UNIVERSITY OF ŽILINA
FACULTY OF MANAGEMENT SCIENCE AND INFORMATICS

RELIABILITY ANALYSIS OF SYSTEMS BASED ON MULTIPLE-VALUED LOGIC

Dissertation thesis

28360020203009

Žilina 2020 **Ing. Andrej Forgáč**

UNIVERSITY OF ŽILINA
FACULTY OF MANAGEMENT SCIENCE AND INFORMATICS

RELIABILITY ANALYSIS OF SYSTEMS BASED ON MULTIPLE-VALUED LOGIC

Dissertation thesis

Registration Number: 28360020203009

Study program:	Applied Informatics
Field of study:	Informatics
Department:	Department of Informatics Faculty of management science and informatics, University of Zilina in Zilina
Supervisor:	prof. Ing. Elena Zaitseva, PhD.

Žilina 2020 **Ing. Andrej Forgáč**

Acknowledgement

I would like to express my special thanks of gratitude to my advisor prof. Ing. Elena Zaitseva, PhD. for her patience, willingness, motivation, comments and valuable advice on my work. Her guidance helped me in all the time of research and writing of this thesis.

I declare that I have written the submitted dissertation thesis independently under the guidance of the supervisor using the bellow-mentioned literary sources and the theoretical knowledge from domestic as well as foreign authors and my own experiences and knowledges gained during the study.

Abstrakt v štátnom jazyku

Forgáč Andrej, Ing.: Analýza spoľahlivosti systémov na základe použitia viachodnotovej logiky (dizertačná práca). Žilinská univerzita v Žiline. Fakulta riadenia a informatiky. Katedra Informatiky. - Vedúca práce: prof. Ing. Elena Zaitseva, PhD. - Žilina FRI UNIZA, 2020. 102 s.

V tejto práci je navrhnutý rozvoj matematických metód pre manipuláciu so štruktúrnou funkciou. Analýza ľubovoľného systému začína definovaním počtu stavov a potom pokračuje vývojom matematického opisu systému. Preto sú tieto dva kroky najdôležitejšie a až po ich spracovaní môžeme vykonať ďalšiu analýzu zložitých systémov.

Hlavným cieľom práce je vývoj a zdokonaľovanie matematického prístupu analýzy spoľahlivosti MSS v kroku vytvárania matematickej interpretácie skúmaného systému vo forme štruktúrnej funkcie s uplatnením matematického prístupu viachodnotovej logiky.

Tento cieľ nás vedie k preskúmaniu a vyvíjaniu metód na zostrojenie minimálnej a ortogonálnej formy štruktúrnej funkcie dvojstavového a viacstavového systému. Takáto forma reprezentácie umožňuje priamy prechod od logického k pravdepodobnostnému opisu a následne k spoľahlivosti systému.

Kľúčové slová: analýza spoľahlivosti, štruktúrna funkcia, viachodnotová logika

Abstrakt v cudzom jazyku

Forgáč Andrej, Ing.: Reliability analysis of systems based on multiple-valued logic [dissertation thesis]. - University of Žilina; Faculty of Management Science and Informatics; Department of Informatics. - Supervisor: prof. Ing. Elena Zaitseva, PhD. - Žilina FRI UNIZA, 2020. pp. 102

In this work, the development of mathematical methods for manipulation of structure function is proposed. The analysis of any system starts by defining the number of states and then proceeding to the development of a mathematical description of the system. Therefore these 2 steps are the most important and only after their processing we can do further analysis of complex systems.

The principal goal of this work is the development and improvement of mathematical approach of multi state system reliability analysis in step of construction of mathematical representation of investigated system in form of the structure function with the application of the mathematical approach of Multiple-Valued Logic.

This goal causes the investigation for the development of methods to construct the minimal and orthogonal form of the structure function of binary and multi state system. Such a form of representation allows moving from a logical to a probabilistic description and then gets to the reliability of the system.

