

---

# Binary Decision Diagrams in Reliability Analysis of Standard System Structures

Miroslav Kvassay, Elena Zaitseva, Vitaly Levashenko, Jozef Kostolny

Faculty of Management Science and Informatics

University of Zilina

Zilina, Slovakia

{miroslav.kvassay, elena.zaitseva, vitaly.levashenko, jozef.kostolny}@fri.uniza.sk

**Abstract**—One of the current issues of reliability analysis is investigation of complex systems consisting of many components. Investigation of such systems requires developing efficient methods for their representation. One of the possible ways is use of Binary Decision Diagrams (BDDs) that allow storing information about system topology in an efficient way. However, the main problem behind BDDs is how to create a good BDD. In this paper, we focus on construction of BDDs for common system structures such as series, parallel and  $k$ -out-of- $n$ . Using these results, we present a method for creation of a good BDD for  $k$ -to- $l$ -out-of- $n$  systems, which are a typical instance of noncoherent systems. In the next phase, we use BDDs and logical differential calculus to develop an efficient method for importance analysis of  $k$ -to- $l$ -out-of- $n$  systems.

**Keywords**—reliability, importance measures, binary decision diagram, direct partial Boolean derivative

## I. INTRODUCTION

Every system consists of one or more components. One of the principal tasks of reliability analysis is identification of components with the greatest influence on system activity. This task is addressed in a special part of reliability engineering that is known as importance analysis [1, 2].

Importance analysis can be qualitative or quantitative. The qualitative analysis deals with identification of circumstances under which system fails or is functioning. These circumstances can be defined in several ways. One possibility is to identify situations in which a failure of a given component causes system failure or in which a repair of the component results in system functioning. These situations are described by so-called critical path vectors and critical cut vectors respectively [1, 2]. Knowledge of these circumstances allows us to identify scenarios whose occurrence result in system failure or to propose maintenance procedures that lead into system functioning. However, it can be quite difficult to investigate influence of individual system components directly from the knowledge of critical path/cut vectors. Such investigation is addressed to the quantitative analysis.

The quantitative analysis deals with quantification of consequences of a failure or repair of a given component on system activity. For this task, a lot of indices known as Importance Measures (IMs) have been developed [2]. Some of the most commonly known are Structural Importance (SI), Birnbaum's Importance (BI), and Criticality Importance (CI). These and other IMs have been developed originally for the

analysis of coherent systems, which are characterized by the fact that there exists no circumstance under which a failure of a system component can result in system repair. Typical examples of such systems are series, parallel, and  $k$ -out-of- $n$  systems [1, 2]. However, there also exist real examples of systems that do not meet the assumption of the coherency. Typical instances of such systems are  $k$ -to- $l$ -out-of- $n$  systems [3], which are functioning if at least  $k$  but not more than  $l$  system components are working. Investigation of such systems has to be done very carefully since most of the methods of reliability analysis are based on the coherency assumption.

A lot of papers have dealt with importance analysis of coherent systems. A comprehensive study about the existing IMs and relations between them has been done in [2]. However, only several works focuses on importance analysis of noncoherent systems. The bases of the analysis of such systems have been given in [4, 5], where several IMs for noncoherent systems described by fault trees have been generalized. Based on the ideas presented in those works, another approach for calculation of SI, BI, and CI measures for noncoherent systems defined using structure function has been developed in [6, 7]. This approach is based on Direct Partial Boolean Derivatives (DPBDs).

DPBDs are a part of logical differential calculus. They have been developed for analysis of dynamic properties of Boolean functions [8, 9]. In case of reliability analysis, they can be applied to find circumstances under which a failure/repair of a given component results in system failure/repair [6, 7, 10]. Identification of such situations plays a key role in computing the aforementioned IMs. However, the primary problem behind them is their efficient calculation for systems consisting of a huge amount of components. One of the possible solutions to this problem is to express system structure in the form of a Binary Decision Diagram (BDD).

A BDD represents a graph structure developed for efficient manipulation with Boolean functions of high dimensions [9, 11]. However, creation of a good BDD, i.e. a BDD of a minimal size, is a difficult problem [9]. In this paper, we focus on several common structures used in reliability engineering. Firstly, we will show how to construct good BDDs for series and parallel systems. Based on this, a way of how to construct a good BDD for a  $k$ -out-of- $n$  system will be shown. These results will be then used to find good BDDs for noncoherent  $k$ -to- $l$ -out-of- $n$  systems. Finally, based on all these results, efficient algorithms for calculation of SI, BI, and CI measures

---

This work was supported by grants VEGA 1/0498/14 and VEGA 1/0038/16.

for series, parallel,  $k$ -out-of- $n$ , and  $k$ -to- $l$ -out-of- $n$  systems will be proposed. The background for these algorithms will be an efficient calculation of DPBDs.

## II. MATHEMATICAL BACKGROUND

### A. Structure Function and System Availability

One of the typical assumptions of reliability engineering is that the system and all its components can be in one of two possible states – functioning and failed. The former state is usually denoted by number 1, while the latter is expressed as number 0. A map that defines the dependency of system states on states of its components is known as structure function. For a system consisting of  $n$  components, this function agrees with the following mapping [1, 2, 10]:

$$\phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_n): \{0,1\}^n \rightarrow \{0,1\}, \quad (1)$$

where  $x_i$ , for  $i = 1, 2, \dots, n$ , is a variable defining state of the  $i$ -th system component (state variable), and  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is a vector defining states of the system components (state vector).

Based on the properties of the structure function, two basic types of systems can be defined – coherent and noncoherent. A coherent system is a system, whose structure function is non-decreasing in all its arguments, i.e. [12]:

$$\phi(1_i, \mathbf{x}) \geq \phi(0_i, \mathbf{x}) \text{ for any } i \in \{1, 2, \dots, n\}, \quad (2)$$

where  $(a_i, \mathbf{x}) = (x_1, x_2, \dots, x_{i-1}, a, x_{i+1}, \dots, x_n)$  for  $a \in \{0,1\}$ . A characteristic property of such system is that there exists no situation in which a failure (repair) of any system component can result in system repair (failure). The noncoherent systems do not have this property because their structure function is not non-decreasing. Typical instances of such systems are logic circuits or  $k$ -to- $l$ -out-of- $n$  systems.

