ORiON, Vol. 16, No. 1, pp. 75-86 *ISSN 0259-191-X*

DISJOINT SUM FORMS IN RELIABILITY THEORY

B. ANRIG[1], F. BEICHELT[2]
[1] **University of Fribourg, Department of Informatics, Rue de Faucigny 2, 1700 Fribourg, Switzerland**
[2] **University of the Witwatersrand, School of Statistics and Actuarial Science Private Bag 3, WITS 2050, Johannesburg, South Africa**

ABSTRACT
The structure function φ of a binary monotone system is assumed to be known and given in a disjunctive normal form, i.e. as the logical union of products of the indicator variables of the states of its subsystems. Based on this representation of φ, an improved Abraham algorithm is proposed for generating the disjoint sum form of φ. This form is the base for subsequent numerical reliability calculations. The approach is generalized to multivalued systems. Examples are discussed.

1. INTRODUCTION

We deal with a basic problem of reliability theory, namely with finding and analyzing algorithms for calculating reliability criteria of systems based on reliability criteria of its elements (subsystems). Even in the simple case of binary monotone systems with independent elements, these algorithms generally have running times which increase exponentially fast with the complexity of the system [5]. Within that limitation it is, however, imperative to develop algorithms, which are relatively fast and applicable to a broad class of problems. The first step is to determine the structure function of the system. This paper requires the structure function to be known and given in a disjunctive normal form.

2. BINARY SYSTEMS

Let S be the system under consideration and $e_1, e_2, ..., e_n$ its n elements. Let furthermore z_s be indicator variable of the system state and z_i the indicator variable of element e_i. Then, in a binary system,

1) Research supported by grant No 2000-061145400 of the Swiss National Foundation for Research
2) Research supported by grant No. 2046914 of the SA National Research Foundation

$$z_S = \begin{cases} 1 & \text{if } S \text{ is operating} \\ 0 & \text{otherwise} \end{cases}, \qquad z_i = \begin{cases} 1 & \text{if } e_i \text{ is operating} \\ 0 & \text{otherwise} \end{cases}.$$

The *structure function* φ of S is a Boolean function, which describes the mutual relationship between the states of the system and its elements:

$$z_S = \varphi(z_1, z_2, \dots, z_n).$$

z_S and the z_i are *Boolean*, *binary* or, more specific, (0-1)-*variable*. For any two (0-1)-variable x and y, *disjunction*, *conjunction* and *complement* are defined as follows:

Disjunction $\quad x \vee y = x + y - xy = \max(x, y)$

Conjunction $\quad x \wedge y = x\, y = \min(x, y)$

Negation $\quad \bar{x} = 1 - x$

Furthermore, x and y are called *disjoint* if $x\, y = 0$. Hence,

$$x \vee y = x + y \quad \text{if } x \text{ and } y \text{ are disjoint}. \tag{1}$$

Let $\mathbf{Z}$ be the set of all 2^n state vectors of the system. The system is *monotone* if φ is nondecreasing in each variable z_i and each z_i is relevant with respect to φ, i.e. z_i influences the value of φ. A vector $\vec{z} \in \mathbf{Z}$ is called a *path vector* of φ if $\varphi(\vec{z}) = 1$. A path vector $\vec{z}$ is called *minimal* if $\varphi(\vec{y}) = 0$ for all $\vec{y} \in \mathbf{Z}$ with $\vec{y} < \vec{z}$. Note that $\vec{y} = (y_1, y_2, \dots, y_n) < \vec{z} = (z_1, z_2, \dots, z_n)$ iff $y_i < z_i$ for all $i = 1, 2, \dots, n$ and $y_j < z_j$ for at least one j. Let $\vec{z}_1$, $\vec{z}_2, \dots,$ $\vec{z}_w$ be the set of all minimal path vectors. The *minimal path set* belonging to $\vec{z} = (z_1, z_2, \dots, z_n)$ is defined as