Key words: Reliability Analysis, Structure Function, Multiple-Valued Logic

Contents

List of Figures

List of Tables

Used Acronyms

MSS - Multi-State System

BSS - Binary-State System

MVL - Multiple-valued logic

MDD - Multi-Valued Decision Diagram

BDD - Binary Decision Diagram

IM - Importance Measures

SI - Structural Importance

DNF - Disjunctive normal form

SDNF – perfect disjunctive normal form

ODNF – Orthogonal disjunctive normal form

BDNF - Non-repetitive disjunctive normal form

PF - Probabilistic function

RBD - Reliability block diagram

CU - Control unit

MD - Main drone

RD - Redundant drone

Introduction

The present status and level of technology cause new trends and conditions in the development of Reliability Engineering. There is a wide range of tasks which are not typically for the reliability engineering which cannot be decided by the application of traditional methods. For example, such tasks are assessing the risk of a terrorist attack (Levitin 2009), business analysis reliability (Solojentsev 2009), estimate the risk and consequences of the technological accidents (Zio 2009), and many others. At the same time modern technologies allow for a virtually fail-free operation of technical part of complex systems. These conditions cause modification of traditional approaches and methods and development of new methods in Reliability Engineering (Birolini 2014, Ushakov 2006, Zio 2009). Prof. E. Zio in the review of Reliability Engineering (Zio 2009) defined reliability engineering as "a well-established, multi-disciplinary scientific discipline which aims at providing an ensemble of formal methods to investigate the uncertain boundaries between system operation and failure, by addressing the following questions:

- Why systems fail, e.g. by using the concepts of reliability physics to discover causes and mechanisms of failure and to identify consequences;
- How to develop reliable systems, e.g. by reliability-based design;
- How to measure and test reliability in design, operation and management;
- How to maintain systems reliable, by maintenance, fault diagnosis and prognosis."

According to the analysis in (Zio 2009) old problems of reliability engineering as:

- the mathematical representation of the system;
- the system quantification analysis;
- the representation, propagation and quantification of the uncertainty in system behavior,

should be considered for the new condition. Therefore we can see that the investigation of mathematical representation of the system is a relevant problem of reliability engineering. These problems cause that tasks have become actual again. Principal steps for the development of the mathematical representation of the system in reliability engineering are (Bris 2014, Aven 2017):

1. the definition of the number of performance levels for a system model;
2. the mathematical representation of the system model;
3. the quantification of the system model (calculation of indices and measures);
4. the measuring of the system behavior.

The first and second step in tasks of reliability analysis correlates with the initial data. The goal of these steps is to construct the mathematical model for reliability evaluation. Therefore these two steps are considered in this work first of all.

In the **first step,** the approach for a general representation of the system is defined. There are two approaches (Figure 1) for system representation that are description based on Multi-State System (MSS) (Barlow, 1978) and Binary-State System (BSS) (Barlow, 1975). BSS allows representing the initial system as a mathematical model with two possible states that are a complete failure and perfect working. MSS permits to consider more than only two states in the behavior of system reliability or availability.

According to (Lisnianski, 2003) conceptions as availability, reliability, and system states can be expressed as "performance level" for MSS. The use of MSS allows analysis of system reliability in more detail, but this analysis is more complicated (Natvig 2010, Lisnianski 2003).