The structure function describes the layout of the system components. Therefore, its analysis allows us to investigate topological properties of the system. However, the structure function itself cannot be used to investigate other, probabilistic, characteristics of the system, such as system availability and unavailability. For this task, state probabilities of the system components have to be known. For the  $i$ -th system component, these probabilities are denoted as follows:

$$p_i = \Pr\{x_i = 1\}, \quad q_i = \Pr\{x_i = 0\} = 1 - p_i, \quad (3)$$

and they are commonly known as component availability and unavailability, respectively, because they agree with proportion of time in which the component is working or failed, respectively.

Knowledge of system structure function and availabilities/unavailabilities of the system components allows us to compute some global characteristics, such as system availability and unavailability [1, 2]:

$$A = \Pr\{\phi(\mathbf{x}) = 1\}, \quad U = \Pr\{\phi(\mathbf{x}) = 0\} = 1 - A. \quad (4)$$

These measures are very important because they can be used to estimate mean time to failure or mean time to repair of the system [1]. On the other hand, they do not allow us to find components with the greatest influence on system activity. For this purpose, IMs are used.

### B. Importance Measures

IMs are a special kind of indices that are used to estimate importance of system components for the system activity. There exist a lot of IMs, but the most commonly known are SI, BI, and CI [2]. These IMs have originally been developed for investigation of coherent systems (Table I).

TABLE I. IMPORTANCE MEASURES FOR COHERENT SYSTEMS

Importance Measure	Meaning
SI	The SI focuses on topological properties of the system, and it is computed as a relative number of state vectors at which a failure of the $i$ -th system component results in system failure.
BI	The BI takes into account system topology and availabilities of system components, and it agrees with the probability that a failure of component $i$ results in system failure.
CI	The CI of component $i$ agrees with the probability that system failure has been caused by the component failure given that the system has failed.

As we can see from Table I, one of the common characteristics of these IMs is that their computation is based on identification of situations in which a failure of a given component results in system failure. These are known as situations in which component failure is critical for system failure, and they are described by so-called critical path vectors [1, 2]. However, in case of noncoherent systems not only failure of a given component can result in system failure but also its repair. Therefore, calculation of these measures for noncoherent systems requires the following steps [4, 5]:

1. identification of influence of a failure of a given component on system failure,
2. identification of influence of a repair of the component on system failure,
3. computation of total influence of a given component on system failure as a sum of the results obtained in the previous steps.

In this case, the SI agrees with the relative number of situations in which a change of a state of a given component results in system failure [6], the BI with the probability that system failure results from a change of a state of a given component [7], and the CI with the probability that system failure has been caused by a change of a state of a given component given that the system is failed [7].

### C. Logical Differential Calculus in Importance Analysis

According to the previous section, the most important thing in computation of the SI, BI, and CI for coherent and noncoherent systems is detection of situations at which a change of a state of the considered component results in a

change of system state. For this task, methods of logical differential calculus can be used.

Logical differential calculus is a special tool developed for analysis of dynamic properties of Boolean functions. The central term of this tool is logic derivative. There exist several types of logic derivatives, but for our purpose, the most important one is a DPBD, which identifies situations at which a given change of a given Boolean variable results in a given change of the value of the analyzed Boolean function [8, 9]. Since the formal definition of system structure function (1) corresponds to definition of Boolean function, this derivative can also be applied in investigation of the structure function. In this case, a DPBD of function  $\phi(\mathbf{x})$  with respect to variable  $x_i$  agrees with the following Boolean function [8, 9]:

$$\begin{aligned} \frac{\partial \phi(j \rightarrow \bar{j})}{\partial x_i(s \rightarrow \bar{s})} &= \{\phi(s_i, \mathbf{x}) \leftrightarrow j\} \{\phi(\bar{s}_i, \mathbf{x}) \leftrightarrow \bar{j}\} \\ &= \begin{cases} 1, & \text{if } \phi(s_i, \mathbf{x}) = j \text{ and } \phi(\bar{s}_i, \mathbf{x}) = \bar{j} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

where symbol " $\leftrightarrow$ " denotes logical biconditional. Please note that the nonzero elements of this derivative agree with state vectors  $(s_i, \mathbf{x})$  at which change of component  $i$  from state  $s$  to the opposite one results in change of system state from value  $j$  to  $\bar{j}$ . Specially, if  $j = 1$ , then this derivative permits us to find circumstances under which system failure is caused by a failure of component  $i$  (DPBD  $\partial \phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$ ) or by a repair of the component (DPBD  $\partial \phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$ ). Clearly, the latter cannot occur in a case of coherent systems and, therefore, only DPBDs  $\partial \phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  can be nonzero for such systems [10]. However, this is not true for noncoherent systems and, therefore, investigation of their failure requires computing not only DPBDs  $\partial \phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  but also DPBDs of the form of  $\partial \phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$ .

In the previous section, we have mentioned that the detection of circumstances under which a failure or repair of a given component results in system failure plays a key role in computation of the SI, BI, and CI. Since these situations can be found using the aforementioned DPBDs, the DPBDs can be used to compute these IMs. For coherent systems, this idea has been considered in [10]. Results from that paper have been generalized for noncoherent systems in [6, 7] where SI, BI, and CI have been defined in the forms presented in Table II. As we can see, in case of noncoherent systems, two partial measures can be defined for every IM. The first one deals with investigation of importance of a failure of a given component for system activity (e.g.  $SI_{i\downarrow}^\downarrow$ ), while the second quantifies importance of a repair of a given component for system failure (e.g.  $SI_{i\uparrow}^\downarrow$ ). If we want to compute the total importance of a given component, then we have to sum up these two partial measures [6, 7] (e.g.  $SI_i^\downarrow = SI_{i\downarrow}^\downarrow + SI_{i\uparrow}^\downarrow$ ). These facts imply that the most important thing in computation of the considered measures is an efficient identification of the nonzero values of

a DPBD. One of the possible ways of how this can be done is use of BDDs [13].

