ELSEVIER

Artificial Intelligence 104 (1998) 71–106

Artificial Intelligence

Model-based diagnostics and probabilistic assumption-based reasoning [☆]

J. Kohlas [a,*], B. Anrig [a], R. Haenni [a], P.A. Monney [b]

[a] *Institute of Informatics, University of Fribourg, rue Faucigny 2, CH-1700 Fribourg, Switzerland*

[b] *Seminar of Statistics, University of Fribourg, Beauregard 11-13, CH-1700 Fribourg, Switzerland*

Received 30 September 1996; received in revised form 29 September 1997

Abstract

The mathematical foundations of model-based diagnostics or diagnosis from first principles have been laid by Reiter (1987). In this paper we extend Reiter's ideas of model-based diagnostics by introducing probabilities into Reiter's framework. This is done in a mathematically sound and precise way which allows one to compute the posterior probability that a certain component is not working correctly given some observations of the system. A straightforward computation of these probabilities is not efficient and in this paper we propose a new method to solve this problem. Our method is logic-based and borrows ideas from assumption-based reasoning and ATMS. We show how it is possible to determine arguments in favor of the hypothesis that a certain group of components is not working correctly. These arguments represent the symbolic or qualitative aspect of the diagnosis process. Then they are used to derive a quantitative or numerical aspect represented by the posterior probabilities. Using two new theorems about the relation between Reiter's notion of conflict and our notion of argument, we prove that our so-called degree of support is nothing but the posterior probability that we are looking for. Furthermore, a model where each component may have more than two different operating modes is discussed and a new algorithm to compute posterior probabilities in this case is presented. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Model-based diagnostics; Assumption-based reasoning; ATMS; Probability; Propositional logic; Support; Algorithm of Abraham

1. Introduction and overview

It can happen that input-output systems made up of several components do not operate as they were designed to do. This discrepancy between the theoretical and actual behavior

☆ Research supported by grant No. 2100-042927.95 of the Swiss National Foundation for Research.

* Corresponding author. Email: juerg.kohlas@unifr.ch.

of the system occurs when there is a difference between the actual output of the system and the output that should have been observed with a system that is working correctly. Such a difference is due to the malfunctioning of one or more components of the system and the diagnosis problem is to identify those faulty components responsible for the malfunctioning of the system as a whole. The identification of faulty components is the diagnosis problem and there are many theories and models to handle it. In this paper we follow and extend the ideas put forward by Reiter [31] in his theory of diagnosis from first principles. Other substantial contributions to model-based diagnostic reasoning are those of Davis [9], de Kleer [10], de Kleer and Williams [13], Genesereth [16], Reggia et al. [29,30].

The goal of this paper is to present a general theory of diagnosis in the sense that it can be used as an aid to identify the faulty components that should be replaced. We use a logic-based approach to solve the diagnosis problem. As in Reiter [31], it is supposed that the system is made up of a set $C = \{c_1, \ldots, c_n\}$ of components. The predicate $ab(c_i)$ indicates that the component c_i is faulty, is in an abnormal state. It is not excluded that several components are faulty. The way the entire system is working, i.e., how the different components interact in the system, is described by a set X_S of logical formulas in a specific language $\mathcal{L}$ built from the predicates $ab(c_i)$, $i = 1, \ldots, n$, as well as other atoms and elements. The observed input and output values of the system are also represented by a set of formulas X_O in the language $\mathcal{L}$.

Then a probabilistic model is introduced into this framework, an aspect that is not considered at all in the paper of Reiter [31]. Before observing anything about the system, prior probabilities are assigned to the components: let p_i denote the probability that the component c_i is intact, i.e., c_i is working correctly. With each component c_i we associate a Boolean variable x_i indicating whether the component is intact ($x_i = 1$) or faulty ($x_i = 0$). It is assumed that the variables x_i are independent random variables. A Boolean vector $x = (x_1, \ldots, x_n)$ in $\{0, 1\}^n$ is called a system state. Then, before any observation is made, the probability of a particular system state $x = (x_1, \ldots, x_n)$ is given by

$$p(x) = \prod_{i=1}^{n} p_i^{x_i} (1 - p_i)^{(1-x_i)}. \tag{1}$$

p is a probability distribution on the set of all system states $\Omega = \{0, 1\}^n$. Now suppose that an observation of the system is made, for example, the values of some variables are measured. These observation are described by a set of formulas X_O in the language $\mathcal{L}$. Then, given this new information represented by the observations, many system states become impossible. Let N_c denote the set of impossible system states and let N_d denote the complementary set of system states that remain compatible with the observations. A system state $x \in N_d$ is a possible explanation of the observations. In other words, it is a possible "diagnosis" of the system. On the other hand, a system state $x \in N_c$ is in conflict with the observations and hence is called a "conflict". So, given the observations, the only possible system states are those in the set N_d. In accordance with the rules of probability theory, this means that the prior probability measure p defined on the set of all system states $\{0, 1\}^n$ by Eq. (1) must be conditioned on the new event N_d, which leads to the new probability measure p' given by

$$p'(x) = \begin{cases} p(x)/p(N_d) & \text{if } x \in N_d, \\ 0 & \text{if } x \in N_c, \end{cases} \tag{2}$$

where $p(N_d)$ is obtained by summing the $p(x)$ over all $x \in N_d$. The probability measure p' is the posterior probability of the system states given the observations. For example, this probability measure can be used to compute the (posterior) probability that component c_i is faulty as follows. If N_i denotes the set of all system states x such that $x_i = 0$ then the event that c_i is faulty is represented by N_i. The prior probability of this event is p_i whereas its posterior probability is given by

$$p'(N_i) = p(N_i|N_d) = \frac{p(N_i \cap N_d)}{p(N_d)} = \sum_{x \in N_i \cap N_d} p'(x). \tag{3}$$

The posterior probability of other events of interest can be computed in a similar way. The posterior distribution gives information about the likelihood of components being faulty. This information is useful for decision making.

Efficient methods for the computation of this posterior distribution is the main subject of this paper. Since there are 2^n states, a direct computation of p' is not feasible even for systems containing a relatively small number n of components. Therefore, a logic-based method to cope with the complexity of this problem is proposed.

The determination of the event N_d, the event on which the probability p is conditioned upon, is of fundamental importance in our approach. Since the observations only implicitly define N_d, it is important to find a way to represent this set. To do this, ideas and theorems presented by Reiter [31] are used. Basically, the set N_d will be represented by a logical formula, more precisely a disjunctive normal form, in the propositional language $\mathcal{S}$ generated by the atoms $ab(c_i)$, $i = 1, \ldots, n$. How to do this is explained in Reiter's paper [31]. When this is done, the problem of computing p' is not solved directly, but another, indirect route will be taken which proves to have a lot of advantages as will be explained. Starting from the symbolic representation of N_d as a logical formula in $\mathcal{S}$, we develop a theory based on the ideas of probabilistic assumption-based reasoning and ATMS proposed by Provan [28], Laskey and Lehner [25] and Kohlas and Monney [22]. As will be explained in the paper, this leads to the determination of the so-called support of an event $H \subseteq \Omega$, which is a logical formula $s(H)$ in the language $\mathcal{S}$ that entails the event H given the system description and the observations (see below for details).

Then it is shown that the probability of provability of H, that is the so-called degree of support of H, is in fact equal to the posterior probability of H we are looking for. In other words, if $sp(H)$ denotes the degree of support of H, then

$$p'(H) = sp(H) \tag{4}$$

for all $H \subseteq \Omega$.

As will be shown in this paper, Eq. (4) is an important new result that allows to efficiently compute the posterior probability of events in Ω. It is a consequence of properties of supports of events that are proved in this paper for the first time. But Eq. (4) is not only important from a computational perspective, but it is also interesting in itself because it establishes in a clear and mathematically precise manner the equivalence between classical posterior probabilities on the one hand, and degrees of support (or degrees of belief as defined in ATMS and assumption-based reasoning) on the other hand. This equivalence is also an interesting contribution of this paper. The following list states some advantages of the symbolic, logic-based approach to compute $p'(H)$ that is presented in this paper:

- a sensitivity analysis of $p'(H)$ with respect to the prior probabilities of the components is possible because $sp(H)$ is derived from the logical formula $s(H)$ and from the prior probabilities of the components (see below);
- due to the properties of the supports of events that are established in this paper, it is not necessary to recompute everything when the posterior probability of another event is needed;
- the support $s(H)$ of the event H can be viewed as an argument supporting this event. This logical formula $s(H)$ can therefore be seen as a symbolic explanation of the numerical value $p'(H) = sp(H)$.

Finally, let us emphasize that our approach also applies to situations where the components have any number of possible operating modes (usually it is assumed that there are only two possible operating modes: intact and faulty).

The paper is organized as follows: Section 3 presents a simple and classical example that serves as an introduction and preparation for the general theory exposed in Section 4. The probabilistic structure is introduced into the model in Section 5. In particular, the case of components with more than two possible operating modes (more than one fault state) is analyzed. Finally, Appendix A presents a new algorithm for computing probabilities of disjunctive normal forms for systems with more than two operating modes; Appendix B lists the proofs of the theorems.

2. Comparison with related work

In Reiter's theory [31], the problem is to find the so-called diagnoses: a diagnosis is a collection of components whose malfunctioning is sufficient to explain the actual observations of the system. In other words, assuming that all components of a diagnosis are faulty, the actual observations X_O become compatible with the system description X_S. Of course, the components of a diagnosis are natural candidates for replacement, but since in general there are many different diagnoses, the problem is then to evaluate their respective value in order to reduce their number. In Reiter [31], it is proposed to make additional measurements to help discriminate between diagnoses. Discrimination between diagnoses is achieved by using more and more information in the form of additional observations, which turns the analysis into an incremental process. Using the notion of conflict set, Reiter [31] also shows that the task of computing diagnoses can essentially be done using theorem proving techniques. The attractiveness of Reiter's approach is that we are provided with an explicit list of diagnoses, and therefore of components to replace, at each step of the incremental process. However, as shown in Reiter's paper, the incremental discrimination process is not necessarily monotone: an additional observation does not guarantee that the new list of diagnoses will be shorter. Unlike ours, Reiter's approach is purely logical or symbolic as there is no numerical or probabilistic aspect involved.

De Kleer and Williams [13] proposed one way of introducing probabilities into diagnostics problems. They consider the notion of a candidate, which is defined as a set of components whose malfunctioning makes the observations compatible with the system description if it is assumed that the components not in the candidate are working correctly. Note that the notion of a candidate is not exactly the same as the notion of a diagnosis in Reiter's paper [31]. Then, de Kleer and Williams [13] determine candidates and also probabilities of candidates. When there is no candidate having a probability that is much higher than the probability of the other candidates, it is necessary to obtain more information in order to discriminate between the candidates. This is achieved by observing the value of some variable somewhere in the system. Given the new observation, the posterior probability of the remaining candidates is then computed according to Bayes' rule. So this is again an incremental process. De Kleer and Williams [13] also show how minimal entropy techniques can be used to guide the selection of the variable to be observed next.

Now let us say a few words about the differences between our approach and the one proposed by de Kleer and Williams [13]. First, it can be easily proved that the posterior probability of a candidate according to their analysis is in general different from the posterior probability of a candidate according to our analysis. This is a consequence of the model they are using, which, in order to be able to apply Bayes rule, absolutely requires a known conditional distribution of some observable variable given a candidate. The problem is that, in most cases, the specification of a candidate does not permit to derive a probability distribution on such a variable, but only a belief function in the sense of the theory of evidence [33]. So de Kleer and Williams [13] simply assume a uniform conditional distribution in such cases, which in our opinion is a somewhat unjustified step. On the other hand, in our theory there is no need for such conditional distributions in order to be able to compute the posterior distribution. This is a big advantage over de Kleer and Williams' [13] approach. The point is that our theory naturally remains within the classical theory of probability without a need for extraneous assumptions. Also, in our theory, the computation of posterior probabilities is not limited to candidates as in de Kleer and Williams' [13] approach: the posterior probability of any event involving components can be computed. For example, we can compute the posterior probability that both components c_i and c_j are faulty.

Our approach is an extension of those of Provan [28] and Laskey and Lehner [25] which are restricted to propositional logic. In our theory, only the assumptions about the status of the components are represented in propositional logic, but not necessarily the description of the system itself. Also, our framework can be used to analyze systems where each component has more than the usual two operating modes (intact or faulty). In addition, important new properties of the so-called quasi-supports are established.

Diagnosis problems can also be modeled and analyzed with Bayesian networks. For example, Pearl [27] considers the problem of diagnosing a simple input-output arithmetical circuit and solves it with Bayesian networks as follows. First, the arithmetical circuit is modeled as a causal network. Then it is turned into a Bayesian network by introducing link probabilities of a node given its parents (see [27, Eq. (5.54), p. 265]) and by assigning prior probabilities to nodes without parents. Prior probabilities must be assigned to two different kinds of nodes in the causal network: those corresponding to input variable and those corresponding to the components themselves. Since the values of the input variables are known it is easy to give a prior probability for them: simply set this probability to 1 at the observed value of the variable and 0 elsewhere. As in our approach, prior probabilities are also assigned to nodes corresponding to components of the circuit. In order to have a complete Bayesian network, the conditional distribution of the output of a component given certain values of its input variables and given its operating status (good or bad) must be specified. Then, given the values of the output variables that are observed, the posterior probabilities of the nodes corresponding to components of the circuit are computed using traditional message propagation algorithms. The result is a mathematical formula giving, for each component of the system, the posterior probability that it is faulty.