$$W_i = \{ j,\ z_j = 1 \}.$$

Obviously, the concepts of (minimal) path vectors and (minimal) path sets are equivalent. For a thorough discussion of these concepts see, for instance, [7] and [15]. It is easy to see that the structure function of any binary monotone system with minimal path sets W_1, $W_2, \dots,$ W_w has structure

$$\varphi(\vec{z}) = A_1 \vee A_2 \vee \cdots \vee A_w \tag{2}$$

with

$$A_j = \prod_{i \in W_j} z_i; \quad j = 1, 2, \dots, w.$$

Since there are computer-aided methods for determining the minimal path sets of any binary monotone system, the disjunctive normal form of φ given by (2) is most widely used for generating *disjoint sum forms* of structure functions. A disjoint sum form of φ has structure

$$\varphi(\vec{z}) = D_1 + D_2 + \cdots + D_d; \quad D_i D_j = 0, \; i \neq j \,, \tag{3}$$

where the D_k; $k = 1,2,...,d$; are products of some z_i and $\bar{z}_j$; $i \neq j$. Hence, φ given by (3) is a sum of mutually disjoint terms D_i .

Let us now take into account that the z_i are random variables with

$$z_i = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } \bar{p}_i = 1 - p_i \end{cases}; \quad i = 1,2,...,n\,;$$

and let $p_s = P(z_s = 1) = P(\varphi(\vec{z}) = 1)$ be the probability that the system is operating. p_s is called the *availability of the system* and p_i is the *availability of element* e_i. Since $z_s = \varphi(\vec{z})$ is a binary random variable,

$$p_s = E(\varphi(\vec{z}))\,.$$

This relationship shows the practical importance of a disjoint sum form (3): The system availability is simply given by

$$p_s = E(D_1) + E(D_2) + \cdots + E(D_d)\,.$$

If the z_i are independent random variables, i.e. the elements operate independently from each other, then $E(D_i)$ is obtained from D_i simply by replacing there the z_i and $\bar{z}_j$ with the corresponding p_i and $\bar{p}_j$, respectively.

Structure functions, in particular disjoint sum forms, are not unique. For the sake of computational efficiency, a disjoint sum form should be of low complexity, i.e. both d and the total number of factors in the D_i should be small. Numerous algorithms transforming the disjunctive normal form (2) into a disjoint sum form of low complexity have been developed. Most popular is the algorithm of Abraham [2]. It has formed the basis for substantially improved versions yielding disjoint sum forms of lower complexity than the original version of Abraham [7], [8], [9], [14]. The probably most efficient algorithm not based on Abraham's one, is due to *Torrey* [22]. For surveys, see [21], [23].

The following version of Abraham's algorithm is based on an algorithm given in [7]. To describe the algorithm, some further notation is needed. Let M be the set of all possible products of some z_i and $\bar{z}_j$. The principle of the algorithm consists in replacing each A_k in (2) with a sum of disjoint products

$$L_k = \sum_{D \in M_k} D, \qquad M_k \subseteq M$$

such that

$$A_1 \vee A_2 \vee \cdots \vee A_w = \sum_{k=1}^{w} L_k = \sum_{D \in M_\varphi} D$$

with $L_1 = A_1$ and the set $M_\varphi = M_1 \cup M_2 \cup \cdots M_w$ consists of mutual disjoint products from M. Thus, the set M_φ can be identified with the structure function φ. The sums L_k are successively generated from sums $L_{1,k}, L_{2,k}, \cdots, L_{k-1,k}$ with property

$$A_1 \vee A_2 \vee \cdots \vee A_j \vee A_k = A_1 \vee A_2 \vee \cdots \vee A_j + L_{j,k},$$

where

$$L_{j,k} = \sum_{D \in M_{j,k}} D, \qquad M_{j,k} \subseteq M .$$

The process starts for each k = 2, 3, …, w at j = 1 and stops at j = k-1, $L_{k-1,k} = L_k$. The transition from $L_{j-1,k}$ to $L_{j,k}$ or, equivalently, from $M_{j-1,k}$ to $M_{j,k}$, depends on which of the following three cases occurs. To characterize these cases, let A be the product of some z_i and $C(A,B) = \{C_1, C_2, ..., C_c\}$ $B \in M$, the set of all those z_i, which are factors in A, but not in B.