TABLE II. IMPORTANCE MEASURES FOR NONCOHERENT SYSTEMS

Importance Measure	Coherent Part (Importance of Component Failure)	Noncoherent Part (Importance of Component Repair)
SI	$SI_{i\downarrow}^\downarrow = \text{TD} \left( \frac{\partial \phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)} \right)$	$SI_{i\uparrow}^\downarrow = \text{TD} \left( \frac{\partial \phi(1 \rightarrow 0)}{\partial x_i(0 \rightarrow 1)} \right)$
BI	$BI_{i\downarrow}^\downarrow = \text{Pr} \left\{ \frac{\partial \phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)} = 1 \right\}$	$BI_{i\uparrow}^\downarrow = \text{Pr} \left\{ \frac{\partial \phi(1 \rightarrow 0)}{\partial x_i(0 \rightarrow 1)} = 1 \right\}$
CI	$CI_{i\downarrow}^\downarrow = BI_{i\downarrow}^\downarrow \frac{q_i}{U}$	$CI_{i\uparrow}^\downarrow = BI_{i\uparrow}^\downarrow \frac{p_i}{U}$

\*note: TD(.)... truth density of the argument interpreted as a Boolean function (i.e. relative number of vectors for which the argument has nonzero (true) value)

#### D. Binary Decision Diagram

A BDD (Fig. 1) is a graph structure developed for an efficient representation and manipulation with Boolean functions of high dimensions [11]. It has two types of nodes – sink nodes (leaves) and non-sink nodes. The sink nodes agree with the values of the Boolean function, while the non-sink nodes correspond to the variables of the Boolean function. A BDD has also one source (non-sink) node denoted as the root. Every non-sink node has 2 outgoing edges denoted by numbers 0 and 1, which represent possible values of the variable represented by the node. Every path from the root to leaf 0 (1) agrees with one or more vectors for which the Boolean function has value 0 (1). For example, the BDD in Fig. 1 represents function  $\phi(\mathbf{x}) = x_1(x_2 \vee x_3)$ . The green dotted path agrees with Boolean vector (1,0,1). Since this path ends in the 1-labeled sink node, the Boolean function takes value 1 for this vector. Similarly, the red solid path ending in the 0-labeled leaf corresponds to the Boolean vectors of the form of  $(0_1, \mathbf{x})$ , therefore, the function takes value 0 for vectors (0,0,0), (0,0,1), (0,1,0), and (0,1,1).

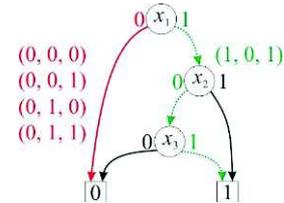


Figure 1. A BDD of function  $\phi(\mathbf{x}) = x_1(x_2 \vee x_3)$ .

Please note that structure function (1) can be viewed as a Boolean function. This implies that a BDD can also be used in reliability analysis to store this function [13]. For example, the BDD depicted in Fig. 1 represents the structure function of a series-parallel system depicted in Fig. 2 that is functioning if component 1 and at least one of components 2 and 3 are working.

A BDD is also very useful in fast computation of system availability/unavailability. These computations can be done simply by traversing the BDD. So, if we want to calculate system availability, then we have to traverse the BDD and

compute the probability that a root-leaf path ends in the 1-labeled leaf, e.g., the availability of the series-parallel system agrees with the following formula (Fig. 3):

$$A = p_1(q_2p_3 + p_2). \quad (6)$$

Similarly, system unavailability can be computed as the probability that a root-leaf path ends in the leaf labeled by number 0, e.g., the unavailability of the series-parallel system is calculated based on the BDD in Fig. 1 as follows (Fig. 3):

$$U = q_1 + p_1q_2q_3. \quad (7)$$

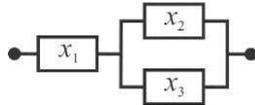


Figure 2. A simple series-parallel system.

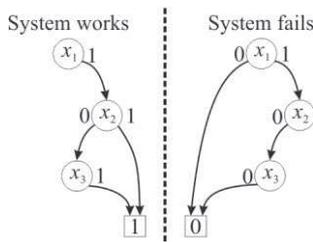


Figure 3. Computation of system availability and unavailability using a BDD.

As has been mentioned in the previous section, availability and unavailability are important characteristics of the system, but they do not allow us to investigate importance of individual system components. Such investigation requires computation of DPBDs. This problem has been studied in [13, 14], where several versions of algorithms for identification of the nonzero elements of a DPBD have been proposed. All those algorithms are based on definition (5) of a DPBD which implies that computation of DPBD  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$  can be done in such a way that we firstly identify a set of the state vectors  $(s_i, \mathbf{x})$  at which  $\phi(s_i, \mathbf{x}) = j$  and a set of the state vectors  $(\bar{s}_i, \mathbf{x})$  at which  $\phi(\bar{s}_i, \mathbf{x}) = \bar{j}$ . Then, we can remove the  $i$ -th element from every state vector from these two sets and find the state vectors  $(\cdot, \mathbf{x}) = (x_1, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$  for which DPBD  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$  is nonzero as the intersection of these sets of state vectors. If the structure function is defined in the form of a BDD, then this algorithm can be expressed in the following steps (Fig. 4):

1. traverse the BDD and identify all paths ending in a sink node labeled by  $j$  that leave the nodes corresponding to component  $i$  via  $s$ -labeled edge,
2. traverse the BDD and detect all paths ending in a sink node labeled by  $\bar{j}$  that leave the nodes corresponding to component  $i$  via edge labeled by  $\bar{s}$ ,
3. using rules defined in [13], compute intersection of the sub-diagrams obtained in steps 1 and 2. (Please note

that the result of this calculation is a BDD representing derivative  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$ , whose paths ending in 1-labeled sink node agree with the state vectors  $(\cdot, \mathbf{x})$  at which the DPBD is nonzero and whose paths terminated by 0-labeled sink node correspond to the state vectors  $(\cdot, \mathbf{x})$  at which the DPBD takes value 0.)

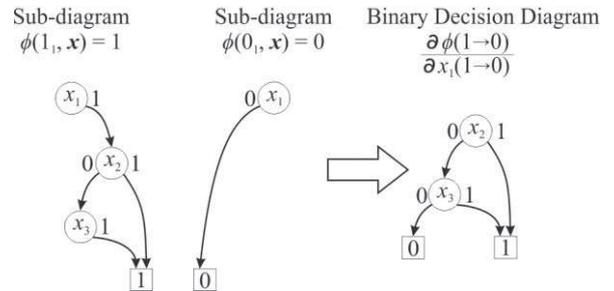


Figure 4. Identification of the nonzero elements of a DPBD using a BDD.

If a BDD corresponding to DPBD  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$  is known, then several IMs can be computed simply. For example, according to Table II, the  $BI_{i \downarrow}$  equals to the probability that DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  is nonzero. This probability can be calculated same as system availability in case of defining the structure function in the form of a BDD (Figure 3). This implies that BI (and also CI) measures can be computed very simply if we have a BDD corresponding to the DPBD based on which the IM is computed.

Several implementations of the algorithm for computation of a DPBD based on BDDs can be developed depending on the traversing used in steps 1 and 2 [13, 14]. However, the question is whether a BDD representing DPBD  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$  can be obtained in a simpler manner if the system has a special structure, e.g., series, parallel,  $k$ -out-of- $n$  in case of coherent systems, or  $k$ -to- $l$ -out-of- $n$  in case of noncoherent systems.

