mathcal{P}$. Then, note that each $\mathcal{P}$ in $\mathbb{P}$ is a disjunction of unique assignments, e.g., $c = N$. The fact $d \models \mathcal{P}$ implies, according to the discussion in Section IV-C, that each $\mathcal{P}$ contains at least one of the assignments in d . Create D^* by taking the conjunction of one of these assignments from each $\mathcal{P}$ in $\mathbb{P}$. It will then hold that $d \models D^*$. By construction, D^* is a conjunction in $\mathcal{Q}^*$. Then, by construction of $\mathcal{Q}_{\min}^*$, there must be a conjunction Q_v^* in $\mathcal{Q}_{\min}^*$ such that $D^* \models Q_v^*$. According to Lemma 5, there is a conjunction Q_v in $\mathcal{Q}$ such that $Q_v \simeq Q_v^*$. Thus, we have $d \models D^* \models Q_v^* \simeq Q_v$. ■

Theorem 2: 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: Take an arbitrary conjunction Q_k in $\mathcal{Q}$. From Theorem 1.1), we know that $\mathcal{Q} \simeq \mathbb{P}$. Thus, we have $Q_k \models \mathcal{Q} \simeq \mathbb{P}$, which means that Q_k is a partial diagnosis.

Now, assume that there is another partial diagnosis d' such that $Q_k \models d'$. Note that this also means that $Q_k \not\models d'$. Since d' is a partial diagnosis, Lemma 6 implies that there is a Q_v in $\mathcal{Q}$ such that $d' \models Q_v$. Thus, we have $Q_k \models d' \models Q_v$. This, together with $Q_k \not\models d'$, contradicts the fact that $\mathcal{Q}$ is in MNF, which is stated by Theorem 1.2). The contradiction means that there is no other partial diagnosis d' such that $Q_k \models d'$, and Q_k must therefore be a kernel diagnosis. ■

REFERENCES

- [1] R. Reiter, "A theory of diagnosis from first principles," *Artif. Intell.*, vol. 32, no. 1, pp. 57–95, Apr. 1987.
- [2] J. de Kleer and B. Williams, "Diagnosing multiple faults," *Artif. Intell.*, vol. 32, no. 1, pp. 97–130, Apr. 1987.
- [3] R. Greiner, B. Smith, and R. Wilkerson, "A correction to the algorithm in Reiter's theory of diagnosis," *Artif. Intell.*, vol. 41, no. 1, pp. 79–88, Nov. 1989.
- [4] F. Wotawa, "A variant of Reiter's hitting-set algorithm," *Inf. Process. Lett.*, vol. 79, no. 1, pp. 45–51, May 2001.
- [5] J. de Kleer, A. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," *Artif. Intell.*, vol. 56, no. 2/3, pp. 197–222, Aug. 1992.
- [6] P. Struss and O. Dressler, "'Physical Negation'—Integrating fault models into the general diagnosis engine," in *Proc. IJCAI*, 1989, pp. 1318–1323.
- [7] J. de Kleer and B. Williams, "Diagnosis with behavioral modes," in *Proc. IJCAI*, 1989, pp. 1324–1330.
- [8] B. Williams and R. Ragno, "Conflict-directed A* and its role in model-based embedded systems," *Discrete Appl. Math.*, vol. 155, no. 12, pp. 1562–1595, Jun. 2007.
- [9] O. Dressler and P. Struss, "Back to defaults: Characterizing and computing diagnoses as coherent assumption sets," in *Proc. ECAI*, 1992, pp. 719–723.
- [10] B. Pulido and C. González, "Possible conflicts, ARRs, and conflicts," in *Proc. Int. Workshop Principles Diagnosis (DX)*, Semmering, Austria, 2002, pp. 122–128.
- [11] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.
- [12] R. Patton, P. Frank, and R. Clark, Eds., *Issues of Fault Diagnosis for Dynamic Systems*. New York: Springer-Verlag, 2000.
- [13] Z. Gao, T. Breikin, and H. Wang, "Reliable observer-based control against sensor failures for systems with time delays in both state and input," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 5, pp. 1018–1029, Sep. 2008.
- [14] Z. Gao, X. Shi, and S. Ding, "Fuzzy state/disturbance observer design for T–S fuzzy systems with application to measurement fault estimation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 3, pp. 875–880, Jun. 2008.
- [15] S. Ruan, Y. Zhou, F. Yu, K. Pattipati, P. Willett, and A. Patterson-Hine, "Dynamic multiple-fault diagnosis with imperfect tests," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 6, pp. 1224–1236, Nov. 2009.
- [16] H. Paiva, R. Galvao, and T. Yoneyama, "A wavelet band-limiting filter approach for fault detection in dynamic systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 3, pp. 680–687, May 2008.
- [17] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 3, pp. 348–361, May 2007.
- [18] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. New York: Springer-Verlag, 2006.
- [19] J. Luo, M. Namburu, K. Pattipati, L. Qiao, and S. Chigusa, "Integrated model-based and data-driven diagnosis of automotive antilock braking systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 2, pp. 321–336, Mar. 2010.

Mattias Nyberg was born in Karlstad, Sweden, in 1970. He received the M.Sc. degree in computer science and engineering from Linköping University, Linköping, Sweden, in 1994, the M.Sc. degree in electrical engineering and applied physics from Case Western Reserve University, Cleveland, OH, in 1994, and the Ph.D. degree in electrical engineering from Linköping University in 1999.

He has spent one year with Daimler Chrysler Research, Stuttgart, Germany, as a Postdoctoral Researcher. He is currently partly an Associate Professor with the Department of Electrical Engineering, Linköping University, and partly a Technical Manager of On-Board Diagnosis with Scania CV AB, Södertälje, Sweden, where he is responsible for development and research in on-board diagnosis. His research interest is model-based diagnosis from the perspectives of both computer science and control theory. In particular, he is interested in its uses for fault-tolerant control of mechatronic systems and for decision support for repair mechanics.