a) $A \cdot B = \varnothing$ (A and B are disjoint) if z_i is a factor in A and $\bar{z}_i$ is a factor in B.

b) $A \vee B = A$ if A and B are not disjoint and $C(A,B) = \varnothing$.

c) $A \vee B = A + \bar{C}_1 B + C_1 \bar{C}_2 B + \cdots + C_1 C_2 \cdots C_{c-1} \bar{C}_c B$ if A and B are not disjoint and $C(A,B) \neq \varnothing$.

To construct $M_{j,k}$ from $M_{j-1,k}$, $C(A_j, B)$ is determined for $B \in M_{j-1,k}$. If case a) applies, then B also is an element of $M_{j,k}$. If b) is true, then B is eliminated, since it does not contribute to the construction of $M_{k-1,k} = M_k$. In case c), $M_{j,k}$ contains the products $\bar{C}_1 B, C_1 \bar{C}_2 B, \cdots, C_1 C_2 \cdots C_{c-1} \bar{C}_c B$. The complete set $M_{j,k}$ is obtained if this procedure is repeated for all $B \in M_{j-1,k}$ and starts with $M_{0,k} = \{A_k\}$. Note that the sets M_k are generated independently from each other. Hence, they can be determined in an arbitrary order.

Algorithm 1

1 Order the A_j according to the number of their factors.

2 Initialize $M_\varphi = \{A_1\}$, $k = 2$

3 Initialize $M_{0,k} = \{A_k\}$, $k = 2$

4 Initialize $M_{j,k} = \varnothing$

5 For all $B \in M_{j-1,k}$:

5.1 If A_j and B are disjoint, B becomes element of $M_{j,k}$. Select another B.

5.2 Determine $C(A_j, B)$.

5.3 If $C(A_j, B) = \varnothing$, delete B and select another B.

5.4 If $C(A_j, B) = \{C_1, C_2, ..., C_c\}$, $c \geq 1$, then $\overline{C}_1 B,\ C_1\overline{C}_2 B,\ \cdots,\ C_1 C_2 \cdots C_{c-1}\overline{C}_c B$ become elements of $M_{j,k}$.

5.5 Select another B.

6 If $j < k-1$, then $j \leftarrow j+1$ and go to 4.

7 Expand M_φ by adding $M_{k-1,k}$.

8 If $k < w$, then $k \leftarrow k+1$ and go to 3. If $k = w$, **STOP**.

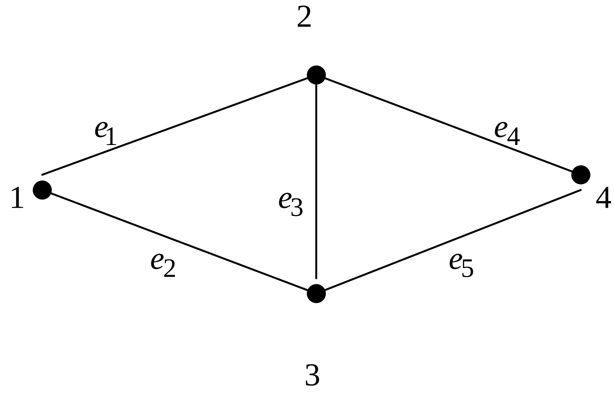

Figure 1 Bridge structure

Example 1 Let us consider a system the reliability block diagram of which is given by the “bridge structure” (Figure 1), i.e. it has the four nodes 1, 2, 3 and 4 with 1 and 4 being entrance and exit nodes, respectively, and edges $e_1 = (1,2)$, $e_2 = (1, 3)$, $e_3 = (2,3)$, $e_4 = (2,4)$, and $e_5 = (3,4)$.

