Model-Based Reliability and Diagnostic: A Common Framework for Reliability and Diagnostics *

Bernhard Anrig and Jürg Kohlas
Department of Informatics, University of Fribourg,
CH–1700 Fribourg, Switzerland
{Bernhard.Anrig,Juerg.Kohlas}@unifr.ch

Abstract. Technical systems are in general not guaranteed to work correctly. They are more or less reliable. One main problem for technical systems is the computation of the reliability of a system. A second main problem is the problem of diagnostic. In fact, these problems are in some sense dual to each other.

In this paper, we will use the concept of probabilistic argumentation systems PAS for modeling the system description as well as observation and specifications of behaviour in one common framework. We show that PAS are a framework which allows to formulate both main problems easily and all concepts for these two problems can clearly be defined therein. Using PAS, reliability and diagnostic can be considered as dual problems. PAS allows to consider one common strategy for computing answers to the questions in the different situations.

1 Introduction and Overview

One main problem for technical systems is the computation of the reliability of a system. This is studied in reliability theory (see for example [7, 8]). The reliability depends on various factors like the quality and the age of components, complexity of the system, etc. The reliability of a system conveys some information about the behavior of the system in the future, based on information about the components, for example probabilistic information about the reliability over time.

A second main problem for technical systems is the problem of diagnostic. Here, the problem is to explain the behavior of the system, usually based on measurements and observations of some parts of the system, together with the system description in some framework. The actual observations and the description of the system are the only ingredients for the computation of the diagnoses. Additionally, if probabilistic knowledge is available about the different operating modes of the components, then the likelihood of the system states can be defined and prior as well as posterior probabilities can be computed for the set of possible system states.

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

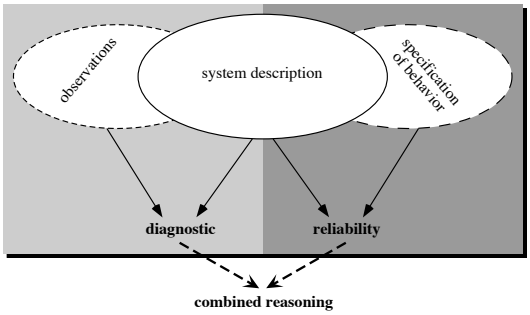

Figure 1. Reliability versus Diagnostic.

The two main problems depend both on a formalization of the system in some framework together with either observations, measurements, or requirements (Fig. 1). Here, we will use the concept of probabilistic argumentation systems PAS for modeling the system description as well as observation and specifications of behaviour in one common framework. The goal of a PAS is to derive arguments in favor and against the hypothesis of interest. An argument is a defeasible proof built on uncertain assumptions, i.e. a chain of deductions based on assumptions that makes the hypothesis true. If probabilistic information is available, a quantitative judgement of the situation is obtained by considering the probabilities that the arguments are valid. The resulting degrees of support and possibility correspond to belief and plausibility, respectively, in the Dempster-Shafer theory of evidence [24, 20]. In fact, PAS combines the strengths of logic and probability in one framework. In this paper we show that probabilistic argumentation systems are a framework which allows to formulate both main problems, i.e. reliability and diagnostic, easily and all concepts therefore can clearly be defined therein. The framework will especially allow to consider one common strategy for computing answers to the questions in the different situations. Some work in this direction but without using PAS has been done by Provan [22].

The main information for both problems is the description of the system in some formalism; we will focus here on a for-

malization using logic. In the case of reliability, we may have a specification which describes the goals which have to be fulfilled by the system. This information will be used to compute the structure function from the system description. Different specifications may lead to different structure functions. Even in the absence of an explicit specification of a reliability requirement, we may deduce a structure function by assuming that the system should be functioning at least if all components are working.

On the other hand, in the case of diagnostic, some observations of the system may indicate that the system is not working as it is supposed to be. This information — together with the system description — allows then to compute the diagnoses of the system, i.e. minimal sets of components whose malfunctioning "explains" the wrong behaviour of the whole system.

2 Reliability

2.1 Combinatorial Reliability

In binary combinatorial reliability, a system is assumed to be composed of a number of different components. Each component is either intact or it is down, and so is the whole system itself, depending on the states of its components. In order to formulate this, binary variables x_i are associated to components $i = 1, 2, \ldots, n$ of the system, where $x_i = \top$ if the component number i works and $x_i = \bot$ otherwise. Let $\mathbf{x}$ be the vector $(x_1, x_2, \ldots, x_n)$ of the component states. This state-vector has 2^n possible values. These values can be decomposed into two disjoint subsets, the set $S_\top$ of working states, for which the system as a whole is assumed to be functioning, and the set $S_\bot$ of down-states, for which the system is supposed to not work properly. The corresponding system state is denoted by x. Its dependence on the state-vector $\mathbf{x}$ is described by a Boolean function ϕ, defined as

$$x \quad = \quad \phi(\mathbf{x}) \quad = \quad \begin{cases} \top & \text{if } \mathbf{x} \in S_\top, \\ \bot & \text{if } \mathbf{x} \in S_\bot. \end{cases} \tag{1}$$

The Boolean function ϕ is called the *structure function* of the system. In combinatorial reliability it is assumed to be given and it forms the base for reliability analysis.