The determination of conditional distributions is easy when the component is working correctly, but when this is not the case Pearl arbitrarily assumes a uniform conditional distribution on the output (regardless of the input values). In our opinion, this arbitrariness is a major weak point of the Bayesian network approach. Generally speaking, the determination of the conditional distributions is a major problem with Bayesian networks. In contrast, with the method presented here, only prior probabilities on the components are needed in order to compute posterior probabilities on the components. The effect of faulty components is specified in the logical system description in an appropriate way. Our approach is therefore based on a more precise and explicit representation of the knowledge about the system. As a consequence, the results obtained by Pearl are different from ours.

It is also interesting to compare the theory developed in this paper with Shenoy's valuation-based systems [34]. When all the variables have a finite number of possible values and when the components themselves have a finite number of possible working modes, it can be proved that a valuation-based model based on belief functions (Kohlas et al. [24]) will give the same posterior probabilities as our approach. However, belief functions on variables with infinite domains are not tractable. Moreover, valuation-based models do not give the qualitative, symbolic information represented by the support $s(H)$. Therefore, the explanation aspect of the diagnosis analysis is again completely absent, it is a purely numerical technique. In addition, the computational method presented in this paper is completely different from the local propagation algorithm of Shafer–Shenoy [34].

The paper of Darwiche [8] considers a model that is very similar to ours. The discussion about the differences and similarities between the two approaches is given further down in the paper because we need to present our model in more details first. This discussion is given at the end of Section 4.

3. An informal introduction using a simple example

In this section we introduce informally our approach to model-based diagnostics. For that purpose we discuss a simple example of a faulty arithmetical network. The example has been introduced by Davis [9] and subsequently used in many papers on model-based diagnostics (e.g., Reiter [31] and de Kleer and Williams [13]). The aim of this informal introduction is to illustrate and clarify the technique of model-based diagnostics and to motivate our formal approach presented in Section 4 and Section 5.

Example 1. Fig. 1 shows a network consisting of three multipliers and two adders. The network has six input and two output ports. Obviously, the given input and the observed

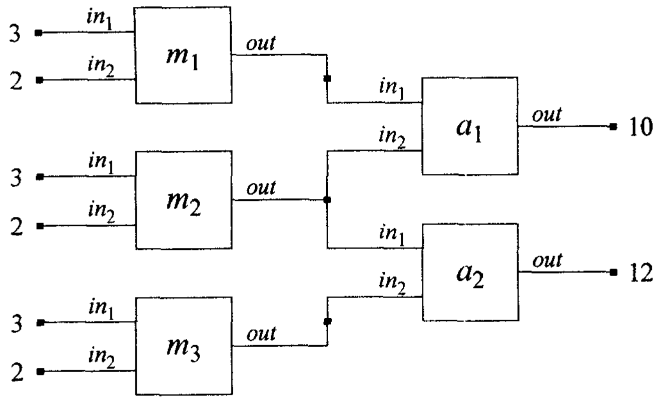

Fig. 1. A network with three multipliers and two adders.

output values as stated in Fig. 1 do not correspond to an intact network. One or several components must therefore be faulty. The problem is to locate and replace the faulty component(s).

The key point of model-based diagnostics is to formulate a descriptive model of the entire system. The model will then be used to solve the diagnosis problem. First, the components of the system have to be modeled. Components like adders and multipliers can be described by specifications of the form

$$\big(\mathit{adder}(X) \wedge {\sim}\mathit{ab}(X)\big) \rightarrow \big[\mathit{out}(X) = \mathit{in}_1(X) + \mathit{in}_2(X)\big],$$
$$\big(\mathit{multiplier}(X) \wedge {\sim}\mathit{ab}(X)\big) \rightarrow \big[\mathit{out}(X) = \mathit{in}_1(X) * \mathit{in}_2(X)\big].$$

Note the presence of the *ab*-predicate. It denotes abnormal (faulty) behavior of a component. In the specifications above it is not explicitly stated how a faulty adder or multiplier behaves; this means that a faulty component may return any result, including the correct one. Other assumptions are possible, e.g., a faulty component may never give the correct result, or always returns the negative of its correct result, or just discards negative signs, etc.

The second part of the model is to describe the system topology. The network topology as shown in Fig. 1 is modeled by listing the components present in the network

$$\mathit{multiplier}(m_1),\ \mathit{multiplier}(m_2),\ \mathit{multiplier}(m_3),$$
$$\mathit{adder}(a_1),\ \mathit{adder}(a_2),$$

and by listing the connections between the input and output ports of the components, namely

$$\mathit{out}(m_1) = \mathit{in}_1(a_1),\ \mathit{out}(m_2) = \mathit{in}_2(a_1),$$
$$\mathit{out}(m_2) = \mathit{in}_1(a_2),\ \mathit{out}(m_3) = \mathit{in}_2(a_2).$$

Finally, the model is completed by adding the observed input and output values. For the present example we have

$$in_1(m_1) = 3,\ \ in_2(m_1) = 2,\ \ in_1(m_2) = 3,\ \ in_2(m_2) = 2,$$
$$in_1(m_3) = 3,\ \ in_2(m_3) = 2,\ \ out(a_1) = 10,\ \ out(a_2) = 12.$$

As mentioned before, it is obvious that the output values do not correspond to the values predicted from the given input values. The above model can now help to identify the faulty component(s) of the system.

In the following an informal analysis of the situation will be performed. First, suppose that only the system description is given, but no observations are available yet. The system contains five components $c_1 = m_1$, $c_2 = m_2$, $c_3 = m_3$, $c_4 = a_1$, $c_5 = a_2$. To each component c_i we associate a Boolean variable x_i that indicates whether the component is intact ($x_i = 1$) or faulty ($x_i = 0$). A Boolean vector $x = (x_1, x_2, \ldots, x_5)$ describes one of the 2^5 possible system states. Examples of such states are $x = (1, 1, 1, 1, 1)$ (all components are intact), $x = (0, 0, 0, 0, 0)$ (all components are faulty), or $x = (0, 1, 1, 1, 1)$ (component m_1 is faulty, all other components are intact).

Now suppose that the above observations are made. For such a particular situation many system states x become impossible and must be excluded. An example of such a state is $x = (1, 1, 1, 1, 1)$, since at least one component must be faulty. More generally, all states x for which $x_1 = x_2 = x_4 = 1$ are impossible, since it is not possible that m_1, m_2, and a_1 are all intact. In Reiter's framework of model-based diagnostics the set of components $\{m_1, m_2, a_1\}$ is called minimal conflict set. Another minimal conflict set for the given situation is $\{m_1, m_3, a_1, a_2\}$ and all states x for which $x_1 = x_3 = x_4 = x_5 = 1$ must be excluded as well.

Given the observations, the state space $\{0, 1\}^5$ decomposes into two disjoint sets, the set N_d of possible states and the set N_c of impossible states. Any state $x \in N_d$ is a possible explanation of the observations, i.e., a possible "diagnosis" of the system behavior. On the other hand, any state $x \in N_c$ is in conflict with the observations and is called a "conflict" or a "contradictory" state. One of the main problems of our approach is to identify and represent the state sets N_d and N_c. Note that the size of the state space grows exponentially with the number of components contained in the system. For a system with a moderately large number of components, it will therefore not be feasible to represent N_d and N_c by explicit lists of system states.

However, consider the propositional language over the propositions $ab(c_i)$. System states are interpretations of these propositions in the sense of propositional logic: under the interpretation x, the proposition $ab(c_i)$ evaluates to "true" if $x_i = 0$, and to "false" if $x_i = 1$. Each formula f of this language can then be evaluated in the usual way for a given system state. Any formula f defines the set $N(f)$ of all interpretations x for which the formula f evaluates to true under x. Therefore, sets like N_d and N_c can be represented efficiently by corresponding propositional formulas d and c with $N_d = N(d)$ and $N_c = N(c)$. For the above example we get

$$\begin{aligned} c = {} & \big({\sim}ab(m_1) \wedge {\sim}ab(m_2) \wedge {\sim}ab(a_1)\big) \\ & \vee \big({\sim}ab(m_1) \wedge {\sim}ab(m_3) \wedge {\sim}ab(a_1) \wedge {\sim}ab(a_2)\big), \end{aligned} \tag{5}$$

$$d = ab(m_1) \vee ab(a_1) \vee \big(ab(m_2) \wedge ab(m_3)\big) \vee \big(ab(m_2) \wedge ab(a_2)\big). \tag{6}$$

The key point in Reiter's framework of model-based diagnostics are the concepts of minimal conflict sets and diagnoses. Note, that minimal conflict sets and diagnoses correspond essentially to our idea of logical representations of possible and impossible states. For example, every conjunction in (6) is a diagnosis in Reiter's sense. The main question treated by Reiter's theory is how to find conflict sets and diagnoses. In this paper we use some of Reiter's results to find the logical representations c and d (see Section 4).

Knowing the diagnoses of a faulty system does not always allow us to locate precisely the faulty component(s). For example, in (6) there are four diagnoses, two of them consisting of only one faulty component and two consisting of two faulty components. If we assume high component reliabilities, then diagnoses stating that only one component is faulty are more likely. In our example the faulty component will therefore probably be either m_1 or a_1 although the other possibilities can not be excluded. Reiter's framework does not permit a further investigation in order to discriminate between m_1 and a_1.

In our approach of possible and impossible system states we can investigate such cases more precisely. If N_i denotes the set of all states x with $x_i = 0$, then N_i represents the event that c_i is faulty. Given the observations, elements $x \in N_i \cap N_d$ can then be seen as arguments in favor of the hypothesis that component c_i is faulty. Furthermore, if mutually independent prior probabilities p_i are given, stating that a component c_i is intact at any given time, then it is possible to evaluate hypotheses of the form "component c_i is faulty" numerically by computing the posterior probability of N_i given N_d. For example, if we assume $p_1 = p_2 = p_3 = 0.95$ for multipliers and $p_4 = p_5 = 0.98$ for adders, then each system state x gets a well defined prior probability given by (1).

Knowing that the actual system state is in N_d changes the initial probability to the conditional probability according to (2). Now, we may be interested in the event that a particular component c_i is faulty. Then, according to (2) we get the following results for the present example (remember that N_i represents the event that component c_i is faulty):

$$p'(N_1) = 0.69, \quad p'(N_2) = 0.09, \quad p'(N_3) = 0.08, \quad p'(N_4) = 0.27, \quad p'(N_5) = 0.03.$$

The given situation can now be judged more precisely: component c_1 (i.e., multiplier m_1) has by far the highest failure probability and should therefore be replaced first. If after replacing component m_1 the system is still working abnormally, then $\sim ab(m_1)$ must be added to the system description (since the new component m_1 can be assumed to be intact), and the model has to be reevaluated.

In case the above method does not help to discriminate between two or more components, it may be possible to use the same analysis to select the most informative additional measurements and to redo the computations according to the new observations. This is not discussed in this paper.

4. The formal model

In this section we follow as far as possible Reiter's [31] framework and use also concepts and results from de Kleer et al. [11]. We adapt the formalism slightly in order get a direct way to introduce a probabilistic structure into the model in the next section.

4.1. Basic definitions

It is assumed that the system and also the observations about the system behavior are described in some language $\mathcal{L}$ which contains in particular the unary predicate $ab(c)$. Another requirement of the language and the logic behind it will be specified below. Otherwise, the language $\mathcal{L}$ can be very general as long as it has a clear and sound semantics. Note that the language of first-order logic, for example, satisfies all of our needs, but other languages are possible as well.

The system is supposed to be composed of a set $C = \{c_1, \dots, c_n\}$ of components and the predicate $ab(c)$ holds if the component $c \in C$ is faulty (i.e., c behaves abnormally). It is not excluded that several components of the system are working abnormally. Let x denote a Boolean vector $(x_1, \dots, x_n)$ whose component x_i describes whether component c_i is intact $(x_i = 1)$ or faulty $(x_i = 0)$. Such a vector x is called a **system state**. Define $ab(c)^e$ as the literal $ab(c)$ if $e = 0$ and $\sim ab(c)$ if $e = 1$. Then a state x can also be logically represented by the conjunction $c(x) = \bigwedge\{ab(c_i)^{x_i}\colon i = 1, \dots, n\}$.

Let now $\mathcal{S}$ denote the propositional language generated by the propositions $ab(c_1), \dots, ab(c_n)$ and assume that $\mathcal{S} \subseteq \mathcal{L}$. Formulas in $\mathcal{S}$ are statements representing a subset of all system states. The set of all system states can be seen as a set of interpretations for the formulas in $\mathcal{S}$ ($ab(c_i)$ evaluates to true under x if $x_i = 0$). If a formula $f \in \mathcal{S}$ evaluates to true under x we write $x \models f$. If $x \models f_1, x \models f_2, \dots, x \models f_m$ we write $x \models f_1, \dots, f_m$. As before, we define $N(f) = \{x\colon x \models f\}$, a subset of the state space $B_n = \{0, 1\}^n$. Note that the following properties hold for the sets $N(f)$:

$$\begin{aligned} &N(f \wedge g) = N(f) \cap N(g), && N(f \vee g) = N(f) \cup N(g), \\ &N(\sim f) = N(f)^c, && N(f) \subseteq N(g) \text{ iff } f \models g. \end{aligned} \tag{7}$$

A conjunction co of literals of $ab(c_i)^e$ is called an **implicant** of a formula $f \in \mathcal{S}$ if $co \models f$; it is called a **prime implicant** if no proper subconjunction of co is an implicant of f. If $d_1, \dots, d_m$ are the prime implicants of f, then it is well known that $d_1 \vee \dots \vee d_m$ is logically equivalent to f. We denote logical equivalence by the equality sign "=". Thus $f = d_1 \vee \dots \vee d_m$ and the right-hand side is called the disjunctive normal form (DNF) of f.[1] A clause cl (that is a disjunction of literals of $ab(c_i)$) is called an **implicate** of f if $f \models cl$; it is called a **prime implicate** if no proper subclause of cl is an implicate of f. If $g_1, \dots, g_r$ are the prime implicates of f, then $f = g_1 \wedge \dots \wedge g_r$ and the right-hand side is called the conjunctive normal form (CNF) of f.[2] These normal forms are convenient representations of propositional formulas.

