ORiON, Vol. 16, No. 1, pp. 47–62 http://orion.journals.ac.za/ *ISSN 0259-191-X*

RELIABILITY AND DIAGNOSTIC OF MODULAR SYSTEMS *

JÜRG KOHLAS, BERNHARD ANRIG & ROMAN BISSIG
Department of Informatics
University of Fribourg
CH – 1700 Fribourg
Switzerland
http://www-iiuf.unifr.ch/tcs

ABSTRACT
Reliability and diagnostic are in general two problems discussed separately. Yet the two problems are in fact closely related to each other. Here, this relation is considered in the simple case of modular systems. We show, how the computation of reliability and diagnostic can efficiently be done within the same Bayesian network induced by the modularity of the structure function of the system.

1. INTRODUCTION

Reliability is concerned with the prediction of the correct functioning of systems. Diagnostic on the other hand aims at finding the cause of the malfunctioning of systems. Usually the two problems are discussed separately, each domain has its own literature. But in fact the two problems are closely related. This will be demonstrated in this paper in a simple context. We consider modular systems. It is well known that modularity helps in computing the reliability of a system. The same structure helps also for the task of diagnostic, once a malfunctioning of a modular system has been observed.

The logical dependence of the good functioning of a system on the intactness of its components can be described by a Boolean function, the structure function. This is well known in reliability theory. The structure function can be represented in different ways: one possible representation is by its minimal paths. Another one depends on the minimal cuts of the system. In any case, the structure function allows to compute the reliability of the system, if the component reliabilities are known. This task however is NP hard,

*Research supported by grant No. 2000-061454.00 of the Swiss National Foundation for Research.

i.e. in the worst case the computational effort grows exponentially with the number of components of the system. That is where modularity comes in. Technical systems are often built up of modules, which can be exchanged as a whole in the case of a defect. This technical structure is reflected in the logical structure of the system, i.e. in the structure function. A modular decomposition of a system can be represented by a corresponding graph, in fact a tree. Reliability computation can be much simplified using this tree. The method works from the leaves to the root, starting at the leaves, which represent the components of the system, computing intermediate nodes, corresponding to modules, up to the root, which designs the system. This approach is presented in Section 2.

This is so far a classical approach of reliability theory. But assume now, that the system is observed of not being functioning correctly. Then a problem of diagnostic arises: Which modules or which components are not functioning, causing the system to be down? Generally, it will not be possible to single out a unique set of modules or components which is responsible for the system break-down. The best we can hope for is to compute the posterior probabilities of each module and each component to be down, given the event of the system break-down. This will then permit to selectively test the most likely candidates for the cause of the system break-down. Using the results of these tests, new posterior probabilities can be obtained, which will progressively point more and more to the true cause of the system break-down. In this paper (Section 3) we show that the tree of the modular decomposition can be used to compute these posterior probabilities. In fact, once the system and module reliabilities have been obtained by traversing the tree from the leaves to the root, we can use the results of these computations, when we traverse this time the tree in the opposite direction from the root to the leaves. Essentially, we apply Bayes' theorem at each node to obtain sequentially the posterior probabilities of the descendents of the node. At the end, the posterior probabilities, given the event of a system break-down, have been computed for all modules and all components. This shows that diagnostic is indeed intimately related to reliability.

This duality between reliability and diagnostic can be enforced, if the calculations are related to a Bayesian network. Bayesian networks are very popular and a well developed computational theory is available for them. We show how a modular decomposition gives rise to a Bayesian network, in fact a Bayesian tree. Therefore, the computational procedures for Bayesian networks apply here and they solve indeed both the reliability as well as the diagnostic problem. We present in Section 4 one of the computational architectures for Bayesian networks and relate it to reliability and diagnostic computation. This gives us a unifying framework for both problems. Also it generalizes the computational approach presented in Birnbaum and Esary [5], Birnbaum et al. [6]. So, this shows again the duality between reliability and diagnostic. We remark, that modular systems are only very simple, also convenient, examples, where this duality can be exploited. Model-based

reliability and diagnostic is a much more general framework, where this duality finds its most general expression. A first discussion of this framework is in Provan [14]; yet this field has still to be explored.

2. RELIABILITY OF MODULAR SYSTEMS

Systems are made up from different components. At any given time point, these components may or may not work. Depending on the states of all its components, the system itself may or may not work. This dependence of the functioning of the system on the functioning of its components is described by a *structure function*. Suppose that the system is composed of $i = 1, \ldots, n$ components. Then we introduce a Boolean variable a_i for each component i with the meaning that

$$a_i \;=\; \begin{cases} \top & \text{if the component with number } i \text{ works,} \\ \bot & \text{otherwise.} \end{cases} \tag{1}$$

We use here the symbols $\bot$ and $\top$ borrowed from logic to denote the states of the components in order to clearly distinguish these states from other concepts introduced in the sequel. Clearly, in reliability theory, the symbols $\bot$ and $\top$ are usually interpreted as numerical values 0 and 1 respectively.