From Figure 1, $W_1 = \{1,4\}$, $W_2 = \{2,5\}$, $W_3 = \{1,3,5\}$, $W_4 = \{2,3,4\}$. Hence,

$$A_1 = z_1 z_4,\; A_2 = z_2 z_5,\; A_3 = z_1 z_3 z_6,\; A_4 = z_2 z_3 z_4,$$

so that the disjunctive normal form (2) is given by

$$\varphi(\vec{z}) = z_1 z_4 \vee z_2 z_5 \vee z_1 z_3 z_5 \vee z_2 z_3 z_4 .$$

To apply algorithm 1, firstly, initialize $M_{0,2} = \{A_2\}$.Then $B = A_2 \in M_{0,2}$ is selected. Since $C(A_1, B) = \{z_1, z_4\}$, $M_2 = M_{1,2} = \{\bar{z}_1 z_2 z_5, z_1 z_2 \bar{z}_4 z_5\}$. Secondly, initialize set $M_{0,3} = \{A_3\}$ and select $B = A_3$. Then $C(A_1, B) = \{z_4\}$. Therefore, $M_{1,3} = \{z_1 z_3 \bar{z}_4 z_5\}$. With $B = z_1 z_3 \bar{z}_4 z_5$, $C(A_2, B) = \{z_2\}$. Thus, $M_3 = M_{2,3} = \{z_1 \bar{z}_2 z_3 \bar{z}_4 z_5\}$. Thirdly, initialize $M_{0,4} = \{A_4\}$. Then, with $B = A_4$, $C(A_1, B) = \{z_1\}$. This gives $M_{1,4} = \{\bar{z}_1 z_2 z_3 z_4\}$. With $B = \bar{z}_1 z_2 z_3 z_4$, the corresponding set $C(A_2, B)$ becomes $C(A_2, B) = \{z_5\}$. Therefore, $M_{2,4} = \{\bar{z}_1 z_2 z_3 z_4 \bar{z}_5\}$. Lastly, with $B = \bar{z}_1 z_2 z_3 z_4 \bar{z}_5$, A_3 and B are disjoint (case a). Hence, B is also element of $M_{3,4}$. This implies $M_4 = M_{3,4} = M_{2,4}$. In view of $M_\varphi = M_1 \cup M_2 \cup M_3 \cup M_4$ with $M_1 = A_1$, the disjoint sum form is

$$\varphi(\vec{z}) = z_1 z_4 + \bar{z}_1 z_2 z_5 + z_1 z_2 \bar{z}_4 z_5 + z_1 \bar{z}_2 z_3 \bar{z}_4 z_5 + \bar{z}_1 z_2 z_3 z_4 \bar{z}_5 .$$

Algorithm 1 is not applicable to generating disjoint sum forms of non-monotone structure functions. However, there are technical systems, whose reliability behaviour can only be described by non-monotone structure functions. Examples are given in [7]. Moreover, the problem of generating disjoint sum forms from disjunctive normal forms also arises in probabilistic model-based reasoning and in the Dempster-Shafer theory of evidence. Here the Boolean functions of interest ("structure functions") are usually non-monotone. Abraham's approach to generating disjoint sum forms from disjunctive normal forms of Boolean functions has been generalized to non-monotone Boolean and even multi-valued functions in [3, 4], 10, 12, 16, 17, 18, 20].

3. MULTIVALUED SYSTEMS

To assume that the system S and its elements $e_1, e_2, ..., e_n$ can only be in either state "available" or "not available" is frequently an inadmissible oversimplification of the real situation. Consider, for example, systems (elements) with different operating and/or failure modes. Hence, it makes sense to assume that the indicator variables z_s and z_i of the states of S and e_i can assume values from sets

$$\mathbf{Z}_s = \{z_{s,1}, z_{s,2}, \ldots, z_{s,r_s}\} \text{ and } \mathbf{Z}_i = \{z_{i,1}, z_{i,2}, \ldots, z_{i,r_i}\}; \quad i = 1,2,\ldots,n,$$

respectively. Then a state vector $\vec{z} = (z_1, z_2, \ldots, z_n)$ of the system is element of