### III. BINARY DECISION DIAGRAMS FOR COMMON COHERENT SYSTEM STRUCTURES

#### A. Series System

A series system (Fig. 5) represents one of the simplest structures used in reliability analysis. This system works if and only if all system components are working and, therefore, its structure function can be expressed in the form of the following conjunction [1, 2]:

$$\phi_{ser}(\mathbf{x}) = \bigwedge_{i=1}^n x_i. \quad (8)$$

This implies that the structure function takes value 1 for only 1 state vector, i.e. for  $\mathbf{x} = (1, 1, \dots, 1)$ . Because of that, only 1 path in a BDD representing this structure function has to end in the 1-labeled sink node, and this path has to contain all  $n$  state variables and leaves each of them via 1-labeled edge. Next, every path leaving a non-sink node corresponding to state variable  $x_i$ , for  $i = 1, 2, \dots, n$ , through 0-labeled edge has to end in the 0-labeled leaf. This implies that a good BDD for a series

system contains only  $n$  non-sink nodes (one node for one state variable), and it can be expressed in the form depicted in Fig. 5. Please note that two forms of BDD are depicted in this figure. The first one is a classical BDD that has two sink nodes. The second is its expanded form. This form admits more than two sink nodes, but each of them has to be labeled by number 1 or 0. The main reason why we will use this form is its regularity in case of common structures studied in this paper, what will allow us to present our approach in more illustrative way.

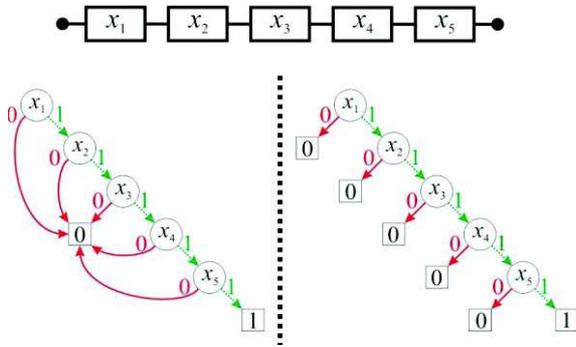


Figure 5. Example of a series system and its structure function expressed in the form of a BDD and expanded BDD.

B. Parallel System

A parallel system (Fig. 6) is another simple structure that occurs in reliability analysis often. This system is functioning if and only if at least one of the system components is working. This implies that its structure function can be expressed in the form of the following Boolean function [1, 2]:

$$\phi_{par}(\mathbf{x}) = \bigvee_{i=1}^n x_i. \tag{9}$$

One can easily realize that this system is nonworking if and only if all system components are failed. This implies that its structure function takes value 0 for only 1 state vector, i.e. for  $\mathbf{x} = (0,0,\dots,0)$ . So, a BDD representing this function has to contain only 1 path that ends in the 0-labeled sink node, and this path has to go through all  $n$  state variables and leaves each of them via 0-labeled edge. The result of this is that every path leaving any of the non-sink nodes via 1-labeled edge has to end in the 1-labeled sink node. So, a good BDD for this kind of systems contains exactly  $n$  non-sink nodes, and it can be expressed as a diagram shown in Fig. 6.

C.  $k$ -out-of- $n$  System

A  $k$ -out-of- $n$  system is the third typical structure used in reliability analysis. This system is working if and only if at least  $k$  system components out of  $n$  are working. This means that the structure function of this system takes value 1 for every state vector in which at least  $k$  state variables have value 1, i.e. for state vectors of the form of  $(1_{u_1}, 1_{u_2}, \dots, 1_{u_m}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-m}})$  and for  $m = k, k+1, \dots, n$ , where  $m$  denotes number of components that are in state 1;  $u_1, u_2, \dots, u_m$  are components that are in state 1; and  $v_1, v_2, \dots, v_{n-m}$  represent components that

are in state 0. Since  $\binom{n}{m}$  such state vectors exist for a given  $m$ , the structure function of  $k$ -out-of- $n$  system takes value 1 for  $\sum_{m=k}^n \binom{n}{m}$  and value 0 for  $\sum_{t=0}^{k-1} \binom{n}{t}$  state vectors.

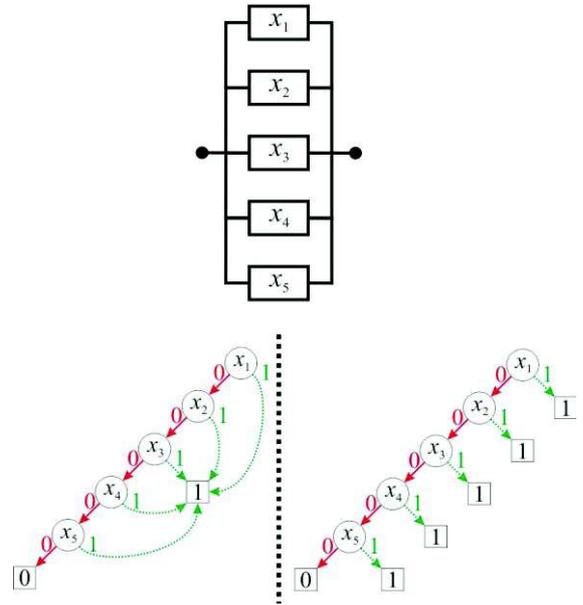


Figure 6. Example of a parallel system and its structure function expressed in the form of a BDD and expanded BDD.

A  $k$ -out-of- $n$  system is functioning if at least  $k$  components are working. This implies that, for a given state vector, at least  $k$  components have to be checked if we want to decide whether the system is working or not. This means that every path ending in the sink node 1 has to contain at least  $k$  non-sink nodes from which exactly  $k$  nodes have to be left via 1-labeled edges. If one or more checked components are in state 0, then we have to investigate other components. Clearly, if we check  $n - k + 1$  components, and all of them are in state 0, then the system has to be in state 0. This implies that every path containing  $n - k + 1$  0-labeled edges has to be terminated by 0-labeled sink node. It has been shown in several works, e.g. [15, 16], that a good BDD that has such properties corresponds to a lattice consisting of  $k$  columns and  $(n - k + 1)$  rows (Fig. 7).