The structure function ϕ is usually assumed to be *monotone*. That is, if $\mathbf{x}_1 \leq \mathbf{x}_2$, then $\phi(\mathbf{x}_1) \leq \phi(\mathbf{x}_2)$. For a monotone structure function, a subset $P \subseteq \{1, 2, \ldots, n\}$ of components is called a *path*, if $\phi(\mathbf{x}) = \top$ for all state-vectors $\mathbf{x}$ for which the components of the set P are working, $x_i = \top$ for all $i \in P$. That is, the elements of a path are sufficient to guarantee the functioning of the system, regardless of the state of the components outside the path. We assume that the set $\{1, 2, \ldots, n\}$ of all components is a path (otherwise the system would never be functioning). A path P is called *minimal*, if no proper subset of P is still a path. Since the paths are upwards closed it is sufficient to know all minimal paths. Let $\mathcal{P}$ denote the set of minimal paths. This set determines the structure function,

$$\phi(\mathbf{x}) = \bigvee_{P \in \mathcal{P}} \bigwedge_{i \in P} x_i. \tag{2}$$

This logical formula expresses the fact, the system is working, if all components of at least one minimal path are working.

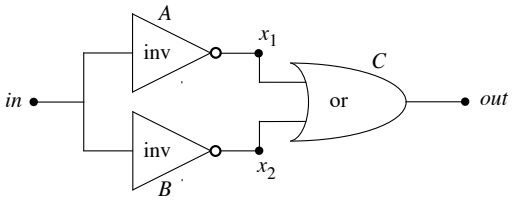

Figure 2. A simple device

Dually, the notion of a *cut* is defined and $\mathcal{C}$ denotes the set of all minimal cuts.

If for every component $i = 1, 2, \ldots, n$ its respective probability p_i of functioning correctly is defined, then the probability that the system is functioning can be computed (assuming the components to be stochastically independent). In fact, $\phi(\mathbf{x})$ is a random variable, and the probability p that the system is functioning is

$$p = E(\phi(\mathbf{x})) = h(\mathbf{p}). \tag{3}$$

Here, $\mathbf{p}$ denotes the vector $(p_1, p_2, \ldots, p_n)$ of probabilities. $h(\mathbf{p})$ is called the reliability function and its computation is a nontrivial task [1, 16, 5].

2.2 Model-Based Reliability

The structure function describes the conditions under which a system is functioning, depending on the states of its components. It is already a compilation of knowledge about the system and its structure. In this section we shall illustrate another approach, where a more physical description of a system is given. Additionally, a specification of the desired behavior of the system is given. These two elements will then allow the deduction of a structure function and its associated reliability function. The discussion in this section will be informal.

Example 1: Detector of Power Failure
(Example adapted from [22])
Consider a simple device which watches a Boolean value *in* and reports an output *out* equal to $\top$, if the value vanishes (becomes $\bot$). A simple version of such a device is depicted in Figure 2. The functionality of this device can be described with propositional logic. Let *in* and *out* be the variables which denote the state of the in- and output respectively. Both variables are binary, i.e. represent the boolean values true or false respectively. Further, there are two internal variables x_1 and x_2, also binary. For every component A, B or C, there is a respective binary variable ok_A, ok_B, and ok_C which describes the working mode of the component.

Consider the inverter A: if it works correctly (ok_A is true), then its input is the negation of its output, *out* is true if and only *in* is false. We express this by the formula $in \leftrightarrow \neg x_1$. So the entire information is modeled as the logical implication $ok_A \rightarrow (in \leftrightarrow \neg x_1)$. Note that so far nothing is said about the behavior of the component, if it is down (ok_A is false). There are several possibilities. One is that in this case the output of the component is always false, i.e. $\neg ok_A \rightarrow \neg x_1$.

For the component B, the same specification can be applied. For the or-gate, if it works correctly, then the output is

true if at least one of its inputs is. So the whole information about the device is modeled by a set of six implications:

$$\Sigma = \left\{ \begin{array}{ll} ok_A \rightarrow (in \leftrightarrow \neg x_1), & \neg ok_A \rightarrow \neg x_1, \\ ok_B \rightarrow (in \leftrightarrow \neg x_2), & \neg ok_B \rightarrow \neg x_2, \\ ok_C \rightarrow (out \leftrightarrow x_1 \vee x_2), & \neg ok_C \rightarrow \neg out \end{array} \right\} \quad (4)$$

This is the *system description*. We add now a specification of what we expect from the system to this physical description of the system. We expect, that negative (false) input is detected, i.e. the output is true. This could be expressed by $\neg in \rightarrow out$. However, this is a weak requirement. It does not exclude that *out* becomes true, even if *in* is true. More stringent would be the specification $\neg in \leftrightarrow out$. This asks that there is an alarm (*out*) if, and only if, *in* is false.

We may now ask under which states, described by the variables ok_A, ok_B, and ok_C, each one of these specifications is fulfilled. This defines the *structure function* of the system associated with the corresponding specification of desired system behavior. We shall see in the next section, that it is a well-defined problem of propositional logic to deduce these structure functions from the system description and the specifications of desired behavior. $\ominus$

This example shows how the physical behavior of systems and the required behavior can be described in the language of propositional logic. We shall examine this structure in the following section in a general context.

3 Probabilistic Argumentation Systems

Probabilistic argumentation systems have been developed as general formalisms for expressing uncertain and partial knowledge and information in artificial intelligence. They combine in an original way logic and probability. Logic is used to derive arguments and probability serves to compute the reliability or likelihood of these arguments. These systems can be used for model-based diagnostics as has been demonstrated in [2, 19]. Here we shall show how they relate to reliability theory.

In this section we give a short introduction into propositional probabilistic argumentation systems. For a more detailed presentation of the subject we refer to [15]. We remark also that such systems have been implemented in a system called ABEL which is available on the internet (cf. [14] for further information).

3.1 Propositional Logic

Propositional logic deals with declarative statements, called called *propositions*, that can be either true or false. Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositions. The symbols $p_i \in P$ together with $\top$ (tautology) and $\bot$ (falsity), are called *atoms*. Compound formulas are built by the following syntactic rules:

- atoms;
- if γ is a formula, then $\neg\gamma$ is a formula;
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$, and $(\gamma \leftrightarrow \delta)$ are formulas.