Now we need to include in the formalism the fact that if some components are assumed to be intact and the others faulty then the system is in contradiction with the observations. So let $X \in \mathcal{L}$ be a set of sentences or formulas of the language $\mathcal{L}$ representing the behavior of the system and the observations. Note that the sentences in X normally contain one or several predicates $ab(c_i)$ because the behavior of the component c_i depends on its state

[1] In general, any disjunction of conjunctions which is equivalent to f is called a DNF; in contrast here we understand by a DNF of f only a disjunctive normal form whose conjunctions are prime implicants. Note that with this definition there are in general more than one DNF of f.

[2] Footnote 1 applies, respectively, also to CNFs and prime implicates.

(intact or faulty). Given a system state x—which is equivalent to specify what are the intact and faulty components—we can evaluate to true or false all occurrences of the predicate $ab(c_i)$ appearing in formula f in X. In this case we say that $f \in X$ is instantiated by x and we write f_x.

Given a system state $x \in B_n$, we suppose that there is a mechanism that permits to decide if the set of instantiated formulas $\{f_x\colon\ f \in X\}$ is consistent or not. We refer to Section 3 for an illustration.

If it is inconsistent, then the system state x is in conflict with the available information represented by X and we write $X, c(x) \Rightarrow \perp$. So it becomes possible to determine the system states that are in conflict with X and those that are compatible with it. If x is compatible with X, i.e., x is not in contradiction with X, we write $X, c(x) \not\Rightarrow \perp$. More generally, if $f_1, \ldots, f_s$ are formulas in the propositional language $\mathcal{S}$, then we write $X, f_1, \ldots, f_s \Rightarrow \perp$ if $X, c(x) \Rightarrow \perp$ for every x such that $x \models f_1, \ldots, f_s$. Note that if $f_1, \ldots, f_s$ are not satisfiable (in the sense of propositional logic) then trivially $X, f_1, \ldots, f_s \Rightarrow \perp$. For obvious reasons, if $X, f_1, \ldots, f_s, {\sim}h \Rightarrow \perp$ for some propositional formulas $f_1, \ldots, f_s, h \in \mathcal{S}$, then we write $X, f_1, \ldots, f_s \Rightarrow h$.

Lemma 1. *For $f, h \in \mathcal{S}$*

$$X, f \Rightarrow h \quad \textit{if and only if} \quad X, c(x) \Rightarrow h \textit{ whenever } x \models f.$$

(The proofs of the theorems and lemmas can be found in Appendix B.)

Given a system, let $X_S \subseteq \mathcal{L}$ be a set of sentences describing the behavior of the system through the relations among its components and let $X_O \subseteq \mathcal{L}$ be a set of sentences representing the observations. Together this gives $X_{SO} = X_S \cup X_O$ which is the available information. In general, there is another set of formulas $\Sigma \subseteq \mathcal{S}$ representing some knowledge about the possible system states. For example, in systems with components having different failure modes, there are formulas stating that each component is either operating correctly or is in exactly one of its failure modes. Then the set of formulas $X_{sys} = X_{SO} \cup \Sigma$ in $\mathcal{L}$ represents all available information about the system.

4.2. Conflict

The set $N_c = \{x \in B_n\colon\ X_{sys}, c(x) \Rightarrow \perp\}$ is called the set of **conflict states**. On the other hand $N_d = \{x \in B_n\colon\ X_{sys}, c(x) \not\Rightarrow \perp\} = N_c^c$ is the set of possible states or **diagnoses** (using the terminology of de Kleer et al. [12]). The formula

$$cont = \bigvee_{x \in N_c} c(x) \tag{8}$$

is called **conflict** or **contradiction**. More precisely, any formula in $\mathcal{S}$ that is logically equivalent to $cont$ is called the conflict, for example, its DNFs $dc_1 \vee \cdots \vee dc_m$ where $dc_1, \ldots, dc_m$ are all prime implicants of $cont$. A conjunction co of literals of $ab(c_i)$ is called a **conflict set** (or a contradiction set) if $X_{sys}, co \Rightarrow \perp$. The following lemma is inspired by [12] and shows that conflict sets and implicants of $cont$ are the same.

Lemma 2. *A conjunction co is a conflict set if and only if co is an implicant of $cont$.*

This lemma implies that the prime implicants dc_j of $cont$ are the minimal conflict sets. Since $X_{sys}, dc_j \Rightarrow \bot$, which is the same as $X_{sys} \Rightarrow \sim dc_j$, the clauses $\sim dc_j$ would be called "prime implicates" of X_{sys}. Indeed, if $\mathcal{L}$ was a propositional language, the clauses $\sim dc_j$ would be real prime implicates.

More generally, a formula $f \in \mathcal{S}$ is called a **conflicting argument** or a **contradictory argument** if

$$X_{sys}, f \Rightarrow \bot.$$

Lemma 3. *The formula cont is the weakest contradictory argument, i.e.:*
(1) $X_{sys}, cont \Rightarrow \bot$.
(2) *If f is a contradictory argument, then $f \models cont$.*

Obviously, this lemma implies that $cont$ is the only weakest contradictory argument (up to logical equivalence).

The conflict, the conflict sets and in particular the minimal ones, play a central role in the theories of diagnostics from first principles, as has been clearly shown by Reiter [31] and de Kleer et al. [12] and many other authors. Note that the notions of conflict and conflict sets are defined differently in [12,31]. Our definition of a conflict set as a conjunction of $ab(c_i)$-literals corresponds to Reiter's [31] notion of a conflict set. In contrast, de Kleer et al. [12] use clauses of $ab(c_i)$-literals for the same purpose and call them conflicts. In our paper, a conflict is a logical description of the conflict states N_c.

In fact it is almost always proposed to first compute the minimal conflict sets from the system description and the observations and then to derive from them all kinds of diagnostic concepts. For example, from the minimal conflict sets $dc_1, \ldots, dc_m$ the DNF

$$cont = dc_1 \vee \cdots \vee dc_m$$

is considered and the prime implicants of $\sim cont$ are called kernel diagnoses by de Kleer et al. [12].

The method for computing the conflict sets depends on the particular language or logic used to describe the system and the observations. If, for example, the system and the observations are described in propositional logic, then resolution methods can be applied to obtain all prime implicates of X_{sys} (not only those in $\mathcal{S}$). Then using results of Reiter and de Kleer [32] the minimal conflicts can be extracted from them. More efficient methods based on the concept of production fields are possible [19,22,35]. These approaches apply for example to the diagnostics of digital circuits [3].

Often systems are built from components which can be described by input-output relations between variables. Observations are then usually obtained by measuring the values of some variables. In such cases assumption-based constraint propagation can be used to obtain the conflict sets [13,15]. This approach has been implemented in ABEL[3] [2,4], which is a general modeling language for assumption-based reasoning under uncertainty.

[3] The source code of an implementation of ABEL is freely available from `http://www2-iiuf.unifr.ch/tcs/abel/`.

4.3. Quasi-support

The conflict sets are also fundamental for the theory to be developed here. In the first place they provide a convenient representation of the set of conflict states N_c and thereby, implicitly, also for its complement, the set of diagnoses N_d.

But there is more. A formula $h \in \mathcal{S}$ can be considered as a hypothesis about the correct but unknown system state. Given X_{sys} it is then interesting to ask whether there are arguments supporting h or arguments rejecting h. An **argument** supporting h is an assumption about the state of certain components, represented by a formula $f \in \mathcal{S}$ such that

$$X_{sys}, f \Rightarrow h. \tag{9}$$

For $h \in \mathcal{S}$ let

$$Q(h) = \{x \in B_n\colon\ X_{sys}, c(x) \Rightarrow h\} \tag{10}$$

and define the **quasi-support** of h as

$$qs(h) = \bigvee_{x \in Q(h)} c(x). \tag{11}$$

Lemma 4. *For every $h \in \mathcal{S}$, $qs(h)$ is the weakest argument supporting h, i.e.:*
(1) $X_{sys}, qs(h) \Rightarrow h$,
(2) *if f is a supporting argument for h, then $f \models qs(h)$.*

It follows from this lemma that there is only one weakest argument supporting a hypothesis h (up to logical equivalence).

By definition, a contradictory argument is an argument supporting $\bot$. By Lemma 3, *cont* is the weakest contradictory argument and hence also the weakest argument supporting $\bot$. But then by Lemma 4 it follows that

$$cont = qs(\bot) \tag{12}$$

which in turn implies that

$$N(qs(\bot)) = N(cont) = N_c. \tag{13}$$

More generally, we have the following result:

Lemma 5. *For every $h \in \mathcal{S}$:*
(1) $N(qs(h)) = Q(h)$,
(2) $qs(\bot) \models qs(h)$.

The quasi-support of h is a support of h because it is an argument of h but not a proper support of h since $qs(\bot) \models qs(h)$ and $qs(\bot)$ is a contradictory argument. This is why $qs(h)$ is only a quasi-support of h. Proper supports will be considered below.

The following theorem is the basic result of the theory developed in this paper.

Theorem 6. *For every $h \in \mathcal{S}$:*

$$N\big(qs(h)\big) = N(h) \cup N_c. \tag{14}$$

It has the following important corollary:

Corollary 7. *For every $h \in \mathcal{S}$:*

$$qs(h) = h \vee qs(\bot). \tag{15}$$

Since $qs(\bot) = cont$ and $cont$ is determined by the minimal conflict sets, Corollary 7 implies that $qs(h)$ is also determined by the minimal conflict sets. This underlines once more the importance of the conflict sets for the theory of diagnostics from first principles. The full importance of (15) will become clear in the following Section 5 where probability calculations are made.

Once $qs(\bot)$ is given, for example, by the disjunction of all minimal conflict sets, the prime implicants $dh_1, \ldots, dh_r$ of $qs(h)$ may be computed by standard methods of propositional logic. This determines then the DNF $qs(h) = dh_1 \vee \cdots \vee dh_r$. The conjunctions dh_j are called **minimal quasi-support sets** of h. They provide a convenient representation of $qs(h)$. Note that the minimal quasi-support sets of $\bot$ are the minimal conflict sets.

Example 2. Let is illustrate these notions and results by continuing the example of the arithmetical network of Fig. 1 discussed in Section 3. As mentioned before there are two minimal conflict sets and therefore we have

$$\begin{aligned} qs(\bot) = {} & \big({\sim}ab(m_1) \wedge {\sim}ab(m_2) \wedge {\sim}ab(a_1)\big) \\ & \vee \big({\sim}ab(m_1) \wedge {\sim}ab(m_3) \wedge {\sim}ab(a_1) \wedge {\sim}ab(a_2)\big). \end{aligned}$$

Suppose now that we are interested in the hypothesis that m_1 is intact, i.e., $h = {\sim}ab(m_1)$. Then, by Corollary 7, $qs({\sim}ab(m_1)) = {\sim}ab(m_1) \vee qs(\bot) = {\sim}ab(m_1)$. On the other hand, for example, $qs({\sim}ab(m_2)) = {\sim}ab(m_2) \vee ({\sim}ab(m_1) \wedge {\sim}ab(a_1) \wedge {\sim}ab(m_3) \wedge {\sim}ab(a_2))$. Section 5 will show how these logical expressions can be used to compute posterior probabilities for the corresponding hypotheses.

The following theorem gives further remarkable properties of quasi-support.

Theorem 8. *For any $h_1, h_2 \in \mathcal{S}$:*

$$qs(h_1 \wedge h_2) = qs(h_1) \wedge qs(h_2), \tag{16}$$

$$qs(h_1 \vee h_2) = qs(h_1) \vee qs(h_2). \tag{17}$$

If $\top$ denotes the tautology, then

$$qs(\top) = \top. \tag{18}$$

This result links this theory of diagnostics to a more general theory. In a more general setting where h belongs to a first propositional language $\mathcal{H}$ (the language of **hypotheses**) and the image $qs(h)$ to another, second propositional language $\mathcal{A}$ (the language of **assumptions**), and for which (16) and (18) (but not necessarily (17)) are satisfied, $qs(h)$ is called an **allocation of support** [20,21]. If probabilities are given to the assumptions, then the probabilities of the formulas $qs(h)$ define a belief (or support) function on the set of hypotheses h in $\mathcal{H}$ in the sense of Dempster–Shafer theory of evidence [20,21,33]. This is an approach to the Dempster–Shafer theory that goes along similar lines as the one proposed by Provan [28].

A more explicit discussion of the relations and links between model-based diagnosis and the Dempster–Shafer theory of evidence can be found in [3,24].

Theorem 8 implies that $qs(h \vee h^c) = qs(h) \vee qs(h^c)$ and this shows that qs is a so-called additive allocation of support. For additive allocations of support it can be shown that the induced belief functions are Bayesian [20,33]. Since Bayesian belief functions are mathematically equivalent to ordinary probability measures, the results given in Section 5 are expressed in probabilistic terms.

4.4. Support

A formula $f \in \mathcal{S}$ is called a **proper argument** of h if f is an argument for h but f is not a contradictory argument, i.e., if

(1) $X_{sys}, f \Rightarrow h$,

(2) $X_{sys}, f \not\Rightarrow \bot$.

The following lemma implies that the second condition saying that f is not a contradictory argument is the same as saying that f does not entail the conflict *cont*.

Lemma 9.

$$f \models cont \quad \textit{if and only if} \quad X_{sys}, f \Rightarrow \bot. \tag{19}$$

For $h \in \mathcal{S}$, define

$$s(h) = qs(h) \wedge \sim qs(\bot). \tag{20}$$

The following lemma says that $s(h)$ is a proper argument of h.