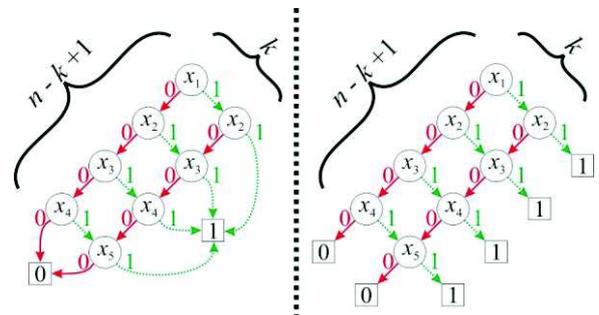


Figure 7. A BDD and expanded BDD for 2-out-of-5 system.

One can easily realize that the series and parallel systems composed of  $n$  components are special instances of  $k$ -out-of- $n$

systems at which  $k = n$  or  $k = 1$ , respectively. This implies that all results obtained for a  $k$ -out-of- $n$  system can be simply applied on a series or parallel system using the assumption that  $k = n$  or  $k = 1$ , respectively. Because of that, we will primarily deal with  $k$ -out-of- $n$  systems in what follows.

#### IV. BINARY DECISION DIAGRAMS FOR COMMON NONCOHERENT SYSTEM STRUCTURES

##### A. To- $l$ -out-of- $n$ System

Let us consider a special system that is composed of  $n$  components and that is working if and only if not more than  $l$  components are working. This system will be named as to- $l$ -out-of- $n$ .

To- $l$ -out-of- $n$  system is functioning if the states of the system components correspond to a state vector of the form of  $(1_{u_1}, 1_{u_2}, \dots, 1_{u_m}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-m}})$  for  $m = 0, 1, \dots, l$ , where  $m$  represents number of components that are in state 1;  $u_1, u_2, \dots, u_m$  agree with components that are in state 1; and  $v_1, v_2, \dots, v_{n-m}$  denote components that are in state 0. As in the case of  $k$ -out-of- $n$  systems, it can be shown simply that the structure function of to- $l$ -out-of- $n$  system takes value 1 for  $\sum_{m=0}^l \binom{n}{m}$  and value 0

for  $\sum_{t=l+1}^n \binom{n}{t}$  state vectors. One can easily realize that the to- $l$ -

out-of- $n$  system is a noncoherent system because if exactly  $l$  components are working then the system is working, but if any of the remaining  $n - l$  components begins working, the system fails. Please note that this system can be identified as the complementary system to  $(l+1)$ -out-of- $n$  system, i.e. to- $l$ -out-of- $n$  system is functioning for a state vector if and only if the complementary  $(l+1)$ -out-of- $n$  system is failed for this state vector. This fact implies if we have a BDD that represents  $(l+1)$ -out-of- $n$  system, then we can transform it into a BDD that will represent the complementary to- $l$ -out-of- $n$  system simply by interchanging sink nodes, i.e. the sink node labeled by number 1 will be labeled by number 0 and vice versa (Fig. 8). Since we are able to construct a good BDD for  $(l+1)$ -out-of- $n$  system (Fig. 7), we are also able to create a good BDD for to- $l$ -out-of- $n$  system (Fig. 8).

Please note that to- $l$ -out-of- $n$  system is not very realistic, but we introduced it mainly to present how to construct a good BDD for  $k$ -to- $l$ -out-of- $n$  system in more illustrative way.

##### B. $k$ -to- $l$ -out-of- $n$ System

A  $k$ -to- $l$ -out-of- $n$  system is a system that is working if and only if at least  $k$  but not more than  $l$  out of  $n$  components are working. Real examples of multiprocessor and transportation systems that behave as  $k$ -to- $l$ -out-of- $n$  systems have been given in [3]. In that paper, efficient method for computation of availability of such systems has been proposed. However, the authors have not considered how to analyze these systems using BDDs and how to investigate importance of the system components.

A  $k$ -to- $l$ -out-of- $n$  system can be viewed as a system created from two subsystems:  $k$ -out-of- $n$  and to- $l$ -out-of- $n$ . Using this

idea, the structure function of  $k$ -to- $l$ -out-of- $n$  system takes value 1 if the structure functions of both  $k$ -out-of- $n$  and to- $l$ -out-of- $n$  subsystems take values 1, while it has value 0 if at least one of the  $k$ -out-of- $n$  and to- $l$ -out-of- $n$  subsystems, from which the  $k$ -to- $l$ -out-of- $n$  system is composed, takes value 0. This implies that the structure function of  $k$ -to- $l$ -out-of- $n$  system can be expressed as the following Boolean formula:

$$\phi_{k\_l\_n}(\mathbf{x}) = \phi_{k\_n}(\mathbf{x}) \wedge \phi_{l\_n}(\mathbf{x}), \quad (10)$$

where  $\phi_{k\_n}(\mathbf{x})$  denotes the structure function of the  $k$ -out-of- $n$  subsystem and  $\phi_{l\_n}(\mathbf{x})$  represents the structure function of the to- $l$ -out-of- $n$  subsystem. This indicates that a good BDD for the  $k$ -to- $l$ -out-of- $n$  system could be obtained as a conjunction of good BDDs for  $k$ -out-of- $n$  (Fig. 7) and to- $l$ -out-of- $n$  (Fig. 8) systems. This conjunction can be done using several rules presented in [17]. According to these rules, the conjunction of two Boolean functions expressed in the form of BDDs can be done by composing the BDDs (Fig. 9). Therefore, a good BDD for the  $k$ -to- $l$ -out-of- $n$  system have the form depicted in Fig. 10.

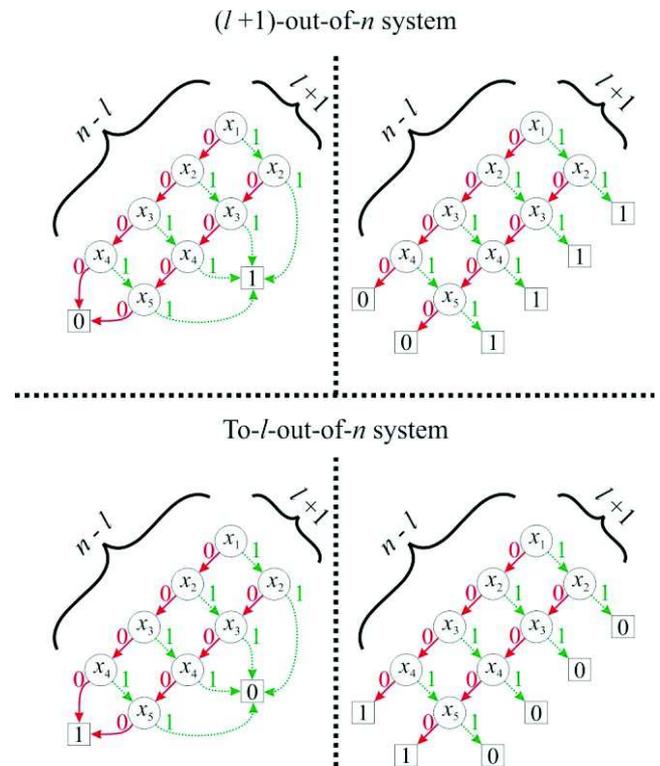


Figure 8. BDDs and expanded BDDs for 2-out-of-5 system and for the complementary to-1-out-of-5 system.