By assigning priority in decreasing ordering to $\neg$, $\wedge$, $\vee$, $\rightarrow$, some parentheses can be eliminated. The set $\mathcal{L}_P$ of all formulas generated by the above recursive rules is called *propositional language* over P.

A literal is either an atom p_i or the negation of an atom $\neg p_i$. A *term* is either $\top$ or a conjunction of literals where every atom occurs at most once (but none of $\bot$ and $\top$), and a *clause* is either $\bot$ or a disjunction of literals where every atom occurs at most once (but none of $\bot$ and $\top$). $C_P \subseteq \mathcal{L}_P$ denotes the set of all terms, and D_P the set of all clauses.

$N_P = \{0,1\}^n$ denotes the set of all 2^n different interpretations for P. If $\gamma \in \mathcal{L}_P$ evaluates to 1 under $\mathbf{x} \in N_P$, then $\mathbf{x}$ is called a *model* of γ. The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$.

A propositional sentence γ *entails* another sentence δ (denoted by $\gamma \models \delta$) if and only if $N_P(\gamma) \subseteq N_P(\delta)$. Sometimes, it is convenient to write $\mathbf{x} \models \gamma$ instead of $\mathbf{x} \in N_P(\gamma)$. Also we write $\gamma \models \bot$ if γ is not satisfiable. Furthermore, two sentences γ and δ are *logically equivalent* (denoted by $\gamma \equiv \delta$), if and only if $N_P(\gamma) = N_P(\delta)$.

3.2 Basic Concepts of Argumentation Systems

Consider two finite sets $P = \{p_1, \ldots, p_m\}$ and $A = \{a_1, \ldots, a_n\}$ of propositional variables with $A \cap P = \emptyset$, the elements of P are called propositions, the elements of A assumptions. We consider a fixed set of formulas $\Sigma \subseteq \mathcal{L}_{A \cup P}$ called the *knowledge base*, which models the information available; sets of formulas are interpreted conjunctively, i.e. $\Sigma = \bigwedge\{\xi \in \Sigma\}$. We assume that this knowledge base is satisfiable. A triple (Σ, A, P) is called a *propositional argumentation system PAS*.

The elements of N_A are called *scenarios* (or *system states*). A scenario represents a specification of all values of the assumptions in A. Define now:

Inconsistent Scenarios: $CS_A(\Sigma) := \{\mathbf{s} \in N_A : \mathbf{s}, \Sigma \models \bot\}$,

Quasi-Supporting Scenarios of $h \in \mathcal{L}_N$**:**
$$QS_A(h, \Sigma) := \{\mathbf{s} \in N_A : \mathbf{s}, \Sigma \models h\},$$

Supporting Scenarios of $h \in \mathcal{L}_N$**:**
$$SP_A(h, \Sigma) := QS_A(h, \Sigma) - CS_A(\Sigma),$$

Possible Scenarios for $h \in \mathcal{L}_N$**:**
$$PL_A(h, \Sigma) := SP_A{}^c(\neg h, \Sigma).$$

Inconsistent scenarios are in contradiction with the knowledge base and therefore to be considered as excluded by the knowledge. Supporting scenarios for a formula h are scenarios, which, together with the knowledge base imply h and are consistent with the knowledge. So, under supporting scenarios, the hypothesis h is true. Possible scenarios for h are scenarios, which do not imply $\neg h$ and thereby do not refute h. Quasi-supporting scenarios for h are the union of supporting scenarios and inconsistent scenarios.

Scenarios are the basic concepts of assumption-based reasoning. However, sets of inconsistent, quasi-supporting, supporting and possible scenarios may become very large. Therefore, more economical, logical representations of these sets are needed. For this purpose, the following concepts are defined:

Set of Supporting Argument for h**:**
$$SP(h, \Sigma) = \{\alpha \in C_A : N_A(\alpha) \subseteq SP_A(h, \Sigma)\},$$

The sets of quasi-supporting and of possible arguments are defined analogously. Remark that supporting arguments are similar to paths for structure functions in reliability theory. This similarity will be exploited later. These sets are

all upward closed. Hence the sets of arguments are already determined by their *minimal* elements. We denote by $\mu QS(h,\Sigma)$, $\mu SP(h,\Sigma)$ and $\mu PL(h,\Sigma)$ the sets of minimal quasi-supporting, supporting and possible arguments. Further,

Conflict: $conf(\Sigma) := \bigvee_{\alpha\in\mu QS(\bot,\Sigma)} \alpha,$

Support of h**:** $sp(h,\Sigma) := \bigvee_{\alpha\in\mu SP(h,\Sigma)} \alpha,$

Quasi-support $qs(h,\Sigma)$ and possibility $pl(h,\Sigma)$ are defined analogously. Clearly, any formula which is logically equivalent to logical representations above can be used as a representation.