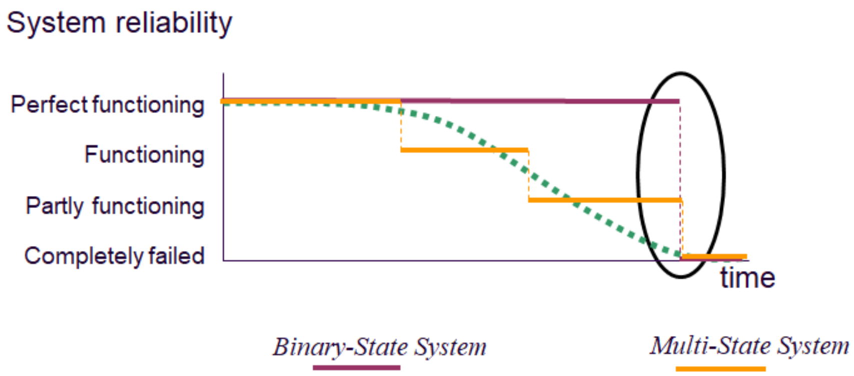


Figure 1 Binary and multi state system

MSS doesn't use in reliability analysis widely because it has two restrictions. First of them is computational complexity (Lisnianski, 2003). The introduction in the analysis of additional system performance levels and components states causes dramatically increase in the dimension of this mathematical representation. The second is the lack of effective methods and algorithms for estimation (qualitative and quantitative analysis) of MSS

(Aven 2014, Birolini 2014, Zio 2009). Therefore the investigation and development of the MSS reliability analysis is an actual problem in Reliability Engineering now.

The algorithms for MSS evaluation depend on the mathematical methods used for system analysis. In (Lisnianski, 2003) authors have indicated four principal groups of mathematical methods of MSS analysis that cause MSS type as structure function, Markov model, universal generation function and mathematical model based on Monte-Carlo simulation. Each of these types of MSS has some advantages. The important advantages of structure function are simplicity of construction, the possibility of application for the system with structural complexity, and simple methods for reliability indices calculation based on the methods of Algebra logic.

A typical approach for MSS structure function analysis is a generalization of methods for BSS structure function analysis that are based on Boolean logic as a rule (Barlow 1975, Barlow 1978, Birolini 2014). However, this approach does not allow using all details of MSS. Another approach is based on the application of Multiple-Valued Logic (MVL) mathematical methods for MSS structure function analysis (Zaitseva 2017, Zaitseva 2012, Kvassay 2017). According to this approach, the structure function is explained as MVL function (Zaitseva 2017). The methods based on MVL mathematical methods of MSS structure function manipulation and analysis are proposed and considered for system availability calculation in (Zaitseva 2017, Zaitseva 2015), analysis of critical system states in (Kvassay 2017, Kvassay 2014), importance analysis (Zaitseva 2012, Zaitseva 2015).

The exact mathematical representation is developed in the **second step** in reliability analysis that is caused by the mathematical methods that will be used for the evaluation of the investigated object/system. Structure function methods of MSS and BSS reliability analysis being older and are widely used in engineering practice and different applications (Barlow 1978, Murchland 1975). Important advantages of the structure function are (Kolowrocki 2014, Lisnianski 2018, Natvig 2010):

- the definition of univalent correlation of system performance level and components states;
- the representation of the system of any structural complexity;
- the complexity of system representation is not dependent on its structure.

An important problem for further development and the use of MSS is the insufficient mathematical basis for MSS analysis.

The structure function is one of the basic representations of MSS. However, the dimension of the structure function increases critically depending on the number of system components (Zaitseva, 2013). The development of structure function methods should be based on orthogonalization and minimalization. In order to use the structural function without complications with probability theory, the logical function of structure function representation must be orthogonal and minimal (Solojentsev, 2009).

The mathematical representation type is caused by the detail (accuracy) of system evaluation and mathematical approach that is used for the calculation of reliability indices and measures.

The quantification of a system in the **third step** expects calculation of individual system reliability indices and measures, such as Reliability function, failure rate, mean time to failure, mean time to repair, mean time between failures, fault coverage, availability, unavailability, importance measures, etc. (Lisnianski, 2010). The mathematical representation of system and chosen methods in second step determine the algorithms and methods for calculation of these indices and measures. Algorithms and methods for computation of system reliability indices and measures depend on representation by the structure function are considered in (Murchland, 1975; Barlow 1978; Natvig 2010).

After we calculate system reliability indices and measures, we could analyze them. Measurement and improvement of system reliability is done in the **fourth step** to develop strategies for increasing system reliability, maintainability and other reliability properties of the system.