Please note that  $k$ -to- $l$ -out-of- $n$  system is the most general system from those considered in this paper. The others can be obtained from this system in the following ways:

- if  $k = 0$ , then we get to- $l$ -out-of- $n$  system,
- if  $l = n$ , then we obtain  $k$ -out-of- $n$  system,
- if  $k = 1$  and  $l = n$ , then we obtain parallel system,
- and if  $k = n$  and  $l = n$ , then we get series system.

Because of that, we will primarily focus on  $k$ -to- $l$ -out-of- $n$  systems in what follows. Clearly, the results that we obtain can be adapted simply to any of the previously mentioned systems using one the previous remarks.

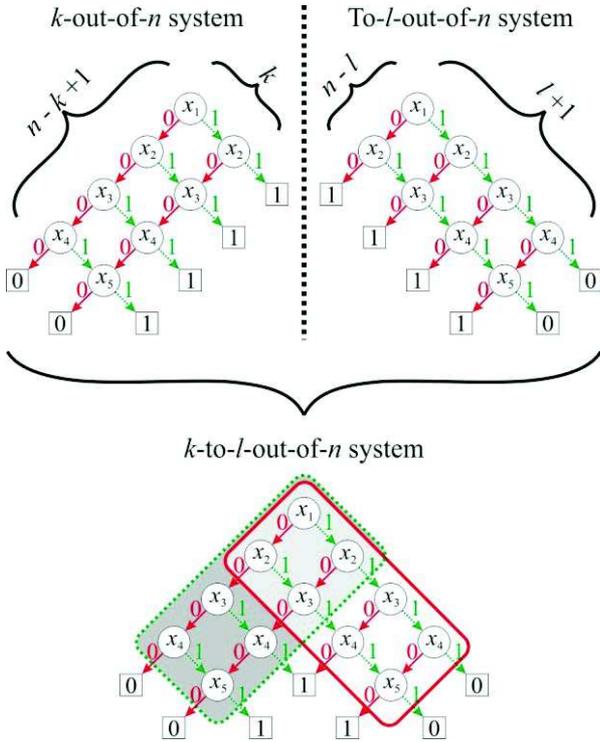


Figure 9. Construction of expanded BDD for 2-to-3-out-of-5 system from expanded BDDs for 2-out-of-5 and to-3-out-of-5 systems.

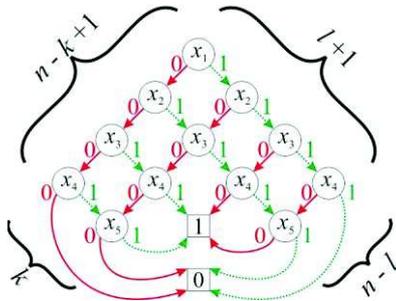


Figure 10. Final BDD for a 2-to-3-out-of-5 system.

V. IMPORTANCE ANALYSIS OF  $k$ -TO- $l$ -OUT-OF- $n$  SYSTEM BASED ON BINARY DECISION DIAGRAM

A. Direct Partial Boolean Derivatives and Binary Decision Diagrams

Let us consider a  $k$ -to- $l$ -out-of- $n$  system. One of the possible ways of how to investigate importance of individual system components is use of DPBDs.

Based on (5), a DPBD  $\partial\phi(j \rightarrow \bar{j})/\partial x_i(s \rightarrow \bar{s})$  investigating a system consisting of  $n$  relevant components can be viewed as a Boolean function of  $n-1$  variables, i.e. a Boolean function of

variables  $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ . This implies that it can also be expressed in the form of a BDD (Fig. 4). For this task, situations at which the DPBD is nonzero have to be identified. In case of  $k$ -to- $l$ -out-of- $n$  systems, this problem can be solved as follows.

Firstly, let us consider DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$ . This derivative identifies situations in which a failure of component  $i$  results in a failure of the system. Obviously, in case of  $k$ -to- $l$ -out-of- $n$  system, such situation can occur if and only if the  $i$ -th component is working and exactly  $k-1$  components different from component  $i$  are working. These situations correspond to state vectors of the form of  $(1_i, 1_{u_1}, 1_{u_2}, \dots, 1_{u_{k-1}}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-k}})$  where  $u_1, u_2, \dots, u_{k-1}$  denote components different from the  $i$ -th component that are in state 1; and  $v_1, v_2, \dots, v_{n-k}$  are components that are in state 0. Since a DPBD computed with respect to variable  $x_i$  does not depend on this variable (definition (5)), the derivative  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  investigating  $k$ -to- $l$ -out-of- $n$  system agrees with a Boolean function that takes value 1 for all vectors of the form of  $(1_i, 1_{u_1}, 1_{u_2}, \dots, 1_{u_{k-1}}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-k}})$ . Please note that such Boolean function can be viewed as the structure function of  $(k-1)$ -to- $(k-1)$ -out-of- $(n-1)$  system. This and the previous section imply that a good BDD representing DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  of  $k$ -to- $l$ -out-of- $n$  system will have the form depicted in Fig. 11. Using the fact that the relation  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0) = \partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$  exists between DPBDs [8, 9], it can be shown easily that a good BDD for DPBD  $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$  can also be constructed as a BDD storing the structure function of  $(k-1)$ -to- $(k-1)$ -out-of- $(n-1)$  system (Fig. 11).

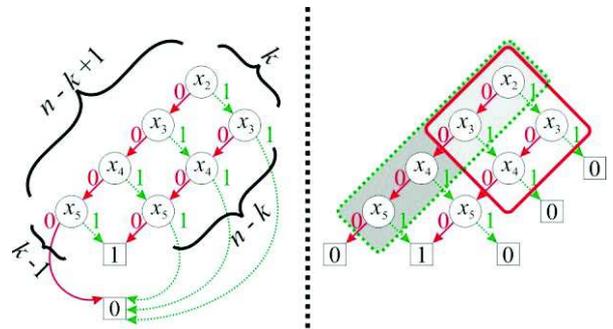


Figure 11. DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  ( $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ ) investigating 2-to-3-out-of-5 system expressed in the form of a BDD and expanded BDD.