Example 2: (Cont. of Example 1)
The information of Example 1 is modeled in an argumentation system as follows: $A = \{ok_A, ok_B, ok_C\}$, $P = \{in, x_1, x_2, out\}$ and Σ as in (4). There are no inconsistent scenarios and for $h = \neg in \rightarrow out$ we have $QS_A(h,\Sigma) = \{(0,1,1),(1,0,1),(1,1,1)\}$ and $PL_A(h,\Sigma) = N_A$. As $CS_A(\Sigma) = \emptyset$, we have $QS_A = SP_A$ in this situation and there are some arguments in favor of the hypothesis, but none against it. Hence, $qs(h,\Sigma) = (ok_A \wedge ok_C) \vee (ok_B \wedge ok_C)$ and $pl(h,\Sigma) = \top$. $\ominus$

3.3 Probabilistic Information

On top of the structure of a propositional argumentation systems, we may easily add a probability structure. Assume that there is a probability $p(a_i) = p_i$ for every assumption $a_i \in A$ given. Assuming stochastic independence between assumptions, a scenario $\mathbf{s} = (s_1, \ldots, s_n)$ gets the probability

$$p(\mathbf{s}) \quad = \quad \prod_{i=1}^{n} p_i^{s_i}(1-p_i)^{1-s_i}. \tag{5}$$

This induces a probability measure p on $\mathcal{L}_A$,

$$p(f) = \sum_{\mathbf{s}\in N_A(f)} p(\mathbf{s})$$

for $f \in \mathcal{L}_A$. A quadruple (Σ, A, P, Π) with $\Pi = (p_1, \ldots, p_n)$ is then called a *probabilistic (propositional) argumentation system* PAS.

The problem of computing the probability $p(f)$ is similar to the problem of computing the reliability of a structure function, except, that monotonicity cannot be assumed in general; for algorithms for efficiently computing the probability $p(f)$ see [20, 9, 13].

Once we have such a probability structure on top of a propositional argumentation system, we can exploit it to compute likelihoods (or in fact, reliabilities) of supporting and possible arguments for hypothese h. First, we note, that Σ imposes that we eliminate the inconsistent scenarios and condition the probability on the consistent ones. In other words, Σ is an event that restricts the possible scenarios to the set $N_A - CS_A(\Sigma)$, hence their probability has to be conditioned on the event Σ. This conditional probability is defined by

$$p'(\mathbf{s}) = \frac{p(\mathbf{s})}{1 - p(qs(\bot,\Sigma))}.$$

for consistent scenarios $\mathbf{s}$. $p(qs(h,\Sigma)) = dqs(h)$ is the so-called degree of quasi-support for h. Now, the degree of support dsp for hypotheses h is defined by

$$dsp(h) = p'(sp(h,\Sigma)) = \frac{dqs(h,\Sigma) - dqs(\bot,\Sigma)}{1 - dqs(\bot,\Sigma)}.$$

This result explains the importance of quasi-support. It is sufficient to compute degrees of quasi-supports. Further, we obtain the degree of plausibility of h,

$$dpl(h) = p'(pl(h,\Sigma)) = \frac{1 - dqs(\neg h,\Sigma)}{1 - dqs(\bot,\Sigma)} = 1 - dsp(\neg h).$$

Degree of quasi-support $dqs(h)$ and of support $dsp(h)$ correspond in fact to unnormalized and normalized belief in the Dempster-Shafer theory of evidence [24, 20, 15].

3.4 Computational Theory

Computing quasi-supports is the basic operation in PAS. It can be based on resolution and variable elimination (forgetting) [15, 12, 13]. In the sequel, we will sketch some of the main concepts for computation.

First, note that the computation of $qs(h)$ can be reduced to the computation of the conflicts with respect to an updated knowledge base: $qs(h,\Sigma) = qs(\bot, \Sigma \cup \{\neg h\})$. So for any hypothesis h, the quasi-supporting arguments $qs(h,\Sigma)$ can be determined by computing the conflicts with respect to the knowledge base $\Sigma \cup \{\neg h\}$. Hence in the sequel, we focus on the computation of the conflicts with respect to a general knowledge base.

The ideas presented in the sequel are based on representations of knowledge in conjunctive normal form (CNF), i.e. a conjunction of clauses. The main step is based on the principle of resolution. Let $x \in A \cup P$. A disjoint decomposition of Σ is then defined as follows:

$$\begin{aligned}
\Sigma_x^+ &= \{\xi \in \Sigma : x \in Lit(\xi)\} \\
\Sigma_x^- &= \{\xi \in \Sigma : \neg x \in Lit(\xi)\} \\
\Sigma_x^0 &= \{\xi \in \Sigma : x \notin Lit(\xi) \text{ and } \neg x \notin Lit(\xi)\}
\end{aligned}$$

$Lit(\Sigma)$ denotes the set of all literals occurring in Σ. A literal is either a (positive) atom or a negated atom.

Consider two clauses $\xi^+ = x \vee \delta^+$ and $\xi^- = \neg x \vee \delta^-$ in Σ_x^+ and Σ_x^- respectively. The clause $\rho(\xi^+, \xi^-) = \delta^+ \vee \delta^-$ is called the resolvent; note that we simplify implicitly the resolvent so that $\rho(\xi^+, \xi^-)$ is again a clause, i.e. double occurrences of atoms etc. are simplified.

Eliminating a variable $x \in P \cup A$ from Σ means now to compute

$$Elim_x(\Sigma) = \mu(\Sigma_x^0 \cup \{\rho(\xi^+, \xi^-) : \xi^+ \in \Sigma_x^+,\ \xi^- \in \Sigma_x^-\})$$

Consider a set $Q \subseteq P \cup A$. We define now, for $Q = \{q_1, \ldots, q_r\}$,

$$Elim_Q(\Sigma) = Elim_{q_r}(\ldots(Elim_{q_2}(Elim_{q_1}(\Sigma)))\ldots)$$

The result does not depend on the very order of the elimination of atoms; yet note that the computations depend *critically* on a "good" ordering, see [15] for a discussion as well as relations to the theory of local computation (in the sense of Shenoy & Shafer [25]).