As it follows from the analysis of the principal steps of mathematical representation construction, the mathematical representation of any system is started by defining the number of states and the development of a mathematical description of the system, which closely correlates with the mathematical methods used for the system evaluation. Therefore these 2 steps are the most important for us and only after their processing we can do further analysis of complex systems.

In this work, the analysis of BSS and MSS are considered. The analysis of possible mathematical descriptions proposed above allows us to choose the structure function, because this mathematical description can be constructed for the system of any structural complexity (Griffith 1980, Lisnianski & Levitin 2003). The structure function defines an univalent correlation of a system performance level and component states. The structure function based methods have been developed by many investigations (Levitin, 2009, Zio 2019, Ushakov 2006). The mathematical methods for the structure function evaluation are often based on the methods of algebra logic. In case of BSS, these methods are developed with the application of Boolean Logic (Wood 1985, Schneeweiss 2009, Ryabinin 1981). The MSS structure function evaluation is implemented using Multiple-Valued Logic (Zaitseva 2017, Rauzy, 2001). The important condition of most of the structure function based methods is a representation of structure function as the orthogonal form (Schneeweiss 2009, Ryabinin 1981, Rauzy, 2001). This form is important for the structure function representation because it allows a very simple transformation of logical interpretation of the structure function into the probability form (Ryabinin 1981, Griffith 1980, Reinske & Ushakov1988, Schneeweiss 2009, Sellers & Singpurwalla 2008). Most reliability indices and measures (availability, unavailability, importance measures and other) can be computed based on the probability form only (Figure 2). Therefore, the computation of reliability indices and measures need the probability form of the structure function, which can be obtained based on a logical orthogonal form of the structure function. It causes the development of algorithms for orthogonalization of initial structure function (Ryabinin 1981).

The problem of the orthogonalization of logical function is a typical problem in algebra logic (Miller & Aaron 2008, Stankovic, Astola & Moraga 2012). There are some orthogonal forms for logical functions. One of well-known forms is perfect disjunctive normal form. The important disadvantage of this form is large dimension that agrees with a number of non-zero values of Boolean function (Ryabinin 1981, Smirnov & Gajdamovich 2001, Rausand & Hoyland 2007). Therefore, the logical function is typically minimizing and then orthogonalization is implemented for such function (Ryabinin 1981, Wood 1985). There are some methods for the orthogonalization of Boolean function that can be effectively used for the forming of a orthogonal structure function in reliability analysis of BSS. In particular, it is a method proposed by Prof. A. Ryabinin in (Ryabinin 1981) based on the construction of the special matrix transformation. But this method application cannot be well used for the function of a large dimension. According to an evaluation in (Ryabinin 1981), this method can be used for function with 20 variables, that means the analysis of BSS with 20 components only. The analysis of other investigations in orthogonalization of logical function shows, that the approach proposed by Prof. A. Zakrevskij and Prof. Yu. Pottosin in (Zakrevskij & Pottosin 2005) can be used for function with a large dimension and can be adopted for structure function reliability analysis. But this method has been developed for the Boolean function only. We need to note that the problem of orthogonalization in Multiple-Valued Logic is not decided.

The problem of orthogonalization in Multiple-Valued Logic correlates closely with the problem of logical function minimization because the Multiple-Valued Logic functions have large dimension (Petrik 2008). Therefore, the MSS structure function orthogonalization should include the minimization of this function in case this function is formed as a disjunctive normal form. Some adaptation and interpretation of the orthogonalization problem similarly to Boolean logic have been proposed by Prof. M. Perkowski (Perkowski 1992). These investigations should be developed for the application in the reliability analysis of MSS. Therefore, orthogonalization is a relevant problem that should be considered in the actual investigation of reliability engineering and first of all, this problem should be considered in the analysis of MSS (Sellers & Singpurwalla 2008).