$$\mathbf{Z} = \mathbf{Z}_1 \times \mathbf{Z}_2 \times \cdots \times \mathbf{Z}_n \tag{4}$$

and the state space $\mathbf{Z}$ has $r_1 \cdot r_2 \cdots r_n$ elements. $\mathbf{Z}_s, \mathbf{Z}_i$ is called the *frame* of the indicator variables z_s, z_i; $i = 1,2,\ldots,n$; respectively. The structure function $z_s = \varphi(\vec{z})$ maps $\mathbf{Z}$ onto $\mathbf{Z}_s$. A *multivalued coherent system* with finite state sets $\mathbf{Z}_s$, $\mathbf{Z}_i$; $i = 1,2,\ldots,n$; and structure function $z_s = \varphi(\vec{z})$, $\vec{z} = (z_1, z_2, \ldots, z_n) \in \mathbf{Z}$, is commonly defined as follows:

1) φ is *nondecreasing* in each argument and

2) $$\min_{i=1,2,\ldots,n} z_i \le \varphi(\vec{z}) \le \max_{i=1,2,\ldots,n} z_i .$$

Usually, the elements of the $\mathbf{Z}_i$ are real numbers. Otherwise, a total order in the set $\mathbf{Z}_1 \cup \mathbf{Z}_2 \cup \cdots \cup \mathbf{Z}_n$ must be given (see, for instance, [1, 11, 19]). In these papers it is generally assumed that the state spaces $\mathbf{Z}_s$ and $\mathbf{Z}_i$ are identical or $r_s = 2$ and $r_i > 2$. Multi-valued systems with nondenumerable state spaces are, for instance, considered in [6]. Here a partial generalization of these models is dealt with: The state spaces $\mathbf{Z}_s, \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n$ need not be identical (although this can be assumed without loss of generality) and the structure function φ need not be nondecreasing. However, as in the previous section, the state z_s of the system can only assume values 0 or 1, i.e. $\mathbf{Z}_s = \{0, 1\}$ (system is not available, is available). To be able to present an algorithm for constructing a disjoint sum form of the structure function, concepts introduced in section 2 have to be generalized. Let $Z \subseteq \mathbf{Z}_i$. A *set constraint* (SC) over z_i with respect to Z is denoted as $\langle z_i \in Z \rangle$ and defined by

$$\langle z_i \in Z \rangle = \begin{cases} 1 & \text{if } z_i \text{ assumes a value from Z} \\ 0 & \text{if } z_i \text{ assumes a value from } Z_i \setminus Z \end{cases}$$

Thus, Z is that subset of states, in which element e_i operates satisfactorily. From the point of view of logic, $\langle z_i \in Z \rangle$ is a *predicate*, which is true iff z_i assumes a value from Z. Obviously, if $\mathbf{Z}_i = \{0, 1\}$, then the Boolean variable z_i is equivalent to $\langle z_i \in \mathbf{Z}_i \rangle$. Hence, SC's are generalizations of Boolean variables with

$$0 = \langle z_i \in \varnothing \rangle < \langle z_i \in \mathbf{Z}_i \rangle = 1$$

Let $Z, Z' \subseteq \mathbf{Z}_i$, $Z'' \subseteq \mathbf{Z}_j$. *Disjunction, conjunction* and *negation* of SC's are defined as

Disjunction: $\langle z_i \in Z' \rangle \vee \langle z_j \in Z'' \rangle = \max\{\langle z_i \in Z' \rangle, \langle z_j \in Z'' \rangle\}$

Conjunction: $\langle z_i \in Z' \rangle \wedge \langle z_j \in Z'' \rangle = \min\{\langle z_i \in Z' \rangle, \langle z_j \in Z'' \rangle\}$

Negation: $\overline{\langle z_i \in Z \rangle} = \langle z_i \in \mathbf{Z}_i \setminus Z \rangle$

In particular, for SC's over the same variable,

Disjunction $\langle z_i \in Z' \rangle \vee \langle z_i \in Z'' \rangle = \langle z_i \in Z' \cup Z'' \rangle$