This allows then to compute the quasi-supporting arguments of a knowledge base Σ as follows:

Theorem 1 ([15])

$$QS_A(h,\Sigma) = N_A^c(Elim_P(\Sigma \cup \{\neg h\}))$$

In other words, this theorem asserts that

$$qs(h,\Sigma) \equiv \neg \bigwedge Elim_P(\Sigma \cup \{\neg h\}).$$

The concept of elimination allows to compute quasi-supporting and therefore also supporting as well as possible arguments for hypotheses. This notation connects the concepts presented here to the more general theory of valuation algebras, a general theory for representing, combining and focusing pieces of information [18, 21].

4 Reliability Analysis Using Probabilistic Argumentation Systems

4.1 Reliability based on Requirement Specification

We discuss now how probabilistic argumentation systems can be used to formulate and solve reliability problem. The basic idea is simple: The system behavior is described in terms of the states of its components. In addition the desired or required behavior of the system is specified. The system description forms a probabilistic argumentation system. The question is then: how likely (probable) is it, that the specified requirement is satisfied? In order to answer this question, the specification of required behavior is taken as a hypothesis. The *support* of this specification determines then essentially the structure function of this reliability problem, and the degree of support of the specified requirement is the reliability of the system with respect to the required behavior. Note that — depending on different goals a system should attain, or services it should provide — different requirements may be formulated. So the corresponding reliability analysis has to be differentiated, but can be carried out within the same framework of probabilistic argumentation systems.

Example 3: (Cont. of Example 1)
We have already formulated Σ and two different specifications $\delta_1 = \neg in \rightarrow out$ and $\delta_2 = \neg in \leftrightarrow out$. We can compute the supports of these two specifications. It turns out, that both are the same,

$$sp(\delta_1,\Sigma) = sp(\delta_2,\Sigma) = (ok_A \wedge ok_C) \vee (ok_B \wedge ok_C).$$

Note that this is just the path representation of the expected structure function. In fact this structure function could be reformulated as $(ok_A \vee ok_B) \wedge ok_C$, which shows that it is a series system composed of component C and a parallel module of the components A and B. The remarkable fact is, that this structure function has been automatically deduced from the system description and the specification of requirements.

The system description is an essential element for this analysis. If it is changed, then this may influence the results of the analysis. Suppose that, in contrast to the model above, we do not know how the faulty components behave. The knowledge base becomes now

$$\Sigma' = \left\{ \begin{array}{l} ok_A \rightarrow (in \leftrightarrow \neg x_1),\ ok_B \rightarrow (in \leftrightarrow \neg x_2), \\ ok_C \rightarrow (out \leftrightarrow x_1 \vee x_2). \end{array} \right\}$$

With this less complete model, the structure function of the two specifications above become different,

$$\begin{array}{rcl} sp(\delta_1,\Sigma') & = & (ok_A \wedge ok_C) \vee (ok_B \wedge ok_C), \\ sp(\delta_2,\Sigma') & = & ok_A \wedge ok_B \wedge ok_C. \end{array}$$

Now, the stronger requirement δ_2 can only be guaranteed if all three components work correctly (a series system), whereas the weaker one still has the same redundancy as before. $\ominus$

In the general case, we have a PAS (Σ, A, P), where the assumable symbols in A correspond to the components of the system. Positive assumptions correspond to working components. Accordingly in the context of reliability analysis, we shall call the scenarios of this argumentation system *system states*. The propositional symbols in P are needed to describe the system behavior. We assume that the system description Σ excludes no system states, that is there are no conflicts, $QS_A(\bot,\Sigma) = \emptyset$. A knowledge base Σ which satisfies this is called A-consistent.

The required behavior is specified by a formula δ. Usually δ will not contain assumptions, but there is no reason to exclude this in general. δ formulates a reliability goal. There may be several such goals.

The set of system states $SP_A(\delta,\Sigma)$ supporting δ contains all states guaranteeing the required specification from the system description. Its complement $SP_A{}^c(\delta,\Sigma) = PL_A(\neg\delta,\Sigma)$ contains the system states where this guarantee is no more assured. These are the unreliable states. So $SP_A(\delta,\Sigma)$ defines the *structure function* associated with the specification δ

$$s \;=\; \phi_{\delta,\Sigma}(\mathbf{s}) \;=\; \left\{ \begin{array}{ll} \top & \text{if } \mathbf{s} \in SP_A(\delta,\Sigma), \\ \bot & \text{if } \mathbf{s} \notin SP_A(\delta,\Sigma). \end{array} \right. \tag{6}$$

The index Σ in $\phi_{\delta,\Sigma}$ will be omitted if Σ is clear from the context. Here, s denotes the "system state", which is $\top$, when the reliability specification is assured and $\bot$ otherwise. Since the set $SP_A(\delta,\Sigma)$ has a logical representation based on minimal arguments, the same holds for the structure function ϕ_δ,

$$\phi_\delta = \bigvee_{\alpha \in \mu SP(\delta,\Sigma)} \alpha \quad = sp(\delta,\Sigma) \tag{7}$$

In the same way, based on minimal possible arguments $PL(\neg\delta,\Sigma)$, we obtain

$$\neg\phi_\delta = \bigvee_{\beta \in \mu PL(\neg\delta,\Sigma)} \beta \quad = pl(\neg\delta,\Sigma).$$

By de Morgan laws this transforms into

$$\phi_\delta = \bigwedge_{\beta \in \mu PL(\neg\delta,\Sigma)} \neg\beta. \tag{8}$$

Note that $\neg\beta$, the negation of a term, is a clause. This is a second logical representation of ϕ_δ.

A comparison with the minimal path and minimal cut representation of monotone structure functions (2) shows that minimal supporting arguments α for δ and minimal possible arguments β for $\neg\delta$ play a role similar to minimal paths and minimal cuts.