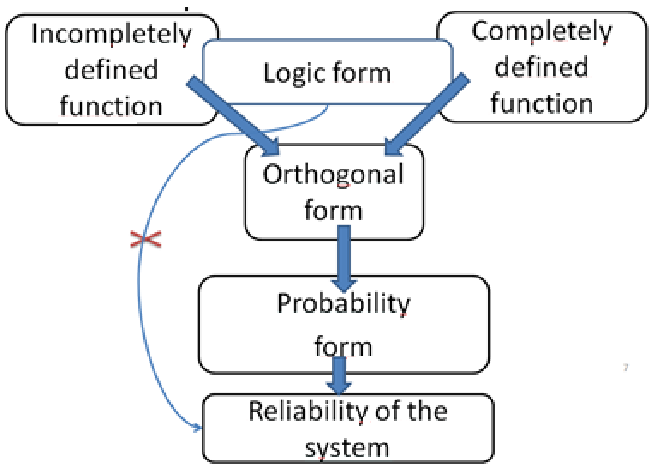


Figure 2 The transition from the logical form to the reliability of the system

The principal goal of this work is the development and improvement of a mathematical approach of MSS reliability analysis in step of the construction of mathematical representation of investigated system in form of the structure function with the application of the mathematical approach of Multiple-Valued Logic. This goal causes the investigation for the development of methods to construct the orthogonal form of the structure function of BSS and MSS. The development of such method results decision tasks:

- continuation investigation in (Zakrevskij & Pottosin 2005) and development algorithm for the orthogonalization of BSS structure function based on the method proposed by authors in (Zakrevskij & Pottosin 2005);
- analysis of the conception of orthogonal form for the Multiple-Valued Logic function and definition of the conception of the orthogonalization for MSS structure function;
- development of algorithms for the MSS structure function minimization and orthogonalization;
- validation of the developed algorithms for BSS and MSS orthogonalization on selected systems (BSS structure functions);
- analysis of the efficiency of the proposed algorithms.

1 Structure function based methods

The quantitative reliability evaluation of any system is possible based on a mathematical representation of the investigated system.

1.1 Structure function

Structure function as a mathematical model has historically been designed as the first model, and the performance level of the system is defined depending on all component states. The structure function is one of the possible mathematical models representing real systems in the theory of Reliability Engineering. The structure function declares a system performance level (reliability/availability) depending on its components states (Natvig 2010, Zio 2009):

$$\phi(\boldsymbol{x})=\phi(x_1,\ldots, x_n): \{0,\ldots,m_1 -1\}\times\ldots\times\{0,\ldots,m_n -1\}\rightarrow\{0,\ldots,M -1\}, \quad (1)$$

where $\phi(\boldsymbol{x})$ is the state of the system from its failure ($\phi(\boldsymbol{x}) = 0$) to perfect functionality ($\phi(\boldsymbol{x}) = M - 1$); $\boldsymbol{x} = (x_1,\ldots, x_n)$ is a state vector; x_i is the state of the component that changes from failure state ($x_i = 0$) to perfect functionality ($x_i = m_i -1$).

A system with structure function (1) is a MSS and allows us to represent and explore some performance levels of the system. If $M = m_i = 2$ structure function (1) represents a BSS that allows us to analyze two system states: failure and perfect functioning. The structure function (1) can be represented as a classification model. According to this representation, all vectors of system states (x_1, ..., x_n) are divided into M classes (Zaitseva, 2016).

According to the mathematical definition (1) the variable of the structure function is interpreted as a component of the system. The structure function allows the representation of different systems.

The structure function has different properties depending on the type of system examined. In this work, coherent systems are considered, i.e.:

- structure function is monotone: $\phi((s -1)_i, \boldsymbol{x}) \leq \phi(s_i, \boldsymbol{x})$ for any $i \in \{1,\ldots, n\}$ and $s \in \{1,\ldots, m_i -1\}$;
- components contained in the system are not irrelevant, where $\phi(s_i, \boldsymbol{x})=(x_1,\ldots,x_{i-1},s,x_{i+1},\ldots,x_n)$.