The state of the variables $a_1, \ldots, a_n$ can be summarized by a vector $\mathbf{a} = (a_1, \ldots, a_n)$. This vector has 2^n different possible states. The set of all possible states is divided into two subsets, the set $S_\top$ of states for which the system works and its complement, the set $S_\bot$ of states for which the system is down. The state of the system is denoted by a Boolean variable a. The dependence of the system state a on the vector $\mathbf{a}$ is denoted by the Boolean function ϕ, defined as

$$a \;=\; \phi(\mathbf{a}) \;=\; \begin{cases} \top & \text{if } \mathbf{a} \in S_\top, \\ \bot & \text{if } \mathbf{a} \in S_\bot. \end{cases} \tag{2}$$

The function ϕ is called the *structure function* of the system. It is a Boolean function, which maps n Boolean variables in a Boolean variable; for the basic definitions for boolean functions see Beichelt [4], Kohlas [11]. In the sequel, we assume that the systems are monotone, which means that the structure function ϕ is nondecreasing in each variable.

There are two important general representations of any monotone Boolean function. A subset $P \subseteq \{1, \ldots, n\}$ of components is called a *path* of ϕ, if $a_i = \top$ for all $i \in P$ implies that $\phi(\mathbf{a}) = \top$, independently of the state of the other components. That is, a path is a subset of components whose functioning is sufficient for the functioning of the system. A path P is called *minimal*, if no proper subset of P is a path. Let $\mathcal{P}$ be the family of minimal paths of ϕ. Then

$$\phi(\mathbf{a}) \;=\; \bigvee_{P \in \mathcal{P}} \bigwedge_{i \in P} a_i. \tag{3}$$

Here $\vee$ denotes disjunction (or) and $\wedge$ conjunction (and). So the formula above simply states that the system is functioning, if, and only if, all (conjunction) components of at least one (disjunction) minimal path are working. This is the *disjunctive normal form*. We assume in this paper, that the set of all components is a path: if all components work, then the system works. If this is the only path in a system, then it is called a *series system*. A component which belongs to no minimal path is irrelevant for the function of the system. It can be eliminated as far as the system reliability is concerned. We assume hereafter that all components are relevant.

A subset of components C is called a *cut* if the system is down, whenever all the elements of C are down (irrespective of the states of the other components). A cut C is called minimal, if no proper subset of C is a cut. Let $\mathcal{C}$ denote the family of all minimal cuts of the system. Then,

$$\phi(\mathbf{a}) \quad = \quad \bigwedge_{C \in \mathcal{C}} \bigvee_{i \in C} a_i. \tag{4}$$

This says that the system is functioning if at least (disjunction) one component in all (conjunction) minimal cuts is working. This is the *conjunctive normal form* of ϕ. The two representations of conjunctive and disjunctive normal forms are dual to each other. We assume that the set of all components is a cut, otherwise there would be no cut, hence no reliability problem. If this is the only cut in a system, then it is called a *parallel system*.

We assume that the probability p_i that the component i is working at a given time is known for every component i. The probabilities $p_1, \ldots, p_n$ are assumed to be mutually independent. Let $\mathbf{p}$ denote the vector $(p_1, \ldots, p_n)$. Then, the structure function ϕ determines the probability p that the system itself is functioning,

$$p \quad = \quad E(\phi) \quad = \quad P\{\phi = 1\}. \tag{5}$$

Clearly, this probability depends on the vector $\mathbf{p}$ of the component probabilities. In order to emphasize this dependence, we write

$$p \quad = \quad h_\phi(\mathbf{p}). \tag{6}$$

The computation of the probability p from the structure function is in general no trivial task. Many methods have been proposed, we refer to Beichelt [4] and Kohlas [11]. One method consists of transforming the disjunctive normal form into a disjunction of *disjoint* terms, whose probabilities can be easily computed and which can be simply summed up, since the terms are disjoint, see Abraham [1], Heidtmann [8]; this subject is discussed in the article "Disjoint Sum Forms in Reliability Theory" of Anrig & Beichelt in this journal [2].

So, in the simplest case of a disjoint form, this means, that there are terms $c_i = \bigwedge_{j \in I_i} l_j$ with $l_j = a_j$ or $\neg a_j$, such that $\phi = \bigvee_i c_i$ and $c_i \wedge c_j = \bot$ if $i \neq j$. Then we have

$$p \quad = \quad \sum_i p(c_i) \tag{7}$$

and

$$p(c_i) \quad = \quad \prod_{a_j \in c_i} p_j \cdot \prod_{\neg a_j \in c_i} (1 - p_j). \tag{8}$$

It is well known that the problem of computing the probability p that the system is functioning is NP-hard, cf. for example Ball [3]. The modular structure of a system may however help to simplify the computation. A subset of components, say for example $i = 1, \ldots, m$ $(m < n)$, is called a *module*, if there are Boolean functions ϕ' and ϕ'' such that

$$\phi(\mathbf{a}) \quad = \quad \phi'(\phi''(a_1, \ldots, a_m), a_{m+1}, \ldots, a_n). \tag{9}$$

ϕ'' is the structure function of the module. Suppose that the set of components $\{1, \ldots, n\}$ of a system decomposes into $m \geq 2$ modules $M_1, \ldots, M_m$. Let $\mathbf{a}_i$ denote the vector of the Boolean variables associated with the components in module M_i and let ϕ_i be the structure function of module M_i under the restriction that the variables in $\mathbf{a}_j$ are disjoint from those in $\mathbf{a}_k$ if $j \neq k$. Then there is a Boolean function ψ, such that

$$\phi(\mathbf{a}) \quad = \quad \psi(\phi_1(\mathbf{a}_1), \ldots, \phi_m(\mathbf{a}_m)). \tag{10}$$

$M_1, \ldots, M_m$ is called a *modular decomposition* of the system and ψ its *organizing structure*.