According to our assumption of A-consistency, we have $QS_A(\bot,\Sigma) = \emptyset$. Thus

$$SP_A(\delta,\Sigma) = QS_A(\bot, \Sigma \cup \{\neg\delta\}). \tag{9}$$

On the other hand, we have also

$$PL_A(\neg\delta, \Sigma) = QS_A{}^c(\bot, \Sigma \cup \{\neg\delta\}). \quad (10)$$

This shows, that a reliability analysis of a system Σ relative to a requirement specification δ, requires essentially the computation of the conflict states $QS_A(\bot, \Sigma \cup \{\neg\delta\})$. We shall see below, that this is exactly also what is required for diagnosis. This is a first hint to the duality between the problems of reliability and diagnosis.

Once probabilities for the assumptions, i.e. component availabilities or reliabilities are defined, system reliability relative to a specification δ is simply the degree of support of δ, (since $QS_A(\bot, \Sigma) = \emptyset$), i.e.

$$p_{\delta,\Sigma} = dsp(\delta, \Sigma) = dqs(\delta, \Sigma) = p(QS_A(\bot, \Sigma \cup \{\neg\delta\})).$$

4.2 Implicitly Defined Reliability

A specification δ is called *consistent* with the system description Σ, if the system state $\mathbf{1}$ belongs to $SP_A(\delta, \Sigma)$. In this section we only consider specifications consistent with the system description.

A system description Σ often contains, besides assumptions, another set O of special propositional atoms, namely those which are *observable*. Then specifications δ can be assumed to be formulated with observables only, $\delta \in \mathcal{L}_O$. Observables are typically input and output variables of some system.

Assume now, that in a system description (Σ, P, A) a set of observable variables O is singled out. Usually, $O \subseteq P$, i.e. component states can not be observed directly. But it does no harm to assume more generally $O \subseteq P \cup A$. Then we define an implicit specification

$$\hat{\delta} = Elim_{(A\cup P)-O}(\Sigma \cup \{a_1 \wedge a_2 \wedge \cdots \wedge a_n\}).$$

That is, $\hat{\delta}$ represents all the functionality of the system in terms of observables which can be obtained from a system with all components working. We call this the *implicit* reliability specification with respect to O. Now, the system may be — with respect to this specification — as good as "new" also for some states including faulty components. Therefore we define the implicit structure function by the set of up-states relative to $\hat{\delta}$, i.e. $SP_A(\hat{\delta}, \Sigma)$. Hence, we obtain

$$\phi_{\hat{\delta}} = \bigvee_{\alpha\in\mu SP(\hat{\delta},\Sigma)} \alpha, \quad \text{or} \quad \phi_{\hat{\delta}} = \bigwedge_{\beta\in\mu PL(\neg\hat{\delta},\Sigma)} \neg\beta.$$

Accordingly, the implicit reliability of such a system can be obtained as the degree of support $dsp(\hat{\delta}, \Sigma)$. This approach helps to decide whether a system has some implicit redundancy, namely, whether $\phi_{\hat{\delta}}$ represents simply a series system, i.e. $\mu SP(\hat{\delta}, \Sigma)$ has only the set of all assumptions as minimal supporting argument for $\hat{\delta}$.

Lemma 2 *If $\delta \in \mathcal{L}_O$ is a consistent specification with respect to Σ, then $\hat{\delta} \models \delta$.*[1]

This shows that $\hat{\delta}$ is the most stringent, consistent specification over observables O. For all specifications over O the implicit specification has least reliability:

[1] For proofs see [6].

Lemma 3 *If $\delta \in \mathcal{L}_O$ is a consistent specification with respect to Σ, then $SP_A(\hat{\delta}, \Sigma) \subseteq SP_A(\delta, \Sigma)$.*

Corollary 4 *If $\delta \in \mathcal{L}_O$ is a consistent specification with respect to Σ, then $p_{\hat{\delta}} \leq p_\delta$.*

5 Model-Based Diagnostic

5.1 Duality Between Reliability and Diagnostics

A problem of diagnostics arises if an observation indicates that a requirement specification δ is violated. Then the question is: how can the required functionality be recovered? That is, one would like to find out those components whose failure caused the system failure and which have to be fixed or replaced. This analysis will be based on the system description Σ and on the specification δ which is violated.

In fact, we ask, which system states are compatible or consistent with the system description Σ and the violation of the specification δ, expressed by $\neg\delta$. Well, these are of course all states which are consistent with $\Sigma \cup \{\neg\delta\}$, that is the set

$$QS_A{}^c(\bot, \Sigma \cup \{\neg\delta\}) = PL_A(\neg\delta, \Sigma). \quad (11)$$

Remark that this is exactly the set of down states relative to the specification δ (see (10)). Here we have the basic *duality* between *reliability analysis* relative to a requirement specification δ and the *diagnostic problem* relative to the same specification. The conflict set $QS_A(\bot, \Sigma \cup \{\neg\delta\})$ is the computational key to both reliability analysis and diagnostics. It gives the up-states which define reliability and its complement gives the possible states explaining the violation of the reliability specification, i.e. possible diagnostics. It is well known in model-based diagnostics that such conflict sets play a key role [23, 10, 19]. The duality implies that they play an equally important role for model-based reliability.

If the structure function $\phi_{\delta,\Sigma}$ is *monotone*, then to the minimal possible arguments $\beta \in \mu PL(\neg\delta, \Sigma)$ correspond the minimal cuts $\neg\beta$. They represent minimal sets of failed components, which explain the violation of the specification δ, independently on the state of the other components.