Evaluation of MSS based on the structural function expects an expression of the probability of individual states for each component of the system.

For illustration how structure function can be used for the representation of human impact let to represent a simple laparoscopic surgery procedure for reliability evaluation. According to (Levashenko, 2016), this procedure can be interpreted as MSS that consists of four components (n = 4): device (a laparoscopic robotic surgery machine (Patel, 2009)), two doctors (anesthesiologist and surgeon) and a nurse. This system as MSS introduces the numbers of states for every component and number of performance levels of the system. The function of the simple laparoscopic surgery procedure can be interpreted as a multiple-valued logic (MVL) function of four variables (n = 4) with three values (m = 3). Let this system has three performance levels: 0 – non-operational (fatal medical error); 1 – partially operational (some imperfection); 2 – fully operational (surgery without any complication). The device (x_1) has three states: 0 – failure; 1 – partially functioning; 2 – functioning. The work of anesthesiologist (x_2), surgeon (x_3) and the nurse (x_4) can be modeled both by 3 levels, i.e.: 0 – (the fatal error); 1 – (sufficient); 2 – (perfect or the work without any complication).

The structure function of the system of this simplified version of laparoscopic surgery is composed of 81 situations (state vectors). The structure function according to expert knowledge and evaluations can be represented by Table 2 (Zaitseva, 2016). In the case of incompletely specified initial data, some of the state vectors are not indicated. For example, we can suppose that information about this system is incomplete and represented by 66 states only (Table 1).

Table 1 The Structure Function of Laparoscopic Surgery Procedure Success

x_3x_4 / x_1x_2	**00**	**01**	**02**	**10**	**11**	**12**	**20**	**21**	**22**
00	0	0	0	0	*	*	0	0	*
01	0	*	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	*	1
11	0	0	0	0	1	*	0	2	2
12	0	0	*	1	1	*	*	2	2
20	0	*	0	0	0	1	0	*	*
21	0	0	1	1	1	2	1	*	2
22	0	0	1	1	*	2	1	*	2

Table 2 The completely specified Structure Function of Laparoscopic Surgery Procedure Success

x_1x_2 \ x_3x_4	**00**	**01**	**02**	**10**	**11**	**12**	**20**	**21**	**22**
00	0	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1
11	0	0	0	0	1	2	0	2	2
12	0	0	0	1	1	1	1	2	2
20	0	0	0	0	0	1	0	0	1
21	0	0	1	1	1	2	1	2	2
22	0	0	1	1	1	2	1	2	2

One more definition of the structure function Laparoscopic Surgery Procedure Success can be in the form of BSS. In this case, all system components (a laparoscopic robotic surgery machine, anesthesiologist, surgeon and nurse) behavior have two states only: that means successful function or error. This BSS is shown in Table 3.

Table 3 Structure Function of Laparoscopic Surgery Procedure Success for BSS

x_1x_2 \ x_3x_4	**00**	**01**	**10**	**11**
00	0	0	0	0
01	0	0	0	1
10	0	0	0	0
11	0	0	1	1

Methods of assessing the reliability of a system based on the representation of structure functions are firmly established. These methods are deterministic and are used in quantitative and qualitative analysis. The structure function can be created based on fully specified data that indicates the correlations of all components and their states. Such data for most real-world systems is incomplete and uncertain. A typical example is human factor analysis and evaluation.

The BSS structure function is interpreted as the Boolean function. This function describes the logical linking of elements in the system, but does not allow analyzing probability conditions - it is a logic function - the exact function does not allow us to say anything about system reliability – the probability that system performs its functions during a defined time assuming that it worked at the beginning. It is important to move from logical form through an orthogonal form to probability form and get to the reliability of the system. MSS will be interpreted as a function of multi-value logic.