Lemma 10. *If $s(h) \neq \bot$ then $s(h)$ is a proper argument of h.*

Corollary 7 implies the following result:

Corollary 11. *For every $f \in \mathcal{S}$:*

$$s(h) = h \wedge \sim qs(\bot). \tag{21}$$

This result in turn implies that $N(s(h)) = N(h) \cap N_a$. This means that $s(h)$ represents the set of all possible system states (i.e., all diagnoses) which prove the hypothesis h, i.e., all possible system states that logically entail h. This is the reason why the proper argument $s(h)$ is called the **support** of h.

Once again, as shown in Eq. (21), the support can be easily determined from the conflict $qs(\bot)$. Also, note that $s(\bot) = \bot$.

At this point, we are ready to see how our theory relates with the ideas and results presented in Darwiche [8]. He considers a system which is a tuple (Δ, P, A, ϕ) where Δ is a database containing formulas in the language $\mathcal{L}_{A+P}$ generated by the atoms in $P \cup A$ and ϕ is a formula in the language $\mathcal{L}_P$ generated by the atoms in P. He calls the atoms in A assumables and those in P non-assumables. The interpretation of these items is that the database describes the system's behavior, the assumables represent the mode of the system components (working or not) and the formula ϕ represents the observed system behavior. Then he explains what is a diagnosis of a system and defines what is the consequence of an observation ϕ. If $\mathcal{L}_A$ denotes the language generated by the atoms in A, then the consequence of an observation $\phi \in \mathcal{L}_P$, written $Cons(\phi)$, is the strongest formula $\alpha \in \mathcal{L}_A$ such that $\Delta \cup \{\phi\} \models \alpha$. This allows him to characterize all diagnoses in terms of $Cons(\phi)$. For two observations ϕ_1 and ϕ_2 in $\mathcal{L}_P$, it is mentioned in the paper that

$$Cons(\phi_1 \vee \phi_2) = Cons(\phi_1) \vee Cons(\phi_2). \tag{22}$$

He also defines the argument for ϕ, written $Arg(\phi)$, which is defined as the weakest formula $\alpha \in \mathcal{L}_A$ such that $\Delta \cup \{\alpha\} \models \phi$, and mentions that

$$Arg(\phi) = {\sim}Cons({\sim}\phi). \tag{23}$$

In Darwiche's paper, the goal of the diagnosis analysis is then to identify a so-called most preferred diagnosis. To reach this goal, the paper gives an algorithm for the computation of $Cons(\phi)$.

How does this relate with our theory? First of all it is important to notice that his definition of a diagnosis corresponds exactly to our definition. Moreover, his concept of a system can be placed into our theory. Indeed, we just have to set $\mathcal{S} := \mathcal{L}_A$, $\mathcal{L} := \mathcal{L}_{A+P}$, $X_S := \Delta$ and $X_O := \{\phi\}$. However, our model is more general than the one of Darwiche [8] because our description of the system behavior and the observation must not necessarily be expressed in propositional logic. For example, in the arithmetical network of Fig. 1, it would be hard to describe the observations in propositional logic, which must be the case in the system of Darwiche.

Now suppose that the system (Δ, P, A, ϕ) is placed into our framework as described above. In this paper, we have defined the function

$$qs : \mathcal{L}_A \to \mathcal{L}_A. \tag{24}$$

Since this function depends on the system description Δ and the observations ϕ, it could be written $qs_{\Delta \cup \{\phi\}}$ instead of just qs.

On the other hand, Darwiche defines the function

$$Cons : \mathcal{L}_P \to \mathcal{L}_A. \tag{25}$$

So, in spite of their similarity, the properties (17) and (22) are completely different because (17) is a property of hypotheses $h \in \mathcal{L}_A$ and (22) is a property of observations $\phi \in \mathcal{L}_P$. The same remark holds true with respect to the property (16) and the property

$$Arg(\phi_1 \wedge \phi_2) = Arg(\phi_1) \wedge Arg(\phi_2) \tag{26}$$

for all ϕ_1, ϕ_2 in $\mathcal{L}_P$. To the best of our knowledge, Theorem 8 is a new result.

However, we have the following interesting relations:

$$Cons(\phi) = {\sim}qs_{\Delta\cup\{\phi\}}(\perp), \tag{27}$$

$$Arg(\phi) = qs_{\Delta\cup\{\sim\phi\}}(\perp). \tag{28}$$

In the next section it is explained how the conflict $qs_{\Delta\cup\{\phi\}}(\perp)$ can be used to compute the posterior probability of some hypothesis of interest h in $\mathcal{L}_A$. The development assumes that the conflict has already been determined by some method (see above). In view of Eq. (27), the algorithm presented in the article of Darwiche to compute $Cons(\phi)$ can be used to find the conflict:

$$qs_{\Delta\cup\{\phi\}}(\perp) = {\sim}Cons(\phi). \tag{29}$$

This shows that Darwiche's paper can be useful for our theory, but the goals of the two papers are different: he wants to identify the most preferred diagnosis whereas we want to evaluate interesting hypotheses about the system.

Example 3. Once again we use the arithmetical network of Fig. 1 to illustrate the concept of support. According to Section 3 we have

$${\sim}qs(\perp) = ab(m_1) \vee ab(a_1) \vee \big(ab(m_2) \wedge ab(m_3)\big) \vee \big(ab(m_2) \wedge ab(a_2)\big). \tag{30}$$

Then from Corollary 11 we obtain

$$\begin{aligned}
&s\big(ab(m_1)\big) = ab(m_1),\\
&s\big(ab(m_2)\big) = ab(m_2) \wedge \big(ab(m_1) \vee ab(a_1) \vee ab(m_3) \vee ab(a_2)\big),\\
&s\big(ab(m_3)\big) = ab(m_3) \wedge \big(ab(m_1) \vee ab(a_1) \vee ab(m_2)\big),\\
&s\big(ab(a_1)\big) = ab(a_1),\\
&s\big(ab(a_2)\big) = ab(a_2) \wedge \big(ab(m_1) \vee ab(a_1) \vee ab(m_2)\big).
\end{aligned}$$

5. Numerical evaluation of hypotheses

Once the probabilities of the system states are defined, the probability of any hypothesis $h \in \mathcal{S}$ about the system one might want to consider can be computed. The model starts with prior probabilities on the possible system states before any observation is made. Observations on the system reduces then in general the set of possible states. The prior probabilities must correspondingly be conditioned on the new set of possible states to obtain posterior probabilities as has already been exemplified in the introductory example of Section 3.

This probabilistic framework is very simple. The real problem is the computation of these conditional probabilities for various hypotheses. The computational problem is similar to the problems encountered in combinatorial reliability theory. Appropriate methods may be borrowed from this field and adapted to the present situation.

5.1. Components with one failure mode

In the simplest case the system is composed of independently failing components where each component can only be in one of two different states: functioning or faulty. The ith component x_i of the system state x indicates in this case simply whether the component is intact ($x_i = 1$) or faulty ($x_i = 0$). The probability model is then based on the sample space B_n, i.e., the state space of the system, and the probability on this space is defined by the probabilities $p(x)$ of the sample points or states. If p_i is the (prior) probability that the ith component is intact (at a given point in time), and if the failures of components are assumed to be stochastically independent, then

$$p(x) = \prod_{i=1}^{n} p_i^{x_i}(1-p_i)^{(1-x_i)}. \tag{31}$$

The probability of any event $A \subseteq B_n$ can in principle be obtained as the sum of these elementary probabilities over the sample points which make the event,

$$p(A) = \sum_{x \in A} p(x). \tag{32}$$

For any hypothesis $h \in \mathcal{S}$ its prior probability $p(N(h))$ can now be calculated.

Now, if observations on the system restrict the possible states to the set $N_d \subseteq B_n$ of diagnoses, then this means that the prior probabilities $p(x)$ must be conditioned on the event N_d to get the posterior probabilities

$$p'(x) = \frac{p(x)}{p(N_d)}, \quad \text{with } p(N_d) = \sum_{x \in N_d} p(x). \tag{33}$$

The posterior probability of an event $A \subseteq B_n$ is then given by

$$p'(A) = \frac{p(A \cap N_d)}{p(N_d)}. \tag{34}$$

Given the observations, the posterior probability of the hypothesis h is then $p'(N(h))$. In order to simplify notation we write $p'(h)$ instead of $p'(N(h))$ for any formula $h \in \mathcal{S}$. Thus for all $h \in \mathcal{S}$:

$$p'(h) = \frac{p(N(h) \cap N_d)}{p(N_d)} = \sum_{x \in N(h) \cap N_d} p'(x). \tag{35}$$

The posterior probability $p'(h)$ expresses the chance that hypothesis h is true. It can easily be proved that $s : \mathcal{S} \to \mathcal{S}$ given by $s(h) = qs(h) \wedge {\sim}qs(\perp)$ is an additive allocation of support because so is the function $qs : \mathcal{S} \to \mathcal{S}$ (see Section 4). Since the function

$$p''(x) = \begin{cases} p(x)/p(N_d) & \text{if } x \in N_d, \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

is a probability measure on B_n, it follows from the general theory of allocations of support [20] that the induced belief (or support) function $sp : \mathcal{S} \to [0, 1]$ given by $sp(h) = p''(N(s(h))$ is a Bayesian belief function, i.e., a probability measure on $\mathcal{S}$. The following theorem shows that the degree of belief, or more in the spirit of this paper, the **degree of support** of a hypothesis h expressed by $sp(h)$ is the same as the posterior probability of h given by $p'(h)$.

Theorem 12. *For every hypothesis $h \in \mathcal{S}$:*

$$sp(h) = p'(h). \tag{37}$$

This means that the chance or likelihood of h being true, namely $p'(h)$, can be viewed as the strength with which the support $s(h)$ actually supports h. This shows that the posterior probabilities of hypothesis are nothing but weights of arguments when the analysis is placed in the context of Dempster–Shafer theory of evidence. This is a fairly convincing way to judge and evaluate hypotheses.

The real problem is the computation of sums as in (33) and (35) because of the large number of terms these sums may contain. In fact, if the system consists of n components, then the expected number of terms in those sums is of the order of 2^n. The computation of such sums becomes quickly infeasible. Therefore, better methods must be found for the computation of the posterior probabilities $p'(h)$. This problem will be addressed in the next subsection.

5.2. Computation of posterior probabilities

First note that there is an alternative formula for a probability like $p'(h)$:

$$\begin{aligned} p'(h) &= \frac{p(N(h) \cap N_d)}{p(N_d)} = \frac{p(N(h) \cup N_c) - p(N_c)}{1 - p(N_c)} \\ &= \frac{p(qs(h)) - p(qs(\bot))}{1 - p(qs(\bot))}. \end{aligned} \tag{38}$$

This last expression is convenient for computational purposes. We consider first the computation of $p(qs(\bot))$.

It is supposed that $qs(\bot)$ is given in the disjunctive normal form $qs(\bot) = dc_1 \vee \cdots \vee dc_m$ where the dc_j are the minimal conflict sets as explained in Section 4. Since $p(\bigvee dc_j) = p(\bigcup N(dc_j))$ it follows that $p(qs(\bot))$ is in fact the probability of a union of events (or the probability of a disjunction of formulas, if we use the logical language, as we will do in the sequel). This is a classical problem of probability theory which has been much studied, especially in combinatorial reliability theory.

A first and simple approach is given by the so-called **inclusion-exclusion formula** [14]:

$$p\big(qs(\bot)\big) = p(dc_1 \vee \cdots \vee dc_m) = \sum_{\emptyset \neq I \subseteq \{1,\ldots,m\}} (-1)^{|I|+1} p\Big(\bigwedge_{i \in I} dc_i\Big). \tag{39}$$

Note that $\bigwedge_{i \in I} dc_i$ is still a conjunction. The probabilities of conjunctions can easily be calculated when components are assumed to fail independently. In fact, if $\bigwedge_{i \in I} dc_i = \bigwedge_{k \in K} ab(c_k) \bigwedge_{h \in H} ({\sim}ab(c_h))$, then

$$p\Big(\bigwedge_{i \in I} dc_i\Big) = \prod_{k \in K} (1 - p_k) \prod_{h \in H} p_h. \tag{40}$$

Since the number of terms in the sum (39) grows exponentially with the number of minimal conflict sets m, the computational effort needed for the sum can quickly become prohibitive.

An alternative method, which is often superior to the inclusion-exclusion method, consists in transforming the disjunctive form $qs(\bot) = dc_1 \vee \cdots \vee dc_m$ into a disjoint form

$$qs(\bot) = f_1 + \cdots + f_r, \quad \text{with } f_i f_j = \bot \text{ whenever } i \neq j. \tag{41}$$

Here we denote the disjunction of disjoint terms by a $+$ and the conjunctions of terms by a product. Disjoint terms f_j correspond to disjoint sets $N(f_j)$ and therefore

$$p\big(qs(\bot)\big) = \sum_{j=1}^{r} p(f_j). \tag{42}$$

The number of terms in such a sum is often much smaller than the number of terms in (39).

Now arises however the problem of finding such a disjoint representation of $qs(\bot)$ from its DNF $dc_1 \vee \cdots \vee dc_m$. In addition, the disjoint form must be such that the probabilities of its terms are easy to compute. Several methods for this problem have been developed in reliability theory, however for so-called monotone Boolean functions only [6]. This monotonicity property does not hold in general in the present context. Fortunately, however, at least two of the best methods can be adapted to the present more general case.