Minimal cuts correspond to kernel diagnoses in model-based diagnostics [23]. Usually model-based diagnostics goes not beyond such concepts of diagnostics. It neglects the important role of probability.[2] The observation of the violation of the specification $\neg\delta$ in fact defines the event $QS_A{}^c(\bot, \Sigma \cup \{\neg\delta\})$ in the sample space N_A. That is, the prior probabilities $p(\mathbf{s})$ defined on the states have now to be conditioned on this event. This gives us the *posterior probabilities*

$$p(\mathbf{s}|\neg\delta) = \frac{p(\mathbf{s})}{1 - p(QS_A(\bot, \Sigma \cup \{\neg\delta\}))} = \frac{p(\mathbf{s})}{dpl(\neg\delta, \Sigma)}, \quad (12)$$

for diagnostic states $\mathbf{s} \in QS_A(\bot, \Sigma \cup \{\neg\delta\})$. This underlines once more the key role of the conflict set $QS_A(\bot, \Sigma \cup \{\neg\delta\})$. Its prior probability is sufficient to compute the posterior probabilities of the possible diagnostic states explaining the violation of δ.

[2] See however [19, 3] for a discussion of this subject, and especially [19] for the problems of the approach of De Kleer & Williams [11]. Other approches focus for example on minimal entropy [26] or on restricting the device to have a Bayesian network model [17].

These posterior probabilities represent important additional diagnostic information. For example we may look for diagnostic states with maximal posterior probability. $\tilde{\mathbf{s}}$ is called a *maximal likelihood state*, if

$$p(\tilde{\mathbf{s}}|\neg\delta) = \max_{\mathbf{s}\in {QS_A}^c(\bot,\Sigma\cup\{\neg\delta\})} p(\mathbf{s}|\neg\delta). \quad (13)$$

There may be several such states. They represent most likely states explaining the violation of δ.

Reiter [23] proposed to look especially at possible diagnostic states with a minimal number of faulty components. Intuitively this makes sense: The failure should be explained with a minimal number of down components. If $\mathbf{s}$ is a state, we define s^- to be the set of its negative (down) components. Then we define a partial order between states: $\mathbf{s}' \leq \mathbf{s}$ if $s'^- \subseteq s^-$. *Reiter diagnoses* are now those diagnostic states $\mathbf{s} \in {QS_A}^c(\bot, \Sigma \cup \{\neg\delta\})$, which are minimal with respect to this partial order. We make the reasonable assumption that for every component i we have $p_i > 0.5$ such that $p_i > 1 - p_i$. I.e. it is more probable that a component works than that it is down. Then $\mathbf{s}' < \mathbf{s}$ implies that $p(\mathbf{s}'|\neg\delta) > p(\mathbf{s}|\neg\delta)$. So maximum likelihood states are Reiter diagnoses. The inverse of course does not hold necessarily. Also, if the structure function ϕ_δ is monotone, the s^- of Reiter diagnoses correspond to minimal cuts relative to the specification δ.

The posterior fault probabilities of the components, $p(\neg a_i|\neg\delta)$, are also of interest. The larger this probability, the more critical is component i for the requirement specification δ. So this is a possible *importance measure* for component i relative to the specification (for other importance measures see [4]).

Example 4: (Cont. of Example 1)
Suppose we observe that, although $\neg in$, we have also $\neg out$, i.e. a power system failure is not detected. Note that $\neg in \wedge \neg out \equiv \neg\delta_1$ (cf. Example 3). So we consult the minimal cuts relative to the specification $\neg\delta_1$. There are two minimal cuts: $\{\neg ok_C\}$ and $\{\neg ok_A, \neg ok_B\}$. To any minimal cut corresponds a Reiter diagnosis, namely, $\{ok_A, ok_B, \neg ok_C\}$ to the first cut, and $\{\neg ok_A, \neg ok_B, ok_C\}$ to the second one. One of these two diagnoses must be the maximum likelihood state. The first one has prior probability $0.99 \times 0.99 \times 0.05 = 0.049$, the second one $0.01 \times 0.01 \times 0.95 = 0.000095$. So clearly, the first one is by far the most likely state. The posterior probability is obtained by dividing the prior probability by the unreliability 0.05 relative to δ_1. We obtain for the maximum likelihood state a posterior probability of 0.98. $\ominus$

5.2 Diagnostics Based on Observations of System Behavior

The actual observation is not necessarily the negation of a system requirement, but may be something stronger, which implies the violation of a specification. Indeed, as we saw in Example 4 we observed $\neg in \wedge \neg out \equiv \neg\delta_1$, but $\neg in \wedge \neg out \models \neg\delta_2$. So, we should reconsider the duality between reliability and diagnostics. In fact, assume that we make some observation of the system behavior, expressed in a formula ω over observables. Then we may test whether $\omega \models \neg\hat{\delta}_\Sigma$. If this is the case, then we have a diagnostic problem, in the sense that at least one component must be down.

The solution of this diagnostic problem is found along similar lines as in the previous section. Possible states are those, which are consistent with the system description and the observation. Or, in other words, the states in the conflict set $QS_A(\bot, \Sigma \cup \{\omega\})$ are those which are excluded by the observation. So, the possible diagnostic states are those of the set $PL_A(\omega, \Sigma) = {QS_A}^c(\bot, \Sigma \cup \{\omega\})$. We see that this diagnostic problem is dual to a (fictitious) reliability problem with a "requirement" specification $\neg\omega$. Note that the specification $\neg\omega$ is always consistent with Σ, since $\hat{\delta}_\Sigma$ is consistent and $\omega \models \neg\hat{\delta}_\Sigma$.

Of course, we get a much sharper diagnostic with an observation $\omega \models \neg\hat{\delta}$, than with the information of $\neg\hat{\delta}$ only. This follows, because according to Lemma 3, we have $PL_A(\omega, \Sigma) \subseteq PL_A(\hat{\delta}, \Sigma)$. So, the more precise the observation, the more states are eliminated. A mere statement that a given reliability specification is violated is less informative than a precise observation implying a violation of a requirement specification.