If we have a modular decomposition of a system ϕ, then we obtain

$$p \quad = \quad h_\phi(\mathbf{p}) \quad = \quad h_\psi(h_{\phi_1}(\mathbf{p}_1), \ldots, h_{\phi_m}(\mathbf{p}_m)), \tag{11}$$

where $\mathbf{p}_i$ is the vector of probabilities corresponding to $\mathbf{a}_i$.

This formula explains how a modular decomposition helps to compute p: the organizing function h_ψ as well as the modular functions h_{ϕ_i} have less, possibly much less, arguments as the original function h_ϕ. And this helpful property of a modular decomposition will be amplified, if the modules themselves possess their own modular decomposition.

This leads then to a hierarchical structure of modules over several levels. We represent this structure by a *tree* (see Fig. 1). The root node at level 0 corresponds to the Boolean system variable, denoted now by a_1^0. Its descendants on level 1 are the Boolean variables a_1^1 to $a_m^1 = a_{h_{1,1}}^1$. In general, a variable (node) a_j^i at level i has descendants $a_{k_{j,i}}^{i+1}$ to $a_{h_{j,i}}^{i+1}$ where $k_{1,i} = 1$ and $k_{j+1,i} = h_{j,i} + 1$. We denote the vector of Boolean variables of the

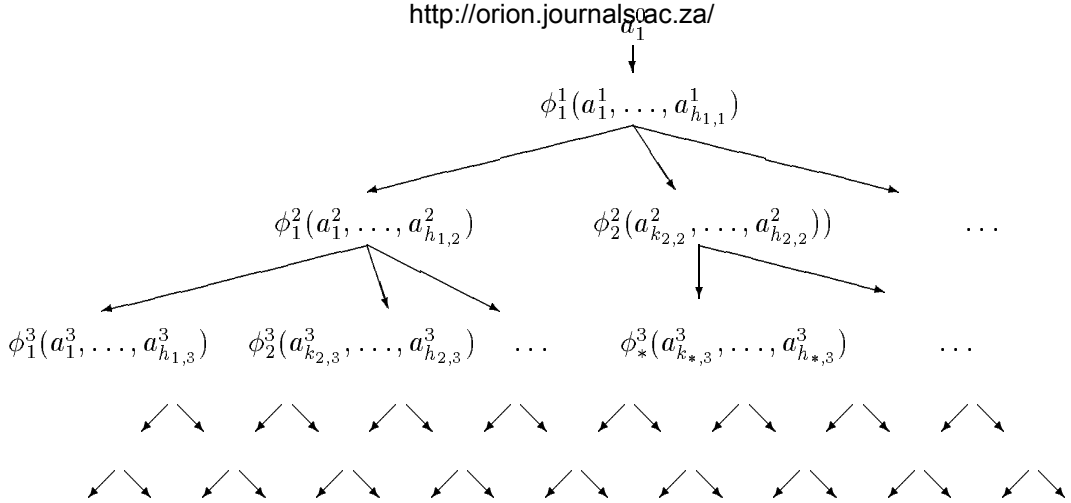

Figure 1: The hierarchy of Boolean functions is arranged in a tree.

descendants of a_j^i by $\mathbf{a}_j^i$. Associated with any node a_j^i of the tree, except the leaves with no descendants, there is a structure function ϕ_j^i such that

$$a_j^i \;=\; \phi_j^i(\mathbf{a}_j^i). \tag{12}$$

So, we have, starting at the root node,

$$a_1^0 \;=\; \phi_1^0(\mathbf{a}_1^0) \;=\; \phi_1^0(a_1^1,\ldots,a_{h_{1,1}}^1), \tag{13}$$

and

$$a_j^i \;=\; \phi_j^i(\mathbf{a}_j^i) \;=\; \phi_j^i(a_{k_{j,i}}^{i+1},\ldots,a_{h_{j,i}}^{i+1}), \tag{14}$$

etc. We denote the probability of a Boolean variable a_j^i by p_j^i and the probability function, associated with a structure function ϕ_j^i by h_j^i. Then we have also

$$p_j^i \;=\; h_j^i(\mathbf{p}_j^i). \tag{15}$$

Thus, we may start with the given component probabilities at the leaves of the tree and compute probabilities upwards in the tree, until we get the system reliability p_1^0 at the root,

$$p_1^0 \;=\; h_1^0(\mathbf{a}_1^0) \;=\; \phi_1^0(p_1^1,\ldots,p_{h_{1,1}}^1). \tag{16}$$

This supposes for example that we determine the paths of each structure function ϕ_j^i for each non-leave node of the tree and use an appropriate orthogonalization of the corresponding disjunctive normal form. Since — hopefully — in a modular hierarchie nodes have only a small number of descendants (components or modules) these computations for each node will be relatively small. We shall discuss in Section 4 an alternative approach to the reliability computation in a modular hierarchy.