The first one is due to Abraham [1]. See Chapter 5 of [23] for its adaptation to our situation. In this method $qs(\bot)$ is developed into disjoint **conjunctions** f_j of literals of $ab(c_i)$. So the probabilities $p(f_j)$ are easily computed by (40), but this method still tends to yield a relatively large number of terms. Therefore, Heidtmann [18] proposed a different, although more complex method which, in most cases, gives less terms. In his method, a f_j is a conjunction of one conjunction of literals and a certain number of negations of conjunctions of literals. In addition, the factors in f_j are independent. The probability of a factor in f_j is easily calculated by (40) (for the negations, the complement to 1 of the probability of the conjunction has to be taken). Then the probability of f_j is simply the product of these factor probabilities. For an adaptation of Heidtmann's method to the present more general case see [7].

Once the (prior) probability of the conflict $p(qs(\bot))$ is computed, Formula (38) can be applied to obtain $p'(h)$ for a hypothesis $h \in \mathcal{S}$. For this purpose $p(qs(h)) = p(h \vee qs(\bot))$ has to be computed. Sometimes $qs(h)$ is already a very simple formula and it is easy to obtain its probability. Otherwise, if h is a simple conjunction of literals of $ab(c_i)$, in particular if $h = ab(c_i)$ or $h = {\sim}ab(c_i)$, $p(qs(h))$ can be obtained by computing $p(h)$ and using the formula $p(h \vee qs(\bot)) = p(h) + p(qs(\bot)) - p(h \wedge qs(\bot))$. If $qs(\bot)$ is already in disjoint form $f_1 + \cdots + f_r$, then $h \wedge qs(\bot) = hf_1 + \cdots + hf_r$ is still a disjoint form, and if the f_i are conjunctions of literals of $ab(c_i)$, then so are the hf_i. So $p(h \wedge qs(\bot))$ can be computed from this disjoint form. In the general case, h has first to be represented as a DNF $h = h_1 \vee \cdots \vee h_t$ and then $qs(h) = h \vee qs(\bot) = (h_1 \vee \cdots \vee h_t) \vee (f_1 + \cdots + f_r)$ has to be further developed into a disjoint form by the generalized methods of Abraham or Heidtmann.

Example 4. In the arithmetical network of Fig. 1 Abraham's method applied to *cont* as given in disjunctive form in Section 4 yields

$$\begin{aligned} qs(\bot) = {} & \big({\sim}ab(m_1) \wedge {\sim}ab(m_2) \wedge {\sim}ab(a_1)\big) \\ & + \big({\sim}ab(m_1) \wedge {\sim}ab(a_1) \wedge {\sim}ab(m_3) \wedge {\sim}ab(a_2) \wedge ab(m_2)\big). \end{aligned}$$

Suppose now that a priori multipliers are working correctly with probability 0.95 and adders are working correctly with probability 0.98. Then it follows that

$$p\big(qs(\bot)\big) = 0.95^2 \cdot 0.98 + 0.95^2 \cdot 0.98^2 \cdot 0.05 = 0.928.$$

Let us now consider the hypothesis that the multiplier m_1 is working correctly, i.e., $h = {\sim}ab(m_1)$. We have seen in Section 4 that $qs({\sim}ab(m_1)) = {\sim}ab(m_1)$ and hence $p(qs({\sim}ab(m_1))) = 0.95$. It follows that

$$p'\big({\sim}ab(m_1)\big) = \frac{p({\sim}ab(m_1)) - p(qs(\bot))}{1 - p(qs(\bot))} = \frac{0.95 - 0.928}{1 - 0.928} = 0.31.$$

In the same way we can compute the posterior probabilities

$$\begin{aligned} p'\big({\sim}ab(m_2)\big) &= 0.91, \\ p'\big({\sim}ab(m_3)\big) &= 0.92, \\ p'\big({\sim}ab(a_1)\big) &= 0.73, \\ p'\big({\sim}ab(a_2)\big) &= 0.97. \end{aligned}$$

This is how the numerical results in Section 3 have been computed. Since p' is a probability measure, $p'({\sim}h) = 1 - p'(h)$ and hence the most probable candidate for a fault is multiplier m_1 according to these results.

Note that the probabilities of faulty components, the complements of the above five probabilities, add up to more than 1. The reason is that there can be more than one faulty component. In other words, the events $ab(m_1)$, $ab(m_2)$, $ab(m_3)$, $ab(a_1)$, and $ab(a_2)$ are not exclusive and hence the probability of conjunctions of such events can be positive. In fact, if for example it turns out that adder a_2 is faulty (although this is relatively unlikely according to the above probabilities), then this is not sufficient to explain the observations: there must be at least one other faulty component. So these five probabilities alone do not tell the whole story and other, composite hypotheses should be considered. But these probabilities do help to focus the search for faulty components to be replaced.

This gives then a method for computing posterior probabilities, given observations of the system behavior, in the simple model of independently failing components with two states (intact or faulty). As can be seen, the crucial point is the probability of the conflict. This is a further illustration of the central role this notion plays in model-based diagnostics. In the next section it is shown that the method can be adapted to a more general situation.

5.3. Components with several failure modes

Besides the simple model of independently failing components with two states introduced in Section 5.1, there is another important and interesting model to be considered in this section. As before it is assumed that the system is composed of components c_i, $i = 1, \ldots, n$. But now some components may also possess more than only one failure mode.

In order to include such a system into the formalism of Section 4, let the predicate $m_j(c_i)$ indicate that the component c_i is in mode j. Usually, $m_1(c_i)$ designates the normal mode of component c_i, whereas $m_j(c_i)$ for $2 \leqslant j \leqslant r_j$ will refer to its failure modes.

Before generalizing to several failure modes, we reconsider the situation with one failure mode from another perspective which is useful to understand the generalization. The state space considered there was B_n and the predicates $ab(c_i)$ and $\sim ab(c_i)$ were used to denote the two possible states of a component. An alternative way is to introduce predicates $m_1(c_i)$ and $m_2(c_i)$ to represent the same information ($m_1(c_i)$ means that c_i is intact, $m_2(c_i)$ that c_i is faulty and $r_i = 2$ for all i). Then we have to state explicitly that the component c_i is in exactly one of these two modes, i.e.,

$$m_1(c_i) \vee m_2(c_i). \qquad \sim\big(m_1(c_i) \wedge m_2(c_i)\big). \tag{43}$$

Instead of the state space B_n we must now consider the larger space B_{2n}. A vector $x = (x_{11}, x_{12}, x_{21}, x_{22}, \ldots, x_{n1}, x_{n2})$ in B_{2n} must be interpreted as follows: $x_{ij} = 1$ means that the component c_i is in mode j and $x_{ij} = 0$ means that it is not in mode j. Let A denote the set of all vectors $x \in B_{2n}$ such that there is at least one index $i \in \{1, \ldots, n\}$ with $x_{i1} = x_{i2}$. Because of (43), the vectors in A do not represent possible configurations. On the other hand, for a vector $x \in A^c$, let

$$I_1 = \big\{i \in \{1, \ldots, n\}\colon x_{i1} = 1\big\},$$
$$I_2 = \big\{i \in \{1, \ldots, n\}\colon x_{i2} = 1\big\}.$$

The components c_i with $i \in I_1$ are intact and those with $i \in I_2$ are faulty. Then we have to define the probability p on B_{2n} as follows:

$$p(x) = \begin{cases} 0 & \text{if } x \in A, \\ \prod_{i \in I_1} p_i \cdot \prod_{i \in I_2} (1 - p_i) & \text{if } x \in A^c, \end{cases} \tag{44}$$

where p_i is the probability that component c_i is intact. This definition puts the probability of impossible configurations to zero.

In the sequel we generalize these ideas to components with several failure modes. As in the case with two failure modes above, it will be assumed that every component c_i must be in exactly in one of its possible operating modes, a requirement which is expressed by the logical formulas

$$m_1(c_i) \vee \cdots \vee m_{r_i}(c_i) \quad \text{and} \quad \sim\big(m_j(c_i) \wedge m_k(c_i)\big) \tag{45}$$

for all $j \neq k$ and for all $i = 1, \ldots, n$. These formulas have to be included into X_{sys}. One can also eliminate the predicate $m_1(c_i)$ (or similarly any other one) and replace each occurrence of it in X_{sys} by the formula $\sim(m_2(c_i) \vee \cdots \vee m_{r_i}(c_i))$. Then the first disjunction in (45) is not needed, and the second conjunction in (45) has only to be stated for $j, k \geqslant 2$. This modeling trick may be useful because there are less predicates than before. An alternative way of dealing with mutually exclusive predicates is described in [5,17,26].

The logical theory developed in Section 4 also applies to this model. Note that among the minimal conflicts there may be the "trivial" conjunctions $m_j(c_i) \wedge m_k(c_i)$ from (45), unless one or both of $m_j(c_i)$ and $m_k(c_i)$ are already conflicts themselves.

We turn now to the probability structure of this model. The natural sample space to be considered here is constructed as follows. For each component c_i let

$$\Theta_i = \{m_1, \ldots, m_{r_i}\}$$

be the set of possible modes. A system state can then be identified by an element of the product space

$$\Theta = \prod_{i=1,\ldots,n} \Theta_i.$$

If p_{ij} denotes the prior probability that c_i is in mode j, then given that each component c_i must be in exactly one of its possible modes, we must have

$$\sum_{j=1}^{r_i} p_{ij} = 1 \tag{46}$$

for all $i = 1, \ldots, n$. Again stochastic independence is assumed between the modes of different components. Then the probability of a system state $x = (m_{j_1}, \ldots, m_{j_n}) \in \Theta$ is given by

$$p(x) = p_{1j_1} \cdots p_{nj_n}. \tag{47}$$

In order to link this probability model with the probability model of Section 4, we need another representation of the state space Θ. To do this we generalize the idea that was described at the beginning of this subsection. Suppose that a state space

$$x = (x_1, \ldots, x_n) \in \Theta$$

is given, i.e., $x_i \in \{m_1, \ldots, m_{r_i}\}$ for all $i = 1, \ldots, n$. Given the system state x we define a Boolean vector $y^{(i)}$ of length r_i for each $i = 1, \ldots, n$ as follows:

$$y_j^{(i)} = \begin{cases} 1 & \text{if } x_i = m_j, \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

Each $y^{(i)}$ is therefore composed of exactly one 1, all other components being 0. Then we define the vector $y = (y^{(1)}, \ldots, y^{(n)})$ of length $m = r_1 + \cdots + r_n$. There is a bijective correspondence between the vectors y and the possible system states x and therefore the probability of a vector y is the same as the probability of its corresponding system state x, i.e., $p(y) = p(x)$. The vectors y are elements of the Boolean cube B_m, which is the set of interpretations of the propositional language $\mathcal{S}$ generated by all the atoms $m_j(c_i)$. In the rest of the paper, $\mathcal{S}$ will always denote the propositional language generated by *all* the atoms $m_j(c_i)$. If we define the probability of a vector in B_m which does not correspond to a possible system state x to be zero, then we have a probability on the entire set B_m. This implies that the probability $p(f)$ of any formula $f \in \mathcal{S}$ is obtained by summing the probabilities of all interpretations satisfying f. By this construction the general formalism considered in the previous sections applies again and the results given there are still valid (in particular, the concepts of argument, quasi-support, and support are well defined). We are interested in computing the posterior probability $p'(h) = sp(h)$ for some hypothesis $h \in \mathcal{S}$ (p' and sp are defined in the same way as in Section 5.1). Since Eq. (38) is still valid, the basic problem is again the computation of $p(qs(\bot))$ and $p(qs(h))$.

The algorithms of Abraham and Heidtmann (in their generalized version presented in [23]) can still be applied to obtain disjoint decompositions of $qs(\bot)$ and $qs(h)$. Among the disjoint conjunctions generated by these algorithms, there might be some of the form

$$\cdots \wedge m_j(c_i) \wedge \cdots \wedge m_k(c_i) \wedge \cdots \quad \text{with } j \neq k. \tag{49}$$

Of course, such conjunctions have zero probability because they do not correspond to a possible system state. Furthermore, these algorithms may also generate conjunctions of the form

$$\cdots \wedge m_j(c_i) \wedge \cdots \wedge \sim m_k(c_i) \wedge \cdots \quad \text{with } j \neq k, \tag{50}$$

but the probability of such conjunctions is no longer simply given by the product of the probabilities of its literals. Although it is not difficult to compute the probability of a conjunctions of the form (50), it can be improved. A new decomposition algorithm presented in Appendix A avoids the generation of conjunctions of the form (49) and (50), thereby improving the efficiency of the computation of $p(qs(\bot))$ and $p(qs(h))$.

6. Conclusion

In this paper model-based diagnostics is looked at from a perspective of argumentation. This means that the actual observations about the system along with the model itself serve as material to generate arguments for hypotheses about the state of some components. To develop the whole theory, it is only needed to know when a system state is in conflict with the knowledge represented by the model and the observations. As a consequence, the language used to describe the available knowledge can be very general. The search for arguments leads to a qualitative or symbolic diagnostic analysis because the support of a hypothesis is explicitly represented by a logical formula. Theorem 8 shows that these ideas can be nicely expressed in the framework of the Dempster–Shafer theory of evidence and assumption-based reasoning.

On top of this qualitative analysis, it is possible to introduce a quantitative or numerical aspect. The importance of the arguments for a hypothesis is measured by the degree of support of the hypothesis. A degree of support of a hypothesis represents the strength with which the available information tends to prove the hypothesis. It corresponds to the idea of probability of provability as introduced by Pearl [27]. It turns out that the obtained support functions—called belief functions in the Dempster–Shafer theory of evidence—are indeed Bayesian, i.e., they are regular probability measures in the model considered here. This means that we can also look at degrees of support as conditional probabilities. Corollary 7 gives a simple representation of the quasi-support and this result is used to design an algorithm to compute degrees of support more efficiently.

Acknowledgement

We thank the anonymous referees for the interesting and useful comments.