6 Combining Diagnostic and Reliability

We conclude this discussion of duality between reliability and diagnostics by remarking that we may have an observation of the system behavior which does neither entail a specification δ nor its violation $\neg\delta$. But still this observation is information and we can use it to improve reliability analysis and also to perform a preventive diagnostic analysis (see [6]). For reliability as well as for diagnostic, additional measurements — or more generally any additional information — can be taken into account in the framework presented above and helps to focus the reasoning.

7 Conclusions

In this paper we have shown how closely reliability and model-based diagnostic are connected. The framework of probabilistic argumentation system appears to be a framework which covers both approaches. Therefore the generic structure of PAS can be used for solving problems in both domains. The approaches can even be combined and the information specified can be used in the common framework. Further, from the system description of an argumentation system, we can derive the appropriate structure function and — if desirable — take into consideration a reliability requirement. PAS allows to use local computation architectures and approximation techniques [25, 15]. This complements the computational theory of reliability theory.

REFERENCES

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

[2] B. Anrig, 'Probabilistic argumentation systems and model-based diagnostics', in *DX'00, Eleventh Intl. Workshop on Principles of Diagnosis, Morelia, Mexico*, eds., A. Darwiche and G. M. Provan, pp. 1–8, (2000).

[3] B. Anrig, *Probabilistic Model-Based Diagnostics*, Ph.D. dissertation, University of Fribourg, Institute of Informatics, 2000.

[4] B. Anrig, 'Importance measures from reliability theory for probabilistic assumption-based reasoning', in *European Conf. ECSQARU'01, Toulouse*, eds., S. Benferhat and P. Besnard, pp. 692–703. Lecture Notes in Artif. Intell., Springer, (2001).

[5] B. Anrig and F. Beichelt, 'Disjoint sum forms in reliability theory', *ORiON J. OR Society South Africa*, **16**(1), 75–86, (2001).
[6] B. Anrig and J. Kohlas, 'Model-based reliability and diagnostic: A common framework for reliability and diagnostics', Technical Report 02-01, Department of Informatics, University of Fribourg, (2002).
[7] R. E. Barlow and R. Proschan, *Statistical Theory of Reliability and Life Testing*, New York, 1975. IAUTOM 3.9.4-10.
[8] F. Beichelt, *Zuverlässigkeits- und Instandhaltungstheorie*, Teubner, Stuttgart, 1993.
[9] R. Bertschy and P.-A. Monney, 'A generalization of the algorithm of Heidtmann to non-monotone formulas', *J. of Computational and Applied Mathematics*, **76**, 55–76, (1996).
[10] R. Davis, 'Diagnostic reasoning based on structure and behaviour', *Artif. Intell.*, **24**, 347–410, (1984).
[11] J. De Kleer and B. C. Williams, 'Diagnosing multiple faults', *Artif. Intell.*, **32**, 97–130, (1987).
[12] R. Haenni, 'Cost-bounded argumentation', *Int. J. of Approximate Reasoning*, **26**(2), 101–127, (2001).
[13] R. Haenni, 'A query-driven anytime algorithm for assumption-based reasoning', Technical Report 01-26, University of Fribourg, Department of Informatics, (2001).
[14] R. Haenni, B. Anrig, R. Bissig, and N. Lehmann. ABEL homepage. `http://diuf.unifr.ch/tcs/abel`, 2000.
[15] R. Haenni, J. Kohlas, and N. Lehmann, 'Probabilistic argumentation systems', in *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, eds., J. Kohlas and S. Moral, volume 5: Algorithms for Uncertainty and Defeasible Reasoning, Kluwer, Dordrecht, (2000).
[16] K. D. Heidtmann, 'Smaller sums of disjoint products by subproduct inversion', *IEEE Transactions on Reliability*, **38**(3), 305–311, (August 1989).
[17] P. H. Ibargüengoytia, L. E. Sucar, and E. Morales, 'A probabilistic model approach for fault diagnosis', in *DX'00, Eleventh Intl. Workshop on Principles of Diagnosis, Morelia, Mexico*, eds., A. Darwiche and G. M. Provan, pp. 79–86, (2000).
[18] J. Kohlas. Valuation algebras: Generic architecture for reasoning. draft, 2002.
[19] J. Kohlas, B. Anrig, R. Haenni, and P.-A. Monney, 'Model-based diagnostics and probabilistic assumption-based reasoning', *Artif. Intell.*, **104**, 71–106, (1998).
[20] J. Kohlas and P.-A. Monney, *A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence*, volume 425 of *Lecture Notes in Economics and Mathematical Systems*, Springer, 1995.
[21] J. Kohlas and R. F. Stärk, 'Information algebras and information systems', Technical Report 96–14, University of Fribourg, Institute of Informatics, (1996).
[22] G. M. Provan, 'An integration of model-based diagnosis and reliability theory', in *DX'00, Eleventh Intl. Workshop on Principles of Diagnosis, Morelia, Mexico*, eds., A. Darwiche and G. M. Provan, pp. 193–200, (2000).
[23] R. Reiter, 'A theory of diagnosis from first principles', *Artif. Intell.*, **32**, 57–95, (1987).
[24] G. Shafer, *The Mathematical Theory of Evidence*, Princeton University Press, 1976.
[25] P. P. Shenoy and G. Shafer, 'Axioms for probability and belief functions propagation', in *Uncertainty in Artif. Intell. 4*, eds., R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer. North Holland, (1990).
[26] P. Struss, 'Testing for discrimination of diagnoses', in *DX'94, Fifth Intl. Workshop on Principles of Diagnosis, New Paltz, USA*, (1994).