3. DIAGNOSTIC OF MODULAR SYSTEMS

The reliability computation presented in the previous section corresponds to a more or

less classical pattern. But in this section we consider a new problem which seems so far hardly having been considered in the framework of modular system structures: the problem of diagnostic. Assume that we observe that the system is down. Then clearly some modules and components must be down. The question is which ones. Given only the observation that the system is down, it will in general not be possible to identify unambiguously the defect module(s) or component(s) which cause the system failure. But, by computing the posterior probabilities of module or component failures, given the event that the system is down, we may be able to point out those modules or components, which are more likely to cause the problem. Performing further tests on selected modules or components, we may identify the cause of the system failure with more and more certainty.

To start with, consider a system described by a structure function

$$a \;=\; \phi(a_1, \ldots, a_n) \tag{17}$$

and assume that the components a_i have probabilities p_i. But assume now, we observe that $a = \bot$, i.e. the system is down. This given event changes the prior probabilities $p_i = P\{a_i = \top\}$ into posterior, conditional probabilities $P\{a_i = \top | a = \bot\}$. How do we compute these posterior probabilities?

In fact, we could ask the same question, if we observe the system to be working, i.e. $a = \top$. So we consider more generally the problem of computing the family of conditional distributions $p(a_i|x)$ where $x = \top$ or $x = \bot$. Of course, we use Bayes' theorem. Assume first that $a = \top$

$$p'(a_i) \;=\; p(a_i|a) \;=\; \frac{p(a_i \wedge a)}{p(a)} \;=\; \frac{p(a_i \wedge \phi(a_1, \ldots, a_n))}{p(a)}. \tag{18}$$

The denumerator of this formula is the result of the reliability computation, which we assume to be done. How can the numerator then be computed? Assume that for the reliability computation, the structure function ϕ has been transformed into a disjoint disjunctive form

$$\phi \;=\; \bigvee_i c_i, \qquad c_i \wedge c_j = \bot \text{ for } i \neq j. \tag{19}$$

From this we deduce that

$$a_i \wedge \phi(a_1, \ldots, a_n) \;=\; \bigvee_i c_i', \tag{20}$$

where

$$c_i' \;=\; \begin{cases} c_i & \text{if } a_i \in c_i, \\ \bot & \text{if } \neg a_i \in c_i, \\ c_i \wedge a_i & \text{otherwise.} \end{cases} \tag{21}$$

Note that the c_i' are still disjoint. So, we have

$$p(a_i \wedge \phi(a_1, \ldots, a_n)) \quad = \quad \sum_i p(c_i') \tag{22}$$

and

$$p(c_i') \quad = \quad \begin{cases} p(c_i) & \text{if } a_i \in c_i, \\ 0 & \text{if } \neg a_i \in c_i, \\ p(c_i) \cdot p_i & \text{otherwise.} \end{cases} \tag{23}$$

For $a = \bot$, we have in the same way

$$p'(a_i) \quad = \quad p(a_i|\neg a) \quad = \quad \frac{p(a_i \wedge \neg\phi(a_1, \ldots, a_n))}{1 - p(a)}. \tag{24}$$

Here, the numerator is computed using the following identities:

$$\begin{aligned} p(a_i \wedge \neg\phi(a_1, \ldots, a_n)) &= p(a_i \wedge (\neg a_i \vee \neg\phi(a_1, \ldots, a_n))) \\ &= p(a_i \wedge \neg(a_i \wedge \phi(a_1, \ldots, a_n))) \\ &= p(a_i) - p(a_i \wedge \phi(a_1, \ldots, a_n)) \end{aligned} \tag{25}$$

The first term is the prior probability p_i and the second one has been computed above. So there is no need to compute new probabilities in this case.

If we consider now the more complicated case of a modular hierarchy, then the computations above permit to obtain the posterior distribution of the descendants of the root node. More generally, for any node a_j^i assume that we have the family of posteriors $p(a_j^i|x)$. How can we now obtain the posterior distribution for the descendants of a_j^i? Let a_k^{i+1} be such a descendant. Then, by the formula of total probability, we have

$$\begin{aligned} p'(a_k^{i+1}) &= p(a_k^{i+1}|x) \\ &= p(a_k^{i+1}|a_j^i)p(a_j^i|x) + p(a_k^{i+1}|\neg a_j^i)p(\neg a_j^i|x). \end{aligned} \tag{26}$$

The conditional probabilities $p(a_k^{i+1}|a_j^i)$ and $p(a_k^{i+1}|\neg a_j^i)$ are computed just as above in the case of a root node, using the reliabilities $p(a_j^i)$ which have been computed beforehand. If we work the tree downwards, then the probabilities $p(a_j^i|a)$ and $p(\neg a_j^i|a) = 1 - p(a_j^i|a)$ have already been computed on a higher, hence previous, level. In this way, we may work downwards down to the leaves of the tree to get the posteriors $p'(a_j^i)$ of all nodes of the tree.

In summary, we work first the tree of a modular hierarchy upwards to get the reliabilities of the nodes of the tree. Then, once we observe the system state, we work the tree downwards, using the results of the reliability computation, to obtain the posterior probabilities of the states of all modules and components. This exhibits nicely the duality inherent between reliability and diagnostic.