Appendix A. A new version of the algorithm of Abraham

In this appendix a version of the algorithm of Abraham is presented that is more efficient than the one given in [23] when applied to systems having more than two operating modes. The reason for the better efficiency is that each component must be in exactly one operating mode, which is expressed by the formulas in $\mathcal{S}$ given in (45). If Σ denotes the collection of all those formulas then define

$$\xi = \bigwedge_{\sigma \in \Sigma} \sigma. \tag{A.1}$$

Two formulas f_1 and f_2 in $\mathcal{S}$ are called ξ-disjoint if $f_1 \wedge f_2 \wedge \xi = \perp$. In this case we write $f_1 \oplus f_2$ instead of $f_1 \vee f_2$. It can easily be proved that $\oplus$ is associative and commutative, so we can write $\bigoplus_{i \in I} f_i$ for a collection f_i, $i \in I$ of pairwise ξ-disjoint formulas. From now on, by a conjunction we mean a conjunction of literals over the atoms $\{m_j(c_i)\colon\ i = 1, \ldots, n, j = 1, \ldots, r_i\}$. When necessary, we always simplify the conjunctions such that each literal appears only once. Note that if two formulas are disjoint then they are also ξ-disjoint.

A conjunction is called **positive** if it does not contain any negated atom of the form $\sim m_j(c_i)$. If a DNF contains only positive conjunctions, it is called a **positive DNF**.

Lemma A.1. *Let c' and c'' be two positive conjunctions. Then c' and c'' are ξ-disjoint if and only if one of the following conditions is satisfied:*

(1) *c' or c'' contains two literals of the form $m_j(c_i)$ and $m_k(c_i)$ for some $i \in \{1, \ldots, n\}$ and $j \neq k$,*

(2) *there exist $i \in \{1, \ldots, n\}$ and two different indices j and k in $\{1, \ldots, r_i\}$ such that $m_j(c_i)$ is a literal of c' and $m_k(c_i)$ is a literal of c''.*

This lemma provides an easy way to test if two conjunctions are ξ-disjoint.

Before we formulate the next lemma, let us introduce some notation. For an atom $y = m_j(c_i)$, if we define the set

$$\overline{y} = \{m_k(c_i)\colon\ k \in \{1, \ldots, r_i\} - \{j\}\} \tag{A.2}$$

then we have

$$\sim y \wedge \xi = \Bigl(\bigvee_{t \in \overline{y}} t\Bigr) \wedge \xi. \tag{A.3}$$

As we are going to see later, to apply the algorithm described below we need a method to transform a DNF φ in $\mathcal{S}$ into a positive DNF η in $\mathcal{S}$ such that

$$\varphi \wedge \xi = \eta \wedge \xi. \tag{A.4}$$

In addition, the conjunctions in the positive DNF η need to satisfy a certain property described below. The method goes as follows. Let c be a conjunction in $\mathcal{S}$ containing a negative literal $\sim y$ and let c' denote the conjunction obtained by removing $\sim y$ from c. Then by Eq. (A.3) we have

$$c \wedge \xi = (\sim y c') \wedge \xi = (\sim y \xi) c' = \left(\bigvee_{t \in \overline{y}} tc'\right) \wedge \xi. \tag{A.5}$$

Since all conjunctions tc' contain one negated atom less then c, a repeated application of this procedure will finally generate a collection of positive conjunctions $\eta_1, \ldots, \eta_s$ such that $c \wedge \xi = (\bigvee_{i=1}^{s} \eta_i) \wedge \xi$. Now if $\varphi = \varphi_1 \vee \cdots \vee \varphi_m$ is a DNF in $\mathcal{S}$, then the above procedure applied to each conjunction c_i generates a collection of positive conjunctions $\eta_1, \ldots, \eta_r$ such that

$$(\varphi_1 \vee \cdots \vee \varphi_m) \wedge \xi = (\eta_1 \vee \cdots \vee \eta_r) \wedge \xi. \tag{A.6}$$

A **positive** conjunction c in $\mathcal{S}$ is called **ξ-compatible** if it does not contain a pair of atoms $m_j(c_i), m_k(c_i)$ with $j \neq k$. Since $\eta_i \wedge \xi = \perp$ for a non ξ-compatible conjunction η_i, we can remove the non ξ-compatible conjunctions η_i in Eq. (A.6). This implies that every DNF φ in $\mathcal{S}$ can be transformed into a positive DNF η containing only ξ-compatible conjunctions and such that $\varphi \wedge \xi = \eta \wedge \xi$.

The following lemma will be used in the proof of the algorithm.

Lemma A.2. *Let c' and c'' be two positive conjunctions which are not ξ-disjoint. Let $Y' = \{y_1, \ldots, y_s\}$ denote the set of all literals contained in c' but not in c''. Then we have*
(1) *if Y' is empty then $c' \vee c'' = c'$,*
(2) *if Y' is not empty then*

$$\begin{aligned}(c' \vee c'') \wedge \xi \\ = \left[c' + \left(\bigoplus_{t \in \overline{y_1}} tc''\right) + \left(\bigoplus_{t \in \overline{y_2}} y_1 tc''\right) + \cdots + \left(\bigoplus_{t \in \overline{y_s}} y_1 \ldots y_{s-1} tc''\right)\right] \wedge \xi.\end{aligned} \tag{A.7}$$

Note that all conjunctions that appear in Eq. (A.7) are positive.

Before we formulate the new algorithm of Abraham we need some additional notation. Let c' and c'' be two positive conjunctions which are not ξ-disjoint. If $Y' = (y_1, \ldots, y_s)$ is not empty, then define the set

$$\begin{aligned}R(c', c'') = \{tc'' \colon t \in \overline{y_1}\} \cup \{y_1 tc'' \colon t \in \overline{y_2}\} \cup \\ \cdots \cup \{y_1 \ldots y_{s-1} tc'' \colon t \in \overline{y_s}\}\end{aligned} \tag{A.8}$$

consisting of all positive conjunctions that are defined in Lemma A.2. Now we are in a position to formulate the following new version of the algorithm of Abraham.

Algorithm. The input of the algorithm is a DNF $g = g_1 \vee \cdots \vee g_m$. Then, using the procedure described above, g is transformed into a positive DNF $f = f_1 \vee \cdots \vee f_{m'}$ such that $g \wedge \xi = f \wedge \xi$ and all conjunctions f_i are ξ-compatible. The positive conjunctions

$f_1, \ldots, f_{m'}$ are then the input for the following procedure whose result is the collection of sets $P_{i,j}$ with $j = 1, \ldots, m'$, $i = 0, \ldots, j-1$.

```
for $j = 1$ to $m'$
    $P_{0,j} := \{f_j\}$
    for $i = 1$ to $j - 1$
        $P_{i,j} := \emptyset$
        for all $d_k$ in $P_{i-1,j}$
            if $d_k$ and $f_i$ are $\xi$-disjoint (see Lemma A.1) then add $d_k$ to $P_{i,j}$
            else define $Y' = \{y_1, \ldots, y_l\}$ to be the set of literals
                    in $f_i$ that are not contained in $d_k$
                if $Y' \neq \emptyset$ then add all elements in $R(f_i, d_k)$ to $P_{i,j}$.
```

Theorem A.3. *Let the DNF $g = g_1 \vee \cdots \vee g_m$ be the input of the above algorithm and let $P_{0,1}, \ldots, P_{m'-1,m'}$ be the sets $P_{j-1,j}$ that the algorithm generates. Then these sets contain only positive conjunctions and we have*

$$(g_1 \vee \cdots \vee g_m) \wedge \xi = \left[\bigoplus_{j=1}^{m'} \bigoplus \{d_k \in P_{j-1,j}\}\right] \wedge \xi. \tag{A.9}$$

Theorem A.4. *The ξ-disjoint conjunctions in $P = \bigcup\{P_{j-1,j}\colon\ j = 1, \ldots, m'\}$ generated by the algorithm are all ξ-compatible.*

Together with Theorem A.3 and Lemma A.1 this result implies that the conjunctions c in P are such that

- c is positive,
- if c contains the atom $m_j(c_i)$, then c does not contain any atom in
$$\{m_1(c_i), \ldots, m_{j-1}(c_i), m_{j+1}(c_i), \ldots, m_{r_i}(c_i)\}, \tag{A.10}$$
- if c' and c'' are two conjunctions in P, then there is at least one $i \in \{1, \ldots, n\}$ and two different elements j and k in $\{1, \ldots, r_i\}$ such that $m_j(c_i)$ is in c' and $m_k(c_i)$ is in c''.

Now we show how to apply Theorems A.3 and A.4 to compute $p(qs(\perp))$. First we need two lemmas:

Lemma A.5. *Let g be any formula in $\mathcal{S}$. Then $p(g \wedge \xi) = p(g)$.*

Lemma A.6. *Let $g_1, \ldots, g_r$ be a collection of mutually ξ-disjoint formulas in $\mathcal{S}$. Then*

$$p\left(\left(\bigoplus_{i=1}^{r} g_i\right) \wedge \xi\right) = \sum_{i=1}^{r} p(g_i). \tag{A.11}$$

Now we are in a position to formulate the following important result.

Corollary A.7. *Let the DNF $g = g_1 \vee \cdots \vee g_m$ be the input of the above algorithm and let $P_{0,1}, \ldots, P_{m'-1,m}$ be the sets $P_{j-1,j}$ that the algorithm generates. Then*

$$p(g) = \sum_{j-1}^{m'} \sum_{d_k \in P_{j-1,j}} p(d_k). \tag{A.12}$$

In this corollary, if we take for g a DNF representation of $qs(\perp)$, i.e., $qs(\perp) = dc_1 \vee \cdots \vee dc_m$, then formula (A.12) implies

$$p\big(qs(\perp)\big) = \sum_{j=1}^{m'} \sum_{d_k \in P_{j-1,j}} p(d_k). \tag{A.13}$$

The probabilities $p(d_k)$ are easy to compute because they are positive ξ-compatible conjunctions according to Theorems A.3 and A.4:

$$\text{if } d_k = \bigwedge_{(i,j)\in K} m_j(c_i), \quad \text{then} \quad p(d_k) = \prod_{(i,j)\in K} p_{ij}, \tag{A.14}$$

where p_{ij} is the prior probability that component i is in mode j. Starting from a DNF representation of $qs(h)$ obtained by using Corollary 7, Corollary A.7 can also be used to compute $p(qs(h))$ efficiently. Then the posterior probability of the hypothesis h is simply given by

$$p'(h) = \frac{p(qs(h)) - p(qs(\perp))}{1 - p(qs(\perp))} \tag{A.15}$$

according to Eq. (38).

Example 5. Consider a system with four components having the following operating modes:

c_1: modes m_1 to m_4, c_3: modes m_1 and m_2,
c_2: modes m_1 to m_4, c_4: modes m_1 and m_2.

Suppose that

$$\begin{aligned} qs(\perp) = {} & \big(m_2(c_1) \wedge {\sim}m_3(c_2) \wedge {\sim}m_4(c_2) \wedge m_2(c_3)\big) \\ & \vee \big(m_1(c_1) \wedge m_2(c_4)\big) \vee \big(m_2(c_1) \wedge m_2(c_4)\big). \end{aligned}$$

First we transform the two conjunctions of $qs(\perp)$ into the positive conjunctions

$$\begin{aligned} f_1 &= m_2(c_1) \wedge m_1(c_2) \wedge m_2(c_3), & f_2 &= m_2(c_1) \wedge m_2(c_2) \wedge m_2(c_3), \\ f_3 &= m_1(c_1) \wedge m_2(c_4), & f_4 &= m_2(c_1) \wedge m_2(c_4). \end{aligned}$$

Then the algorithm produces the following sets for $j = 1, \ldots, 3$:

$$P_{0,1} = \{f_1\},$$
$$P_{0,2} = \{f_2\} = P_{1,2},$$
$$P_{0,3} = \{f_3\} = P_{1,3} = P_{2,3}$$

as the three conjunctions f_1, f_2 and f_3 are mutually ξ-disjoint. More interesting is the step $j = 4$: we start with $P_{0,4} = \{f_4\}$ and the iteration over i goes as follows:

$i = 1$: f_1 and f_4 are not ξ-disjoint and we calculate $Y' = \{m_2(c_3), m_1(c_2)\}$. Therefore, we get

$$\begin{aligned} P_{1,4} = \big\{&m_2(c_1) \wedge m_2(c_4) \wedge m_1(c_2) \wedge m_1(c_3),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_3(c_2),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_4(c_2),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_2(c_2)\big\}. \end{aligned}$$

$i = 2$: the first three conjunctions in $P_{1,4}$ are ξ-disjoint with f_2 so they will be put unmodified into $P_{2,4}$, whereas the fourth conjunction is not ξ-disjoint with f_2 and so we have $Y' = \{m_2(c_3)\}$. This gives

$$\begin{aligned} P_{2,4} = \big\{&m_2(c_1) \wedge m_2(c_4) \wedge m_1(c_2) \wedge m_1(c_3),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_3(c_2),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_4(c_2),\\ &m_2(c_1) \wedge m_2(c_4) \wedge m_2(c_2) m_1(c_3)\big\}. \end{aligned}$$

$i = 3$: as all conjunctions in $P_{2,4}$ are ξ-disjoint with f_3 we have $P_{3,4} = P_{2,4}$.