Conjunction $\langle z_i \in Z' \rangle \wedge \langle z_i \in Z'' \rangle = \langle z_i \in Z' \cap Z'' \rangle$

Two SC's $\langle z_i \in Z' \rangle$ and $\langle z_j \in Z'' \rangle$ are said to be *disjoint* if $\langle z_i \in Z' \rangle \wedge \langle z_j \in Z'' \rangle = 0$. If $i = j$, then $\langle z_i \in Z' \rangle$ and $\langle z_i \in Z'' \rangle$ being disjoint is equivalent to $Z' \cap Z'' = \varnothing$.

An SC $\langle z_i \in Z \rangle$ is called *proper* if $Z \neq \varnothing$ and $Z \neq \mathbf{Z}_i$.

An *SC-clause* (*SC-term*) is a disjunction (conjunction) of proper SC's with every variable z_i occurring at most once.

For any two (0,1)-functions f and g defined on $\mathbf{Z}$,

$$f \wedge g = \min(f, g) \text{ and } f \vee g = \max(f, g)$$

and $f = f(\vec{y})$ and $g = g(\vec{z})$ are called *disjoint* if $f(\vec{y}) \wedge g(\vec{z}) = 0$ for all $\vec{y}, \vec{z} \in \mathbf{Z}$. ($f$ and g may actually only depend on $k, k < n,$ of the variables $z_1, z_2, ..., z_n$. In this case, the residual $n - k$ variables are *irrelevant* to f and g and can be deleted.) If f and g are disjoint, then

$$f \vee g = f + g$$

Let $Z_{i_k} \subset \mathbf{Z}_{i_k}$; $k = 1,2,...,r;\ 0 < r < n,$ and f be the corresponding SC-term, i.e.

$$f = \langle z_{i_1} \in Z_{i_1} \rangle \wedge \langle z_{i_2} \in Z_{i_2} \rangle \wedge \cdots \wedge \langle z_{i_r} \in Z_{i_r} \rangle$$

As in chapter 2, it will be assumed that the $z_1, z_2, ..., z_n$ are independent. Then, since SC's are random (0-1)-variables,

$$P(f = 1) = E(f) = P\{\langle z_{i_1} \in Z_{i_1} \rangle = 1\} P\{\langle z_{i_2} \in Z_{i_2} \rangle = 1\} \cdots P\{\langle z_{i_r} \in Z_{i_r} \rangle = 1\}$$

with

$$P\left(\langle z_k \in Z_k\rangle = 1\right) = \sum_{z_{k,j}\in Z_k} P(z_k = z_{k,j})$$

Let the structure function of a possibly noncoherent multivalued system be given in the form

$$\varphi(\vec{z}) = f_1 \vee f_2 \vee \cdots \vee f_m, \tag{5}$$

where the f_k are proper terms with SC's over all or some of the $z_1, z_2, \ldots, z_n$. The usefulness of transforming structure function of type (5) into disjoint sum forms is motivated as in chapter 2. The following algorithm is an adaptation of algorithm 1 to noncoherent system functions with multi-valued arguments [3, 20]. It is based on a version firstly presented in [18]. An alternative approach using the information that every element is in exactly one mode is presented in [4].

Algorithm 2

; input: $\varphi = f_1 \vee f_2 \vee \cdots \vee f_m$ (order: f_j contains not more SC's than f_{j+1})

; output: sets $M_{i,j}$; $j = 1, \ldots, r$; $i = 1, \ldots, j-1$

for $j = 1$ to m

  $M_{0,j} := \{f_j\}$

  for $i = 1$ to $j-1$

  $M_{i,j} := \varnothing$

  for all D in $M_{i-1,j}$

    if D and f_i are disjoint, then add D to $M_{i,j}$

    else define $I := (I_i - I_D) \cup \{k \in I_i \cap I_D : Y_k \not\subseteq Z_k\}$

        with $f_i = \bigwedge_{k\in I_i} \langle z_k \in Z_k\rangle$ and $D = \bigwedge_{k\in I_D} \langle z_k \in Y_k\rangle$

        if $I = \{i_1, i_2, \ldots, i_t\} \neq \varnothing$, then add the following formulas to $M_{i,j}$:

          $D \wedge \langle z_{i_1} \in \mathbf{Z}_{i_1} \setminus Z_{i_1}\rangle$

          $D \wedge \langle z_{i_1} \in \mathbf{Z}_{i_1}\rangle \wedge \langle z_{i_2} \in \mathbf{Z}_{i_2} \setminus Z_{i_2}\rangle$

          $\vdots$

          $D \wedge \langle z_{i_1} \in Z_{i_1}\rangle \wedge \cdots \wedge \langle z_{i_{t-1}} \in Z_{i_{t-1}}\rangle \wedge \langle z_{i_t} \in \mathbf{Z}_{i_t} \setminus Z_{i_t}\rangle$

Note that one has to make sure that the algorithm generates only proper SC-terms. By using an appropriate data structure for representing SC-terms, this can be done efficiently. The sets $M_{i,j}$ only contain SC-terms and the corresponding disjoint sum form is

$$\varphi = \sum_{j=1}^{m} \sum_{D \in M_{j-1,j}} D$$

Example 2 Consider variables z_1, z_2, z_3 with identical frame $\mathbf{Z} = \{1,2,3,4\}$. Let us assume a non-monotone system function of type (5) given by $\varphi = f_1 \vee f_2$ with

$$f_1 = \langle z_1 \in \{3,4\} \rangle \wedge \langle z_2 \in \{1\} \rangle, \;\; f_2 = \langle z_1 \in \{1,3,4\} \rangle \wedge \langle z_3 \in \{1,3\} \rangle.$$

Applying algorithm 2 yields:

$j = 1$: Initiate $M_{0,1} := \{f_1\}$

$j = 2$: Initiate $M_{0,2} := \{f_2\}$

$i = 1$: The only element of $M_{0,2}$ is not disjoint with f_1. Hence, determine $I = \{1,2\}$ and construct formulas

$$\langle z_1 \in \{1,3,4\} \rangle \wedge \langle z_3 \in \{1,3\} \rangle \wedge \overline{\langle z_1 \in \{3,4\} \rangle}$$

$$\langle z_1 \in \{1,3,4\} \rangle \wedge \langle z_3 \in \{1,3\} \rangle \wedge \langle z_1 \in \{3,4\} \rangle \wedge \langle z_3 \in \{1,3\} \rangle \wedge \overline{\langle z_2 \in \{1\} \rangle}$$

Simplifying these formulas to obtain SC-terms yields

$$\left\{ \langle z_1 \in \{1,3,4\} \rangle \wedge \langle z_3 \in \{1,3\} \rangle \wedge \langle z_1 \in \{3,4\} \rangle \wedge \overline{\langle z_2 \in \{1\} \rangle} \right\}$$

Hence, the system availability becomes

$$P(\varphi = 1) = \sum_{D \in M_{0,1}} P(D=1) + \sum_{D \in M_{1,2}} P(D=1)$$

$$= P(\langle z_1 \in \{3,4\} \rangle) \cdot P(\langle z_2 \in \{1\} \rangle) + P(\langle z_1 \in \{1\} \rangle) \cdot P(\langle z_3 \in \{1,3\} \rangle)$$

$$+ P(\langle z_1 \in \{3,4\} \rangle) \cdot P(\langle z_2 \in \{2,3,4\} \rangle) \cdot P(\langle z_3 \in \{1,3\} \rangle).$$

REFERENCES

[1] A.M. ABOUAMMOH,. and M.A. Al-KADI, Multistate coherent systems of order *k*, *Microelectronics and Reliability*, **35**, 1415-1421 (1995).

[2] J.A. ABRAHAM, An improved algorithm for network reliability, *IEEE Trans. on Reliab.*, **28**, 58-61 (1979).