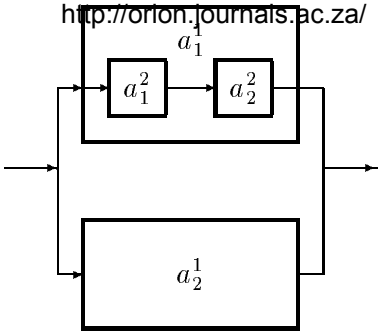

Figure 2: A serial module in a parallel organizing structure.

To illustrate this theory consider the example in Fig. 2. In this example the function ϕ_1^1 has two parameters a_1^1 and a_2^1. The first parameter depends of an other module $a_1^1 = \phi_1^2(a_1^2, a_2^2)$. The two Boolean functions are defined as

$$\phi_1^1(a_1^1, a_2^1) = a_1^1 \vee a_2^1, \qquad \phi_1^2(a_1^2, a_2^2) = a_1^2 \wedge a_2^2. \tag{27}$$

Assume the prior probabilities of a_2^1, a_1^2 and a_2^2 to be

$$p(a_2^1) = 0.7, \qquad p(a_1^2) = 0.8, \qquad p(a_2^2) = 0.6. \tag{28}$$

So in a first step the prior probabilities of a_1^1 and a_1^0 have to be calculated as

$$\begin{aligned} p(a_1^1) &= p(a_1^2 \wedge a_2^2) = p(a_1^2)p(a_2^2) = 0.8 * 0.6 = 0.48, \\ p(a_1^0) &= p(a_1^1 \vee a_2^1) = 1 - (1 - p(a_1^1))(1 - p(a_2^1)) \\ &= 1 - (1 - 0.48)(1 - 0.7) = 0.844. \end{aligned} \tag{29}$$

Now, if the variable a_1^0 is observed to be $\perp$, then its posterior probability changes to $p'(a_1^0) = 0$. And the posterior probabilities of all underlying variables change as well:

$$\begin{aligned} p'(a_1^1) &= p(a_1^1 \mid \neg a_1^0) = \frac{p(a_1^1 \wedge \neg a_1^0)}{1 - p(a_1^0)} = \frac{p(a_1^1 \wedge \neg \phi_1^1(a_1^1, a_2^1))}{1 - p(a_1^0)} \\ &= \frac{p(a_1^1) - p(a_1^1 \wedge \phi(a_1^1, a_2^1))}{1 - p(a_1^0)} = \frac{p(a_1^1) - p(a_1^1)}{1 - p(a_1^0)} = \frac{0}{0.156} = 0. \end{aligned}$$

In the same way we obtain $p'(a_2^1) = 0$. It is clear, that if a parallel system is down, then all its components must be down. The calculation of $p'(a_1^2)$ is done as follows:

$$\begin{aligned} p'(a_1^2) &= p(a_1^2 \mid a_1^1)p'(a_1^1) + p(a_1^2 \mid \neg a_1^1)p'(\neg a_1^1) = p(a_1^2 \mid \neg a_1^1) = \frac{p(a_1^2 \wedge \neg a_1^1)}{1 - p(a_1^1)} \\ &= \frac{p(a_1^2 \wedge \neg \phi_1^2(a_1^2, a_2^2)}{1 - p(a_1^1))} = \frac{p(a_1^2) - p(a_1^2 \wedge \phi_1^2(a_1^2, a_2^2))}{1 - p(a_1^1)} \\ &= \frac{p(a_1^2) - p(a_1^2 \wedge a_2^2)}{1 - p(a_1^1)} = \frac{0.8 - 0.8 \cdot 0.6}{1 - 0.48} = 0.667 \end{aligned}$$

To conclude this little example, we give all prior and posterior probabilities under the observation a_1^0 being $\perp$ and $\top$ respectively:

Variable	Prior	Posterior Probability	
		$a_1^0 = \perp$	$a_1^0 = \top$
a_1^0	0.7	0	1
a_1^1	0.48	0	0.569
a_2^1	0.7	0	0.829
a_1^2	0.8	0.667	0.834
a_2^2	0.6	0.25	0.668

(30)

We may now select some modules or components and test them. This will introduce additional information and therefore lead to revised posterior probabilities. In order to discuss these more general issues, we link modular hierarchies to a more general structure, namely Bayesian network. This will allow us to draw on a well developed computational theory and apply it to our particular problem.

4. MODULAR SYSTEMS AND BAYESIAN NETWORKS

The tree of a modular hierarchy can be considered as a Bayesian network. If we put the system into this framework, then we can use computational procedures developed for Bayesian networks (Jensen [9], Cowell et al. [7]). We shall study this approach in this section and compare it to the computational methods introduced in the previous sections.

In a tree of a modular hierarchy, each node corresponds either to the system (the root node), to a module or to a component (leaf nodes). We number the nodes of the tree from 0 (root) to m. For any node i of the tree, we denote the descendants of i (the elements of the module) by $D(i)$. For leaves i of course we have that $D(i) = \emptyset$. Fig. 3 shows the modular tree of the example in Section 3.