Then the algorithm stops and according to Corollary A.7 we have

$$\begin{aligned} p\big(qs(\bot)\big) = {}& p\big(m_2(c_1) \wedge m_1(c_2) \wedge m_2(c_3)\big)\\ &+ p\big(m_2(c_1) \wedge m_2(c_2) \wedge m_2(c_3)\big)\\ &+ p\big(m_1(c_1) \wedge m_2(c_4)\big)\\ &+ p\big(m_2(c_1) \wedge m_2(c_4) \wedge m_1(c_2) \wedge m_1(c_3)\big)\\ &+ p\big(m_2(c_1) \wedge m_2(c_4) \wedge m_3(c_2)\big)\\ &+ p\big(m_2(c_1) \wedge m_2(c_4) \wedge m_4(c_2)\big)\\ &+ p\big(m_2(c_1) \wedge m_2(c_4) \wedge m_2(c_2) \wedge m_1(c_3)\big). \end{aligned}$$

Appendix B. Proofs of theorems

Proof of Lemma 1. First suppose that $X, f \Rightarrow h$, i.e., $X, f, {\sim}h \Rightarrow \bot$, which is equivalent to $X, c(x) \Rightarrow \bot$ whenever $x \models f, {\sim}h$. We have to prove that $X, c(x) \Rightarrow h$ whenever $x \models f$, i.e., $X, c(x), {\sim}h \Rightarrow \bot$ whenever $x \models f$, which is equivalent to $X, c(y) \Rightarrow \bot$ whenever $y \models c(x), {\sim}h$ for all x such that $x \models f$. So let x such that $x \models f$ and y such that $y \models c(x), {\sim}h$ and we must prove that $X, c(y) \Rightarrow \bot$. But $y \models c(x)$ implies that $y = x$ and we must prove that $X, c(x) \Rightarrow \bot$. But $x \models f$ and $x \models {\sim}h$ (using $x = y$) imply that $X, c(x) \Rightarrow \bot$ by hypothesis.

Conversely, let x such that $x \models f$ and $x \models \sim h$ and we must prove that $X, c(x) \Rightarrow \bot$. But by hypothesis $X, c(y) \Rightarrow \bot$ for all y such that $y \models c(x), \sim h$ and for all x such that $x \models f$ (see above). This implies that we can take $y = x$ and hence $X, c(x) \Rightarrow \bot$. □

Proof of Lemma 2. Suppose that *co* is an implicant of *cont*. We must prove that $X_{sys}, co \Rightarrow \bot$, i.e., $X_{sys}, c(y) \Rightarrow \bot$ for all y such that $y \models co$. So let y such that $y \models co$. Since $co \models \bigvee_{x \in N_c} c(x)$ we have $y \models \bigvee_{x \in N_c} c(x)$ and hence there is $x \in N_c$ such that $y \models c(x)$, which implies that $y = x$. Since $x \in N_c$ we have $X_{sys}, c(x) \Rightarrow \bot$ and hence $X_{sys}, c(y) \Rightarrow \bot$ because $x = y$.

Conversely, suppose that *co* is a conflict set, i.e., $X_{sys}, co \Rightarrow \bot$, i.e., $X_{sys}, c(x) \Rightarrow \bot$ for all x such that $x \models co$. We must prove that $co \models cont$, i.e., $co \models \bigvee_{x \in N_c} c(x)$. So let y such that $y \models co$. By hypothesis it follows that $X_{sys}, c(y) \Rightarrow \bot$ and hence $y \in N_c$. This implies that $y \models c(y) \models \bigvee_{x \in N_c} c(x)$ and hence $y \models \bigvee_{x \in N_c} c(x)$ which completes the proof of the lemma. □

Proof of Lemma 3. First we prove (1). We must prove that if $x \models cont$ then $X_{sys}, c(x) \Rightarrow \bot$. But $x \models cont$ implies that $x \in N_c$ and hence $X_{sys}, c(x) \Rightarrow \bot$.

Now assertion (2) is proved. Assume that $X_{sys}, f \Rightarrow \bot$, i.e., $X_{sys}, c(x) \Rightarrow \bot$ for all x such that $x \models f$. We must prove that if $x \models f$ then $x \models cont$. But by assumption $x \models f$ implies that $X_{sys}, c(x) \Rightarrow \bot$ and hence $x \in N_c$ and therefore $x \models \bigvee_{y \in N_c} c(y)$, i.e., $x \models cont$. □

Proof of Lemma 4. First we prove (1). Using Lemma 1, we must prove that if $x \models qs(h)$ then $X, c(x) \Rightarrow h$. But $x \models qs(h)$ implies that there is $y \in Q(h)$ such that $x \models c(y)$, i.e., there is $y \in Q(h)$ such that $x = y$. But this implies that $X_{sys}, c(x) \Rightarrow h$.

Now assertion (2) is proved. Let x such that $x \models f$. Thus we must prove that $x \models qs(h)$, i.e., $x \in Q(h)$, i.e., $X_{sys}, c(x) \Rightarrow h$. But by hypothesis $X, f \Rightarrow h$ and by Lemma 1 $X, c(x) \Rightarrow h$ since $x \models f$. □

Proof of Lemma 5. First we prove (1):

$$N(qs(h)) = \{x\colon x \models qs(h)\} = \left\{x\colon x \models \bigvee_{y \in Q(h)} c(y)\right\} = Q(h).$$

Now (2) is proved: it must be proved that $qs(\bot) \models qs(h)$, i.e., $N(qs(\bot)) \subseteq N(qs(h))$, i.e., $N_c \subseteq Q(h)$. So let $x \in N_c$, i.e., $X_{sys}, c(x) \Rightarrow \bot$ and we must show that $X_{sys}, c(x) \Rightarrow h$. But $X_{sys}, c(x) \Rightarrow h$ by definition means that $X_{sys}, c(y) \Rightarrow \bot$ for all y such that $y \models c(x), \sim h$, which in turn is the same as $X_{sys}, c(y) \Rightarrow \bot$ for all $y \in \{x\} \cap \{t\colon t \models \sim h\}$. So let $y \in \{x\} \cap \{t\colon t \models \sim h\}$. Then $y = x$ and since $X_{sys}, c(x) \Rightarrow \bot$ it follows that $X_{sys}, c(y) \Rightarrow \bot$ and the lemma is proved. □

Proof of Theorem 6. Since $N(qs(h)) = Q(h)$ we must prove that $Q(h) = N(h) \cup N_c$. First we prove that $Q(h) \subseteq N(h) \cup N_c$. Let $x \in Q(h)$, i.e., $X_{sys}, c(x) \Rightarrow h$. If $x \in N(h)$ then we are done. So assume that $x \notin N(h)$ and it must be proved that $x \in N_c$, i.e., $X_{sys}, c(x) \Rightarrow \bot$. But $X_{sys}, c(x) \Rightarrow h$ is the same as

$$X_{sys}, c(x), \sim h \Rightarrow \bot, \tag{B.1}$$

which is the same as

$$X_{sys}, c(y) \Rightarrow \bot \tag{B.2}$$

for all y such that $y \models c(x), \sim h$, which is the same as

$$X_{sys}, c(y) \Rightarrow \bot \tag{B.3}$$

for all $y \in \{x\} \cap N(\sim h)$. But since $x \notin N(h)$ it follows that $x \in N(\sim h)$. Thus (B.3) implies that $X_{sys}, c(x) \Rightarrow \bot$, which is what was to be proved.

Now we prove that $N(h) \cup N_c \subseteq Q(h)$. Since by the second assertion of Lemma 5 we have that $N_c \subseteq Q(h)$, it remains to prove that $N(h) \subseteq Q(h)$. So let $x \in N(h)$ and we must prove that $x \in Q(h)$, which in turn is the same as

$$X_{sys}, c(y) \Rightarrow \bot \quad \text{for every } y \in \{x\} \cap N(\sim h) \tag{B.4}$$

as we have seen above. But $x \in N(h)$ implies that $x \notin N(\sim h)$ and hence $\{x\} \cap N(\sim h) = \emptyset$ which implies that condition (B.4) is satisfied. □

Proof of Corollary 7. This follows from Theorem 6 and (7). □

Proof of Theorem 8. Everything follows easily from Theorem 6:

$$\begin{aligned} qs(h_1 \wedge h_2) &= (h_1 \wedge h_2) \vee qs(\bot) = \big(h_1 \vee qs(\bot)\big) \wedge \big(h_2 \vee qs(\bot)\big) \\ &= qs(h_1) \wedge qs(h_2), \\ qs(h_1 \vee h_2) &= (h_1 \vee h_2) \vee qs(\bot) = \big(h_1 \vee qs(\bot)\big) \vee \big(h_2 \vee qs(\bot)\big) \\ &= qs(h_1) \vee qs(h_2), \\ qs(\top) &= \top \vee qs(\bot) = \top. \quad \square \end{aligned}$$

Proof of Lemma 9. If $X_{sys}, f \Rightarrow \bot$ then f is a contradictory argument and hence $f \models cont$ by the second assertion of Lemma 3. Conversely, assume that $f \models cont$, i.e., $N(f) \subseteq N(cont)$. We must prove that $X_{sys}, f \Rightarrow \bot$, which is equivalent to $X_{sys}, c(x) \Rightarrow \bot$ for all $x \in N(f)$ according to Lemma 1. But by Lemma 3, (1) we have $X_{sys}, cont \Rightarrow \bot$ and hence by Lemma 1 $X_{sys}, c(x) \Rightarrow \bot$ for all $x \in N(cont)$. Since $N(f) \subseteq N(cont)$ it follows that $X_{sys}, c(x) \Rightarrow \bot$ for all $x \in N(f)$. □

Proof of Lemma 10. First we prove that $s(h)$ is an argument of h, i.e., $X_{sys}, s(h) \Rightarrow h$. By Lemma 1 we must prove that $X_{sys}, c(x) \Rightarrow h$ for all $x \in N(s(h)) = N(qs(h)) \cap N(\sim qs(\bot))$. By assertion (1) of Lemma 4 and by Lemma 1 we have $X_{sys}, c(x) \Rightarrow h$ for all $x \in N(qs(h))$. Since $N(s(h)) \subseteq N(qs(h))$ it follows that $X_{sys}, c(x) \Rightarrow h$ for all $x \in N(s(h))$.

Now we prove that $s(h)$ is not a contradictory argument, which is the same as $s(h) \not\models qs(\bot)$ according to Lemma 9. In other words, we must prove that

$$N\big(s(h)\big) \cap N\big(qs(\bot)\big)^c \neq \emptyset. \tag{B.5}$$

But we have

$$\begin{aligned} N(s(h)) \cap N(qs(\perp))^c &= N(qs(h)) \cap N(\sim qs(\perp)) \cap N(qs(\perp))^c \\ &= N(qs(h)) \cap N(qs(\perp))^c \cap N(qs(\perp))^c \\ &= N(qs(h)) \cap N(qs(\perp))^c \\ &= N(qs(h) \wedge \sim qs(\perp)) \\ &= N(s(h)) \end{aligned}$$

and hence Eq. (B.5) is satisfied because $N(s(h)) \neq \emptyset$ since $s(h) \neq \perp$. □

Proof of Corollary 11. By Corollary 7, we can write

$$s(h) = qs(h) \wedge \sim qs(\perp) = (h \vee qs(\perp)) \wedge \sim qs(\perp) = h \wedge \sim qs(\perp). \qquad \square$$

Proof of Theorem 12. First remark that p'' restricted to N_d equals p'. Then, according to (21), we can write

$$\begin{aligned} sp(h) &= p''(N(s(h))) = p''(N(h \wedge \sim qs(\perp))) \\ &= p''(N(h) \cap N_d) = p'(N(h) \cap N_d) \\ &= \frac{p((N(h) \cap N_d) \cap N_d)}{p(N_d)} \\ &= \frac{p(N(h) \cap N_d)}{p(N_d)} = p'(h). \qquad \square \end{aligned}$$

Proof of Lemma A.1. The "if" part is obvious. To prove the "only if" part, assume that (1) and (2) are false. Then since (1) is false, $c' \wedge \xi$ and $c'' \wedge \xi$ are satisfiable and hence, since (2) is false, it is possible to find an interpretation $x \in B_m$ satisfying $c' \wedge c'' \wedge \xi$. This shows that $c' \wedge c'' \wedge \xi$ is not contradictory and the lemma is proved. □

Proof of Lemma A.2. Since c' and c'' are not ξ-disjoint they are not disjoint. Then Theorem 5.27 [23, p. 112] implies that if Y' is empty then (1) is satisfied. Otherwise, if Y' is not empty, the same theorem implies that

$$c' \vee c'' = c' + (\sim y_1)c'' + y_1(\sim y_2)c'' + \cdots + y_1 \ldots y_{s-1}(\sim y_s)c''. \qquad \text{(B.6)}$$

Along with (A.3) this in turn implies that

$$\begin{aligned} &(c' \vee c'') \wedge \xi \\ &= c'\xi + \sim y_1 c''\xi + y_1 \sim y_2 c''\xi + \cdots + y_1 \ldots y_{s-1} \sim y_s c''\xi \\ &= c'\xi + \Big(\bigvee_{t \in \overline{y_1}} tc''\Big)\xi + \Big(\bigvee_{t \in \overline{y_2}} y_1 tc''\Big)\xi + \cdots + \Big(\bigvee_{t \in \overline{y_s}} y_1 \ldots y_{s-1} tc''\Big)\xi \\ &= \Big[c' + \Big(\bigoplus_{t \in \overline{y_1}} tc''\Big) + \Big(\bigoplus_{t \in \overline{y_2}} y_1 tc''\Big) + \cdots + \Big(\bigoplus_{t \in \overline{y_s}} y_1 \ldots y_{s-1} tc''\Big)\Big] \wedge \xi. \qquad \square \end{aligned}$$

Proof of Theorem A.3. For a fixed j, first we have to prove by induction on i that $P_{i,j}$ contains only positive conjunctions for all $i < j$. This is true for $i = 0$, i.e., $P_{0,j} = \{g_j\}$. Now suppose that this is true for $P_{i-1,j}$. Then, with $d_k \in P_{i-1,j}$ the algorithm puts into $P_{i,j}$ either the positive conjunction d_k or the positive conjunctions in $R(f_i, d_k)$. This proves that all conjunctions in $P_{i,j}$ are positive and for $i = j - 1$ we obtain that $P_{j-1,j}$ only contains positive conjunctions. Since j was arbitrary, this proves the first statement of the theorem.