Next, let us consider DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$ . This DPBD detects circumstances under which a repair of the  $i$ -th system component causes system failure. Such situations, which can exist only in noncoherent systems, occur in case of  $k$ -to- $l$ -out-of- $n$  system if and only if component  $i$  is nonworking and exactly  $l$  components different from component  $i$  are working. Clearly, these situations can be described by state vectors that have form of  $(0_i, 1_{u_1}, 1_{u_2}, \dots, 1_{u_l}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-l-1}})$  where  $u_1, u_2, \dots, u_l$  agree with components that are in state 1; and  $v_1, v_2, \dots, v_{n-l-1}$  denote components different from the  $i$ -th component that are in state 0. As mentioned above, the DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$  does not depend on variable  $x_i$  and,

therefore, it agrees with a Boolean function that takes value 1 for all vectors of the form of  $(\cdot, \cdot, 1_{u_1}, 1_{u_2}, \dots, 1_{u_l}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-l}})$  in case of investigation of  $k$ -to- $l$ -out-of- $n$  system. Since such function corresponds to the structure function of  $l$ -to- $l$ -out-of- $(n-1)$  system, a good BDD representing this DPBD can be obtained as a BDD for such system (Fig. 12). Finally, definition (5) of DPBD implies that the following relation exists:  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1) = \partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$ . It follows that a BDD for DPBD  $\partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$  agrees with a BDD representing derivative  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$  and, therefore, a good BDD for this derivative corresponds to a good BDD for  $l$ -to- $l$ -out-of- $(n-1)$  system assuming that the derivative investigates  $k$ -to- $l$ -out-of- $n$  system (Fig. 12).

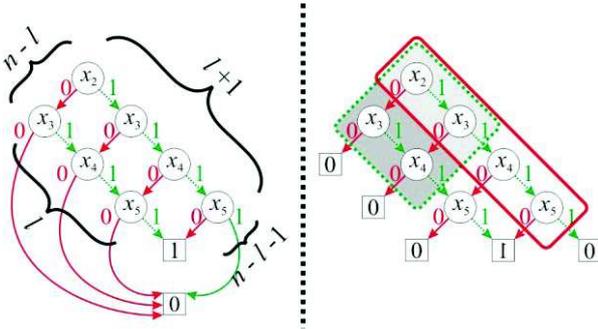


Figure 12. DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$  ( $\partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$ ) investigating 2-to-3-out-of-5 system expressed in the form of a BDD and expanded BDD.

### B. Computation of Importance Measures based on Binary Decision Diagrams

DPBDs can be used to compute several IMs, e.g., SI, BI, CI (Table II). If a DPBD is expressed in the form of a BDD, then computation of these measures requires traversing the BDD and identifying the sub-diagram with the 1-labeled sink node (section II.D). Since the traversing agrees with the visit of every node of the BDD, running time of this algorithm depends on a number of the non-sink nodes. In case of DPBDs  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  and  $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ , this number equals to  $(n-k)k + (k-1)$  (Fig. 11) while, in case of DPBDs  $\partial\phi(1 \rightarrow 0)/\partial x_i(0 \rightarrow 1)$  and  $\partial\phi(0 \rightarrow 1)/\partial x_i(1 \rightarrow 0)$ , it is equal to  $(n-l)l + (n-l-1)$  (Fig. 12).

The BDDs could also be used to compute the SI measures. To show this, let us consider a Boolean function  $f(\mathbf{z})$  of  $w$  Boolean variables, i.e.  $\mathbf{z} = (z_1, z_2, \dots, z_w)$ , and assume that value true agrees with number 1 and value false with number 0. If Boolean variable  $z_r$  takes value 1 with the probability  $p_r$ , for  $r = 1, 2, \dots, w$ , then the probability of function  $f(\mathbf{z})$  being 1 can be computed as follows:

$$\Pr\{f(\mathbf{z}) = 1\} = \sum_{\mathbf{z} \in \{0,1\}^w} f(\mathbf{z}) \prod_{r=1}^w (p_r z_r + (1-p_r)(1-z_r)). \quad (11)$$

Next, let us assume that  $p_r = 0.5$  for all Boolean variables, i.e. for  $r = 1, 2, \dots, w$ . It follows that the previous formula will have the following form:

$$\begin{aligned} \Pr\{f(\mathbf{z}) = 1\} &= \sum_{\mathbf{z} \in \{0,1\}^w} f(\mathbf{z}) \prod_{r=1}^w (0.5z_r + 0.5(1-z_r)) \\ &= \frac{1}{2^w} \sum_{\mathbf{z} \in \{0,1\}^w} f(\mathbf{z}) = \text{TD}(f(\mathbf{z})), \end{aligned} \quad (12)$$

what implies that the truth density of a Boolean expression can be calculated as the probability that the Boolean expression has value 1 assuming that every Boolean variable of the expression takes value 1 with the probability 0.5.

Based on the previous paragraphs, the SI measures can be computed in the same way as the BI measures, i.e. by traversing a BDD representing an appropriate DPBD assuming that all state variables (the non-sink nodes) take value 1 with the same probability as value 0, i.e. with the probability 0.5. However, much more efficient approach exists in case of  $k$ -to- $l$ -out-of- $n$  systems.

Let us consider the  $\text{SI}_{i\downarrow}^\downarrow$ . This measure agrees with the truth density of DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  (Table II). It was shown in section V.A that this derivative is nonzero for the state vectors of the form of  $(\cdot, \cdot, 1_{u_1}, 1_{u_2}, \dots, 1_{u_{k-1}}, 0_{v_1}, 0_{v_2}, \dots, 0_{v_{n-k}})$ , i.e. for all state vectors  $(\cdot, \cdot, \mathbf{x})$  in which exactly  $k-1$  variables have value 1. Clearly,  $\binom{n-1}{k-1}$  such state vectors exist. Since DPBD  $\partial\phi(1 \rightarrow 0)/\partial x_i(1 \rightarrow 0)$  is a Boolean function of  $n-1$  variables (the previous section), its truth density can be computed in the following way:

$$\text{TD}\left(\frac{\partial\phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)}\right) = \frac{1}{2^{n-1}} \binom{n-1}{k-1} \quad (13)$$

and, therefore, the SI investigating consequences of a failure of component  $i$  on system failure from topological point of view in case of  $k$ -to- $l$ -out-of- $n$  system can be calculated based on the following formula:

$$\text{SI}_{i\downarrow}^\downarrow = \frac{1}{2^{n-1}} \binom{n-1}{k-1}. \quad (14)$$

Using the same procedure as in the case of the  $\text{SI}_{i\downarrow}^\downarrow$ , it can be shown simply that the  $\text{SI}_{i\uparrow}^\downarrow$  can be obtained using the next expression:

$$\text{SI}_{i\uparrow}^\downarrow = \frac{1}{2^{n-1}} \binom{n-1}{l}. \quad (15)$$

We have obtained two closed-form expressions that allow us to investigate topological importance of components in case of  $k$ -to- $l$ -out-of- $n$  systems and all their variations, such as  $k$ -out-of- $n$ , to- $l$ -out-of- $n$ , series, and parallel systems. Clearly, such investigation is not meaningful in case of one system because the SI measures show us only that all system components have the same topological importance, but it can be useful if we want to analyze several different systems and compare their topological properties.