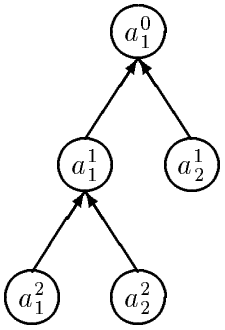

Figure 3: Modular Tree of the Example.

By $\mathbf{a}$ we denote the vector of all Boolean variables associated with the nodes (system, modules, components) of the tree. More generally, if J is a subset of nodes, then $\mathbf{a}_J$ denotes the vector of the Boolean variables a_j for $j \in J$. To every non-leaf node i corresponds a structure function $\phi_i(\mathbf{a}_{D(i)})$. In order to translate the structure into a

Bayesian network, we associate with this structure function a 0-1 conditional probability matrix,

$$p(a_i|\mathbf{a}_{D(i)}) = \begin{cases} 1 & \text{if } \phi_i(\mathbf{a}_{D(i)}) = \top, \\ 0 & \text{if } \phi_i(\mathbf{a}_{D(i)}) = \bot. \end{cases} \tag{31}$$

The prior probability $p_i = p(a_i)$ is defined for every leave i. The tree, together with these prior and conditional probabilities constitutes a Bayesian network. We refer to Cowell et al. [7] for details about the theory of Bayesian networks.

We sketch here the necessary elements in order to understand the application of Bayesian networks to the reliability and diagnostic of modular Boolean systems. First of all note that we may put $p(a_i) = p(a_i|D(i))$ for leaves, since $D(i)$ is empty in this case. With this convention, we can define the overall multivariate distribution of all variables in the tree by

$$p(\mathbf{a}) = \prod_i p(a_i|\mathbf{a}_{D(i)}), \tag{32}$$

where the product is to be taken over all nodes of the tree. The system reliability is then simply obtained from the marginal of this distribution with respect to the variable a_0,

$$p(a_0) = \sum_{a_i, i \neq 0} p(\mathbf{a}). \tag{33}$$

That is, we sum out all variables in $p(\mathbf{a})$, except a_0. Assume furthermore, that we observe now a certain variable a_i, say that $a_i = \bot$. Then we look at the conditional distribution of the other variables, given this event;

$$p(a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_m | a_i = \bot) = \frac{p(a_0, \ldots, a_{i-1}, \bot, a_{i+1}, \ldots, a_m)}{p(a_i = \bot)}. \tag{34}$$

Hence the conditional probability is *proportional* to $p(a_0, \ldots, a_{i-1}, \bot, a_{i+1}, \ldots, a_m)$ which is obtained from Equation (32) simply by putting the i-th variable a_i to the value $\bot$. In fact, the conditional probability distribution is computed from the family of probabilities $p(a_0, \ldots, a_{i-1}, \bot, a_{i+1}, \ldots, a_m)$ simply by normalizing it to 1. This is also true for any marginal of the conditional distribution: It can be obtained by normalizing the corresponding marginal of $p(a_0, \ldots, a_{i-1}, \bot, a_{i+1}, \ldots, a_m)$. Finally, we remark that

$$p(a_0, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_m | a_i = \bot) \sim p(a_0, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_m)\psi(a_i), \tag{35}$$

where ψ is defined by $\psi(\top) = 0$, and $\psi(\bot) = 1$

Based on these considerations, we may replace for computational purposes a probability distribution p in a Bayesian network by any non-negative function ψ, which is proportional to the distribution. Such non-negative functions are called *potentials*. According to (35), observed events can also be represented by potentials. So Bayesian networks reduce to a

calculus of potentials, which will be sketched in the sequel. A potential ψ refers always to some set of variables (nodes) J, i.e. ψ is a non-negative function of $\mathbf{a}_J$, and we define $d(\psi) = J$. We remark that a potential is just a multidimensional table of non-negative numbers. If $K \subseteq J$, then we denote the projection of the vector $\mathbf{a}_J$ by $\mathbf{a}_J^{\downarrow K}$. Also, if ψ is a potential on J, then its marginal with respect to $K \subseteq J$ is obtained by summing out the variables outside K,

$$\psi^{\downarrow K}(\mathbf{a}_K) \quad = \quad \sum_{\mathbf{a}_{J-K}} \psi(\mathbf{a}_K, \mathbf{a}_{J-K}). \tag{36}$$

So, given the Bayesian network related to a modular hierarchy together with, for example, some observations (tests) on certain modules or components, the problem consists of computing marginals relative to variables a_i of a product of potentials. For example for the system reliability, we compute

$$\left(\prod_i p(a_i | \mathbf{a}_{D(i)}) \right)^{\downarrow \{0\}}. \tag{37}$$

Suppose we observe then that $a_0 = \bot$, i.e. the system is down. Therefore, define $\psi_0(a_0)$ by $\psi_0(\top) = 0$ and $\psi_0(\bot) = 1$. Then we are interested in the marginals relative to some other variables a_i

$$\left(\prod_i p(a_i | \mathbf{a}_{D(i)}) \psi_0(a_0) \right)^{\downarrow \{i\}}. \tag{38}$$

This gives, up to normalization, the posterior probability of module or component i, given that the system is down. There may also be several observations, not only on the system as a whole, but on other modules. This leads to similar problems of the computation of marginals of some product of potentials.