For Eq. (A.9), we prove again by induction on i that $P_{i,j}$ has the following properties:
(1) $(f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee (f_j \wedge \xi) = (f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee \{d_k \wedge \xi\colon d_k \in P_{i,j}\}$;
(2) all conjunctions in $P_{i,j}$ are ξ-disjoint;
(3) for all $d_k \in P_{i,j}$, f_k and d_k are ξ-disjoint for all $k \in \{1, \ldots, i\}$.
This is clearly true for $i = 0$, i.e., $P_{0,j} = \{f_j\}$. We suppose that $P_{i-1,j}$ satisfies these properties and let us show that $P_{i,j}$ also satisfies them. First we prove assertion (1). By the induction hypothesis we have

$$\begin{aligned}&(f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee (f_j \wedge \xi)\\ &\quad = (f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee \{d_k \wedge \xi\colon d_k \in P_{i-1,j}\}\end{aligned}$$

and we must prove that

$$\begin{aligned}&(f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee \{d_k \wedge \xi\colon d_k \in P_{i-1,j}\}\\ &\quad = (f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi) \vee \{d_k \wedge \xi\colon d_k \in P_{i,j}\}. \qquad (B.7)\end{aligned}$$

First we show that if the left-hand side (LHS) of (B.7) is true then so is its right-hand side (RHS). Suppose that $d_k \wedge \xi$ is true with $d_k \in P_{i-1,j}$. Two cases can happen. If d_k and f_i are ξ-disjoint, then d_k is also in $P_{i,j}$ and hence the RHS of (B.7) is also true. If d_k and f_i are not ξ-disjoint then again two cases can happen: if Y' is empty then Lemma A.2(1) (take $c' = f_i$ and $c'' = d_k$) implies that $(f_i \wedge \xi) \vee (d_k \wedge \xi) = f_i \wedge \xi$ and hence $f_i \wedge \xi$ is true and so is the RHS of (B.7); if Y' is not empty then by Lemma A.2(2) we have

$$(f_i \wedge \xi) \vee (d_k \wedge \xi) = (f_i \wedge \xi) \vee \big\{r \wedge \xi\colon r \in R(f_i, d_k)\big\}. \qquad (B.8)$$

Since $d_k \wedge \xi$ is true, the RHS of (B.8) is also true and hence the RHS of (B.7) is also true since $R(f_i, d_k) \subseteq P_{i,j}$.

Now let us prove that if the RHS of (B.7) is true then its LHS is also true. So let $d_k \wedge \xi$ be true, with $d_k \in P_{i,j}$. There are only two ways for d_k to be in $P_{i,j}$: either d_k itself is in $P_{i-1,j}$ in which case the LHS of (B.7) is true; or else there is a d'_k in $P_{i-1,j}$ such that $d_k \in R(f_i, d'_k)$. In this case $d_k = d'_k \wedge v$ for some conjunction v. Hence, since $d_k \wedge \xi$ is true, so is $d'_k \wedge v \wedge \xi$ and so is $d'_k \wedge \xi$, which implies that the LHS of (B.7) is true because $d'_k \in P_{i-1,j}$.

The assertions (2) and (3) are clearly true.

Note that property (3) implies that $(f_1 \wedge \xi) \vee \cdots \vee (f_i \wedge \xi)$ and $\bigoplus\{d_k \wedge \xi\colon d_k \in P_{i,j}\}$ are ξ-disjoint. Then, taking $i = j - 1$, we get

$$\begin{aligned}&(f_1 \wedge \xi) \vee \cdots \vee (f_j \wedge \xi)\\ &\quad = \big[(f_1 \wedge \xi) \vee \cdots \vee (f_{j-1} \wedge \xi)\big] \oplus \Big[\bigoplus\{d_k \wedge \xi\colon d_k \in P_{i,j}\}\Big]\end{aligned}$$

and then by induction on j we obtain

$$(f_1 \wedge \xi) \vee \cdots \vee (f_{m'} \wedge \xi)$$
$$= \bigoplus \{d_k \wedge \xi \colon d_k \in P_{0,1}\} \bigoplus \cdots \bigoplus \{d_k \wedge \xi \colon d_k \in P_{m'-1,m'}\}.$$

Since $g \wedge \xi = f \wedge \xi$ (see the preamble of the algorithm) it follows that

$$(g_1 \vee \cdots \vee g_m) \wedge \xi = \left(\bigoplus_{j=1}^{m'} \bigoplus \{d_k \in P_{j-1,j} \right) \wedge \xi$$

and the theorem is proved. □

Proof of Theorem A.4. The result follows from the fact that all sets $P_{i,j}$, $j = 1, \ldots, m'$, $i = 0, \ldots, j-1$ generated by the algorithm contain only ξ-compatible conjunctions. Indeed, $P_{0,j} = \{f_j\}$ and f_j is ξ-compatible and it remains to prove that if all conjunctions in $P_{i-1,j}$ are ξ-compatible then all conjunctions in $P_{i,j}$ are also ξ-compatible. Let d_k in $P_{i-1,j}$. If d_k and f_i are ξ-disjoint then the new element d_k in $P_{i,j}$ is ξ-compatible. It remains to prove that if d_k and f_i are not ξ-disjoint and $Y' \neq \emptyset$ then all conjunctions in $R(f_i, d_k)$ are ξ-compatible. To prove this, it is clearly sufficient to show that if $m_j(c_h)$ is an atom in Y', then it does not appear as an atom of d_k. On the contrary, suppose that the atom $m_l(c_h)$ is in d_k. Since f_i and d_k are not ξ-disjoint, we must have $j = l$ and hence $m_j(c_h) = m_l(c_h)$ which is a contradiction because $m_j(c_h) \in Y'$ and Y' does not contain any atom of d_k. □

Proof of Lemma A.5. First note that $p(\sim\xi) = 0$ because the interpretations satisfying $\sim\xi$ do not correspond to possible system states. Since $g = (g \wedge \xi) + (g \wedge \sim\xi)$ it follows that

$$N(g) = N(g \wedge \xi) + N(g \wedge \sim\xi) \tag{B.9}$$

and hence

$$p(g) = p(g \wedge \xi) + p(g \wedge \sim\xi) = p(g \wedge \xi) \tag{B.10}$$

because $0 \leqslant p(g \wedge \sim\xi) \leqslant p(\sim\xi) = 0$. □

Proof of Lemma A.6. By the inclusion-exclusion formula (39) we have

$$p\left(\bigoplus_{i=1}^{r} (g_i \wedge \xi) \right) = \sum_{\emptyset \neq I \subseteq \{1,\ldots,r\}} (-1)^{|I|+1} p\left(\bigwedge_{i \in I} (g_i \wedge \xi) \right). \tag{B.11}$$

But, since the g_i are mutually ξ-disjoint, for all I containing at least two elements k and l we have

$$p\left(\bigwedge_{i \in I} (g_i \wedge \xi) \right) \leqslant p\big((g_k \wedge \xi) \wedge (g_l \wedge \xi) \big) = p(\bot) = 0. \tag{B.12}$$

Therefore,

$$\sum_{\emptyset \neq I \subseteq \{1,\ldots,r\}} (-1)^{|I|+1} p\left(\bigwedge_{i \in I} (g_i \wedge \xi)\right) = \sum_{i=1}^{r} p(g_i \wedge \xi) \tag{B.13}$$

and since $p(g_i \wedge \xi) = p(g_i)$ by Lemma A.5 we obtain Eq. (A.11). □

Proof of Corollary A.7. We can write

$$\begin{aligned} p(g) &= p(g \wedge \xi) && \text{(by Lemma A.5)} \\ &= p\left(\left(\bigvee_{i=1}^{m} g_i\right) \wedge \xi\right) \\ &= p\left(\left(\bigoplus_{j=1}^{m'} \bigoplus \{d_k \in P_{j-1,j}\}\right) \wedge \xi\right) && \text{(by Theorem A.3)} \\ &= \sum_{j=1}^{m'} \sum_{d_k \in P_{j-1,j}} p(d_k) && \text{(by Lemma A.6).} \end{aligned}$$ □

References

[1] J. Abraham, An improved algorithm for network reliability, IEEE Transactions on Reliability 28 (1979) 58–61.

[2] B. Anrig, R. Haenni, J. Kohlas, N. Lehmann, Assumption-based modeling using ABEL, in: D. Gabbay, R. Kruse, A. Nonnengart, H. Ohlbach (Eds.), 1st International Joint Conference on Qualitative and Quantitative Practical Reasoning; ECSQARU–FAPR-97, Lecture Notes in Artif. Intell., Springer, Berlin, 1997.

[3] B. Anrig, R. Haenni, J. Kohlas, P.A. Monney, Probabilistic analysis of model-based diagnosis, in: IPMU-96, Proc. 6th International Conference, Granada, Spain, 1996, pp. 123–128.

[4] B. Anrig, R. Haenni, N. Lehmann, ABEL—a new language for assumption-based evidential reasoning under uncertainty, Technical Report 97–01, University of Fribourg, Institute of Informatics, 1997.

[5] B. Anrig, N. Lehmann, R. Haenni, Reasoning with finite set constraints, Technical Report 97–11, Institute of Informatics, University of Fribourg, 1997.

[6] R. Barlow, F. Proschan, Statistical Theory of Reliability and Life Testing, Probability Models, New York, 1975.

[7] R. Bertschy, P.A. Monney, A generalization of the algorithm of Heidtmann to non-monotone formulas, Journal of Computational and Applied Mathematics 76 (1996) 55–76.

[8] A. Darwiche, Model-based diagnosis using causal networks, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 211–217.

[9] R. Davis, Diagnostic reasoning based on structure and behaviour, Artificial Intelligence 24 (1984) 347–410.

[10] J. de Kleer, Local methods for localizing faults in electronical circuits, MIT AI Memo 394, MIT, Cambridge, MA, 1976.

[11] J. de Kleer, Using crude probability estimates to guide diagnosis, Artificial Intelligence 45 (1990) 381–391.

[12] J. de Kleer, A. Mackworth, R. Reiter, Characterizing diagnosis and systems, Artificial Intelligence 56 (1992) 197–222.

[13] J. de Kleer, B. Williams, Diagnosing multiple faults, Artificial Intelligence 32 (1987) 97–130.

[14] W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd ed., Wiley, New York, 1968.

[15] K. Forbus, J. de Kleer, Building Problem Solvers, MIT Press, Cambridge, MA, 1993.

[16] M. Genesereth, The use of design description in automated diagnosis, Artificial Intelligence 24 (1984) 411–436.

[17] R. Haenni, N. Lehmann, Assumption-based reasoning with finite set constraints, in: IPMU-98, Proc. 7th International Conference, Paris, 1998, pp. 1289–1295.

[18] K. Heidtmann, Smaller sums of disjoint products by subproduct inversion, IEEE Transactions on Reliability 38 (3) (1989) 305–311.

[19] K. Inoue, Linear resolution for consequence finding, Artificial Intelligence 56 (1992) 301–353.

[20] J. Kohlas, Mathematical foundations of evidence theory, in: G. Coletti, D. Dubois, R. Scozzafava (Eds.), Mathematical Models for Handling Partial Knowledge in Artificial Intelligence, Plenum Press, New York, 1995, pp. 31–64.

[21] J. Kohlas, Allocation of arguments and evidence theory, Theoretical Computer Science 171 (1997) 221–246.

[22] J. Kohlas, P.A. Monney, Probabilistic assumption-based reasoning, in: D. Heckermann, A. Mamdani (Eds.), Proc. 9th Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1993.

[23] J. Kohlas, P.A. Monney, A Mathematical Theory of Hints. An Approach to Dempster–Shafer Theory of Evidence, Lecture Notes in Economics and Mathematical Systems, Vol. 425, Springer, Berlin, 1995.

[24] J. Kohlas, P.A. Monney, R. Haenni, N. Lehmann, Model-based diagnostics using hints, in: C. Fridevaux, J. Kohlas (Eds.), Symbolic and Quantitative Approaches to Uncertainty, European Conference, ECSQARU-95, Fribourg, Springer, Berlin, 1995, pp. 259–266.

[25] K. Laskey, P. Lehner, Assumptions, belief and probabilities, Artificial Intelligence 41 (1989) 65–77.

[26] P.A. Monney, B. Anrig, Computing the probability of formulas representing events in product spaces, in: IPMU-98, Proc. 7th International Conference, Paris, 1998.

[27] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, 2nd ed., Morgan Kaufmann, San Mateo, CA, 1991.

[28] G. Provan, A logic-based analysis of Dempster–Shafer theory, Internat. J. of Approximate Reasoning 4 (1990) 451–495.

[29] J. Reggia, D. Nau, Y. Wang, Diagnostic expert systems based on set covering model, Internat. J. Man-Machine Stud. 19 (1983) 437–460.

[30] J. Reggia, D. Nau, Y. Wang, A formal model of diagnostic inference, 1: Problem formulation and decomposition, Inf. Sci. 37 (1985) 227–256.

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

[32] R. Reiter, J. de Kleer, Foundations of assumption-based truth maintenance systems, in: Proc. AAAI-87, Seattle, WA, 1987, pp. 183–188.

[33] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[34] P.P. Shenoy, G. Shafer, Axioms for probability and belief functions propagation, in: R. Shachter et al. (Eds.), Uncertainty in Artificial Intelligence 4, North-Holland, Amsterdam, 1990.

[35] P. Siegel, Représentation et utilisation de la connaissance en calcul propositionel, Ph.D. thesis, Université d'Aix-Marseille II, Luminy, France, 1987.