## VI. CONCLUSION

One of the current issues of reliability engineering is analysis of systems that are composed of many components. Investigation of such systems requires development of new approaches that allow us to represent them in a reasonable form. One of the possible ways is application of BDDs.

BDDs have been developed for efficient representation of and manipulation with Boolean functions. In case of reliability analysis, they can be used to store information about system structure. However, the main issue behind BDDs is how to create a good BDD for a given Boolean function or for a given system structure.

In this paper, we considered several common structures that are frequently used by reliability engineers. We showed that all these structures, i.e. series, parallel, and  $k$ -out-of- $n$  systems, can be viewed as special instances of more general systems referred to as  $k$ -to- $l$ -out-of- $n$  systems, which are functioning if at least  $k$  but not more than  $l$  components are working. By introducing a special type of system that is referred to as  $l$ -out-of- $n$  system, we showed how a good BDD could be constructed for  $k$ -to- $l$ -out-of- $n$  system. This result was used in the remaining part of the paper, which focused on importance analysis of  $k$ -to- $l$ -out-of- $n$  systems. Using logical differential calculus and BDDs, we developed efficient methods for calculation of the most commonly used IMs for such systems (Table III). These measures and methods developed for their computation can be applied in the analysis of different types of complex systems, such as offshore installations considered in [18] or temporal database systems studied in [19, 20]. Furthermore, they could also be used in other research fields, such as knowledge discovery in databases and data mining, where they can be used to investigate importance of individual attribute in datasets [21].

TABLE III. TIME COMPLEXITY OF COMPUTATION OF IMPORTANCE MEASURES BASED ON THE PROPOSED METHODS

Importance Measure	Method	Coherent Part	Noncoherent Part
SI	Closed-form expression	$O(1)$	$O(1)$
BI	BDD	$O(kn)$	$O(ln)$
CI	BDD	$O(kn)$	$O(ln)$

## REFERENCES

- [1] M. Rausand and A. Høyland, *System Reliability Theory*, 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc., 2004.
- [2] W. Kuo and X. Zhu, *Importance Measures in Reliability, Risk, and Optimization: Principles and Applications*. Chichester, UK: Wiley, 2012.
- [3] S. J. Upadhyaya and H. Pham, "Analysis of noncoherent systems and an architecture for the computation of the system reliability," *IEEE Transactions on Computers*, vol. 42, no. 4, pp. 484–493, Apr. 1993.
- [4] J. D. Andrews and S. Beeson, "Birbaum's measure of component importance for noncoherent systems," *IEEE Transactions on Reliability*, vol. 52, no. 2, pp. 213–219, Jun. 2003.
- [5] S. Beeson and J. D. Andrews, "Importance measures for non-coherent-system analysis," *IEEE Transactions on Reliability*, vol. 52, no. 3, pp. 301–310, Sep. 2003.
- [6] J. Kostolny, M. Kvassay, and S. Kovalik, "Reliability analysis of noncoherent systems by logical differential calculus and binary decision diagrams," *Komunikacie*, vol. 16, no. 1, pp. 114–120, Feb. 2014.
- [7] M. Kvassay, E. Zaitseva, J. Kostolny, and V. Levashenko, "Reliability analysis of noncoherent systems based on logical differential calculus," in *ESREL 2016*, accepted.
- [8] S. B. Akers, "On a theory of Boolean functions," *Journal of the Society for Industrial and Applied Mathematics*, vol. 7, no. 4, pp. 487–498, Dec. 1959.
- [9] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook*, vol. 2. Boca Raton, FL: CRC Press, 2005.
- [10] E. N. Zaitseva and V. G. Levashenko, "Importance analysis by logical differential calculus," *Automation and Remote Control*, vol. 74, no. 2, pp. 171–182, Feb. 2013.
- [11] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, August 1986.
- [12] A. Kaufmann, D. Grouchko, and R. Croun, *Mathematical Models for the Study of the Reliability of Systems*. New York, NY: Academic Press, 1977.
- [13] E. Zaitseva, V. Levashenko, and J. Kostolny, "Importance analysis based on logical differential calculus and Binary Decision Diagram," *Reliability Engineering & System Safety*, vol. 138, pp. 135–144, Jun. 2015.
- [14] J. Kostolny, M. Kvassay, and S. Kovalik, "Analysis of system reliability by logical differential calculus and decision diagrams," in *The International Conference on Digital Technologies 2013*, 2013, pp. 170–175.
- [15] S. M. Li, S. B. Si, S. Zhang, and H. Y. Dui, "Decision diagrams method for  $k$ -out-of- $n$ : G systems," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013 International Conference on*, 2013, pp. 263–267.
- [16] L. Xing and S. V. Amari, *Binary Decision Diagrams and Extensions for System Reliability Analysis*, Hoboken, NJ: John Wiley & Sons, Inc., 2015.
- [17] T. Sasao and M. Fujita, Eds., *Representations of Discrete Functions*. Norwell, MA: Kluwer Academic Publishers, 1996.
- [18] R. Briš, "Risk modelling of personnel fatality at accidents on offshore installations," in *Information and Digital Technologies (IDT), 2015 International Conference on*, 2015, pp. 44–52.
- [19] M. Kvet, K. Matiaško, and M. Kvet, "Transaction Management in Fully Temporal System," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 148–153.
- [20] M. Kvet, "Index distribution management in attribute level temporal model," in *2015 International Conference on Information and Digital Technologies*, 2015, pp. 206–214.
- [21] Zaitseva, E., Kvassay, M., Levashenko, V., Kostolny, J.: "Introduction to knowledge discovery in medical databases and use of reliability analysis in data mining," in *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015*, 2015, pp. 311–320.