[3] B.A. ANRIG, A generalization of the algorithm of Abraham, In: M. Nikulin, M. and Limios, N. (eds.), *Proc. of the 2nd Int. Conf. on Mathematical Methods in Reliability*, **II**, Bordeaux, July 3-4, 2000, 95-98 (2000).

[4] B. ANRIG, and P.A. MONNEY, Using propositional logic to compute the probability of diagnoses in multistate systems, *Internaional. Journal of Approximate Reasoning* **20**, 113-143 (1999).

[5] M.O. BALL, and J.S. PROVAN, Disjoint products and efficient computation of reliability, *Operations Research*, **36**, 703-715 (1988).

[6] L.A. BAXTER, Continuum structures I, *Journal of Applied Probabillity*, **21**, 802-815 (1984).

[7] F. BEICHELT, *Reliability and Maintenance Theory* (in German), Teubner, Stuttgart (1993).

[8] F. BEICHELT and L. SPROß, An improved Abraham-method for generating disjoint sums, *IEEE Transactions on Reliability*, **36**, 70-74 (1987).

[9] F. BEICHELT and L. SPROß, Comment on "An improved Abraham-method for generating disjoint sums.", *IEEE Transactions on Reliability*, **38**, 422-424 (1989).

[10] R. BERTSCHY and P.A. MONNEY, A generalization of the algorithm of Heidtmann to non-monotone formulas. *Journal of Computational and Applied Mathematics*, **76**, 55-76 (1996).

[11] W.S. GRIFFITH, Multistate reliability models, *Journal of Applied Probability*, **17**, 735-744 (1980).

[12] R. HAENNI, R. and LEHMANN, Building argumentation systems on set constraint logic, In: Bouchon-Meunier, B. et al (eds.), *Information, Uncertainty and Fusion*, Kluwer Academic Publishers, 393-406 (2000).

[13] R. HÄHNLE,, Advanced many-valued logics, In: Gabbey, D. (ed.), *Handbook of Philosoph. Logic, Vol. 6*: *Alternatives to Classical Logic I (*2. ed.), Kluwer, 2000.

[14] K.D. HEIDTMANN, Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability*, **38**, 305-311 (1989).

[15] J. KOHLAS, *Reliability and Availability* (in German). Teubner, Stuttgart (1987).

[16] J. KOHLAS, B. ANRIG, R. HAENNI,. and P.A. MONNEY, Model-based diagnostics and probabilistic assumption-based reasoning, *Artificial Intelligence*, **104**, 71-106, (1998).

[17] J. KOHLAS, R. HAENNI. and N. LEHMANN, Probabilistic argumentation systems, In: Kohlas, J. and Moral, S. (eds.), *Handbook of Defeasible Reasoning and*

Uncertainty Management Systems, vol. 5: Algorithms for Uncertainty andDefeasible Reasoning, Kluwer (2000).

[18] J. KOHLAS,. and P.A. MONNEY, *A Mathematical Theory of Hints, An Approach to the Dempster-Shafer Theory of Evidence,* Vol. 25 of Lecture Notes in Economics and Mathematical Systems, Springer (1995).

[19] F.C. MENG, Characterizing the Barlow-Wu structure functions, *Naval Research Logistics,* **41**, 661-668 (1994).

[20] P.A. MONNEY,. and B. ANRIG, Computing the probability of formulas representing events in product spaces, In: Bouchon-Meunier, B. et al (eds.), *Information, Uncertainty and Fusion,* Kluwer Academic Publishers, 197-208 (2000).

[21] S. RAI, M. VEERARAGHAVAN, M. and S. TRIVEDI, A survey of efficient reliability computation using disjoint products approach, *Networks,* **25**, 147-163 (1995).

[22] J. TORREY, A pruned tree approach to reliability computation, *IEEE Transactions on Reliability*, **32**, 170-174 (1983).

[23] M. VEERARAGHAVAN,. and S. TRIVEDI, Multiple variable inversion techniques, Chapter 2 in: Misra, K. (ed.), *New trends in system reliability evaluation,* Elsevier, 39-74, (1994).