Although it seems straightforward to compute the marginal of a product of potentials, in practice the product refers to 2^m different states and an explicit computation of such a product, followed by summing out all the variables not in the marginal, is not feasible. Therefore Lauritzen and Spiegelhalter [13] proposed a more realistic approach, based on graphical structures like Bayesian networks. Jensen et al. [10] introduced later a more efficient variant, which we shall present next for our modular hierarchy. For this purpose, we transform our tree, which represents the Bayesian network into another tree. We take the nodes i of the original tree, but add nodes, each of which represent a non-empty set $D(i)$. We introduce edges between i to $D(i)$ and $D(i)$ to every $j \in D(i)$. We add a node $\langle 0 \rangle$ and link it to the original root node 0; similarly, we we add a node $\langle j \rangle$ to every leave node j. Such a tree is called a *join* or *junction tree*. The nodes $D(i)$ together with the added nodes $\langle 0 \rangle$ and $\langle j \rangle$ for every leave node j form a set of nodes which we denote by $\mathcal{V}$, whereas the other nodes i (including the original 0) form a set of nodes $\mathcal{S}$. Note

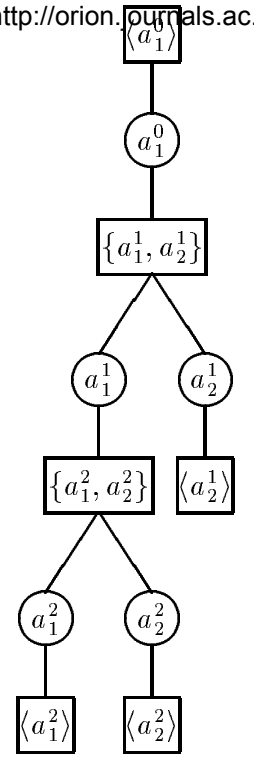

Figure 4: Join Tree of the Example. The Circles denote the Separators in $\mathcal{S}$.

that there is exactly one node $i \in \mathcal{S}$ between a pair $D(j)$ and $D(i)$ of nodes of $\mathcal{V}$ with $i \in D(j)$. The nodes of $\mathcal{S}$ are called separators.

Fig. 4 represents the join tree constructed from the modular tree in Fig. 3.

Assume now, in a general setting, that on every node $v = D(i)$ of $\mathcal{V}$ there is a potential ψ_v with $d(\psi_v) = D(i)$, and on every node i of $\mathcal{S}$ there is a potential ψ_i with $d(\psi_i) = \{i\}$. We make further the important assumption that, if node i is the separator between nodes v and w, then

$$\begin{aligned} \psi_i(a_i) = 0 \quad \text{implies} \quad & \psi_v(\mathbf{a}_v) = 0, \psi_w(\mathbf{a}_w) = 0, \\ & \text{whenever } \mathbf{a}_v^{\downarrow\{i\}} = \mathbf{a}_w^{\downarrow\{i\}} = a_i. \end{aligned} \tag{39}$$

We direct first the edges in the join tree towards the root $\langle 0 \rangle$. We pass the messages between nodes, starting with the leaves. Once a node $v = D(i)$ has obtained a message, it passes its message to its separator i, which passes its message to $w = D(j), i \in D(j)$. If ψ_v, ψ_w and ψ_i are the potentials on the corresponding nodes of the join tree before the messages are passed, and ψ_v^*, ψ_w^* and ψ_i^* the contents of the nodes after the messages are passed, then

$$\begin{aligned} \psi_v^*(\mathbf{a}_v) &= \psi_v(\mathbf{a}_v), \\ \psi_i^*(a_i) &= \psi_v^{\downarrow\{i\}}(a_i), \\ \psi_w^*(\mathbf{a}_w) &= \frac{\psi_w(\mathbf{a}_w)\psi_i^*(\mathbf{a}_w^{\downarrow\{i\}})}{\psi_i(\mathbf{a}_w^{\downarrow\{i\}})}. \end{aligned} \tag{40}$$

Note that $\psi_i(a_i) = \psi_i(\mathbf{a}_w^{\downarrow\{i\}})$ can be zero for some value of a_i. But, then by (39) the numerator vanishes also. In this case we fix the result arbitrary to, say, 0. In the sequel, we will write the formulas from (40) simply as follows:

$$\begin{aligned} \psi_v^* &= \psi_v, \\ \psi_i^* &= \psi_v^{\downarrow\{i\}}, \\ \psi_w^* &= \frac{\psi_w \psi_i^*}{\psi_i}. \end{aligned} \tag{41}$$

This message passing continues, until the root is reached. This is called the *collect phase*. It can be proved, that at the end of the collect phase, we have the following marginal in the root $\langle 0 \rangle$ (Jensen et al. [10]):

$$\psi_{\langle 0 \rangle}^* = \left(\frac{\prod_{v \in \mathcal{V}} \psi_v}{\prod_{i \in \mathcal{S}} \psi_i} \right)^{\downarrow\{\langle 0 \rangle\}}. \tag{42}$$

Usually we start with $\psi_v = p(a_i | \mathbf{a}_{D(i)})$ and unit tables ψ_i on the separators. Then the last result shows that we get the reliability $p(a_0)$ at the end of the collect phase on the root. This corresponds clearly to the reliability computation in a modular hierarchy, as presented in Section 2. Only, instead of computing with disjoint disjunctions for structure function, we use the tabular form of conditional probabilities and the corresponding multiplications and summations of tables. If this were all, the approach based on Bayesian networks would, in most cases, be computationally less efficient than the approach of Section 2. Bayesian networks become interesting however if diagnostic problems arise.

Assume then that an observation of the system state is made ($\top$ or $\bot$). We then add a corresponding potential $\psi'_{\langle 0 \rangle}$ to our product. That is, the content $\psi^*_{\langle 0 \rangle}$ of root node becomes

$$\psi^*_{\langle 0 \rangle} \psi'_{\langle 0 \rangle}. \tag{43}$$

Now, the edges are oriented away from the root. Then, starting with the root, messages are passed outwards towards the leaves. The message passing mechanism is exactly the same as in the collect phase, (41). This message passing scheme stops when all leaves have received their message. This is called the *distribute phase*. Jensen et al. [10] show that at the end of the distribute phase every node $v = D(i) \in \mathcal{V}$ and every node $i \in \mathcal{S}$ contains the marginal $\psi^{\downarrow D(i)}$ or $\psi^{\downarrow\{i\}}$ respectively, where

$$\psi = \frac{\prod_{v \in \mathcal{V}} \psi_v}{\prod_{i \in \mathcal{S}} \psi_i}. \tag{44}$$

In our case this product equals the common probability distribution, with a_0 instantiated to the observation, say

$$p(\bot, a_1, \ldots, a_m), \tag{45}$$

if $a_0 = \perp$ was observed. This shows that $\psi^{\downarrow i}$ is then, up to normalization the posterior distribution of the state of module or component i, given the system observation.

This distribute phase corresponds clearly to the diagnostic computation discussed in Section 3. Now the Bayesian networks become interesting. We may introduce further observations of other modules or components besides the one of the whole system already introduced. It suffices to take the corresponding node as a new root, direct all edges of the join tree outwards, i.e. away from the new root, and add a new distribute phase based on the actual contents of the node of the join tree, cf. Jensen et al. [10] and Kohlas [12]. This will give the updated posterior probabilities for all other modules and components. This incremental procedure helps to identify the cause of a system breakdown by computing posterior probabilities, selecting modules to test on the base of these posterior probabilities and updating the posterior probabilities by a distribute phase. This can be repeated until the faulty elements are identified with sufficient certainty.

5. CONCLUSION

The computation of the reliability of a modular system is a classical problem of reliability theory. But there is the dual problem of diagnostic: if the failure of the system or of some of its modules is observed, what can then be the possible cause of this observation? Which submodules or components are down? Or also: how does this change the reliability of the system? We show in this paper that the modular structure of the system helps also to answer these questions. In fact, it can be used to compute the posterior probabilities of the submodules and the components given some observation. This calculation uses the results of the previous reliability computation. The duality of reliability and diagnostic becomes most clear, if we consider a modular hierarchy as a particular Bayesian network. We then get a unifying framework, in which we may compute reliability and diagnostic information using any information or observation about module states we may obtain. In fact, it is possible to use incremental procedures, where one observation at a time is added. The new updated posterior probabilities may be used to decide which modules to test next.

The unifying framework of Bayesian networks shows that reliability and diagnostic of modular systems are two sides of the same coin. It offers well developed computation procedures. The Bayesian network of a modular hierarchy has however a very special form. This particular structure may help to adapt the Bayesian methods and to render them more efficient in this particular case. This issue is still open. So are many other issues related to the application of Bayesian networks in the study of modular systems. For example the time dependency of the availability and reliability, related to aging, and other features could be integrated into this framework. Also, the duality between reliability and diagnostic can be explored in more general structures than modular system.

This is the subject of model-based reliability, which extends the well-known approach of model-based diagnostic.

References

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

[2] B. Anrig and F. Beichelt. Disjoint sum forms in reliability theory. *South African J. on Operations Research ORiON*, 2001.

[3] M. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, 35:230–239, 1986.

[4] F. Beichelt. *Zuverlässigkeits- und Instandhaltungstheorie.* Teubner, Stuttgart, 1993.

[5] Z.W. Birnbaum and J.D. Esary. Modules of coherent binary systems. *J. of the Society for Industrial and Applied Mathematics*, 13(2):444–462, 1965.

[6] Z.W. Birnbaum, J.D. Esary, and S.C. Saunders. Multicomponent systems and structures, and their reliability. *Technometrics*, 3:55–77, 1961.

[7] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer, 1999.

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

[9] F.V. Jensen. *An Introduction to Bayesian Networks.* UCL Press, London, 1996.

[10] F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Comp. Stat. Q.*, 4:269–282, 1990.

[11] J. Kohlas. *Zuverlässigkeit und Verfügbarkeit.* Teubner, 1987.

[12] J. Kohlas. Valuation algebras: Generic architecture for reasoning. draft, 2001.

[13] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. of Royal Stat. Soc.*, 50(2):157–224, 1988.

[14] G. Provan. An integration of model-based diagnosis and reliability theory. In A. Darwiche and G. Provan, editors, *DX'00, Eleventh Intl. Workshop on Principles of Diagnosis, Morelia, Mexico*, pages 193–200, 2000.