The probability of system performance level is defined for every performance level as:

$$A_j = \Pr\{\phi(x) = j\}, j = 1, \dots, M - 1. \quad (2)$$

In papers (Barlow 1978, Hudson 1983, Lisnianski 2003) authors showed that any system state s ($j = 1, \dots, M - 1$) for fixed components state vector of a coherent MSS according to the assumption can be calculated as the product of probabilities of components states:

$$p_{is} = \Pr\{x_i = s\}, s = 0, \dots, m_i - 1. \quad (3)$$

As was shown in the investigation (Barlow 1978, Hudson 1983), the structure function (1) can be used to calculate the system availability (2) if the variables of the structure function describe the independent events. It is possible if the structure function is canonical and orthogonal form. Two theorems of probability theory are used for the calculation of the system availability (2):

1. The probability of the product of independent events a and b (simultaneous happening) is equal to the product of the probabilities of these events:

$$Pr(ab) = Pr(a)Pr(b). \quad (4)$$

2. The probability of the sum of incompatible events a and b (at least one of them happening) is equal to the sum of the probabilities of these events:

$$Pr(a + b) = Pr(a) + Pr(b). \quad (5)$$

Practical application of two theorems (4) and (5) supposes the change of the variables x_i ($i = 1, \dots, n$) of the structure function (1) by the probabilities of system components states (3) if the structure function is described as canonical and orthogonal form. In (Caldarola, 1980) it is shown that in the interpretation of coherent MSSs, the probability of system state j ($j = 0, \dots, m-1$) can be computed for fixed state vector $\boldsymbol{x} = (x_1,\dots,x_n)$ as a product of probabilities $\Pr\{x_i = s\}$ of components states, where $s = 0, \dots, m_i-1$ defines possible states of component i. One of the conditions of the non-coherent system is that the variables are independent of each other. As we consider the variables, their functionality is independent of each other and therefore we can use the rule (4) to analyze them. That means that if we work with 2 variables and calculate the probability of the state when the first variable failed and the second variable is in state 1 (working), we just use the probability of the first variable (failed) and the probability of the second variable (working) and we apply multiplication because these variables are independent events. If it is considered to take all possible states for which the system will fail, these states must be incompatible (5) among each other and that means one variable cannot be in the same state for a working and fail system.

For example, the availability of the Laparoscopic Surgery Procedure Success for BSS is defined according to Table 3:

$$A_1 = p_{10} \cdot p_{21} \cdot p_{31} \cdot p_{41} + p_{11} \cdot p_{21} \cdot p_{31} \cdot p_{40} + p_{11} \cdot p_{21} \cdot p_{31} \cdot p_{41} = p_{21} \cdot p_{31} \cdot (p_{11} + p_{41} - p_{11} \cdot p_{41}) \quad (6)$$

Availability (6) can be calculated for various probability values of the components in structure function, for instance in Table 4 is shown for specific values.

Table 4 Availability for specific values.

p_{21}	p_{31}	p_{11}	p_{41}	Availability (A_1)
0,2	0,15	0,3	0,25	0,01425
0,35	0,5	0,1	0,1	0,03325
0,7	0,3	0,9	0,55	0,20055
0,85	0,6	0,15	0,3	0,20655
0,2	0,15	0,75	0,85	0,028875

Therefore, an important aspect of the system availability calculation is the construction of the canonical and orthogonal form of the structure function. This aspect can be considered based on the methods of the Boolean Logic for BSS and based on the Multiple-Valued Logic for MSS.

1.2 Representation of system based on structure function

The structure function can be represented in different forms. As a rule, for the structure function we use the typical tree representations in Reliability Engineering:

- Table
- Graphical
- Analytical (formulas)

1.2.1 Table representation

The table representation of the Boolean or BSS structure function is possible in a form of the truth table or truth table column vector. The table form of representation of a structure function is orthogonal. The truth table is a representation of all possible values of functions depending on all possible values of function variables. For example, Table 1, 2 (for MSS), and 3 (for BSS) are typical truth tables for the structure function of Laparoscopic Surgery Procedure Success. More steps that contain an approach to the Boolean truth table (Mohd, 2006) are:

1. Identify all major system components where the fault leads to a system failure and its collapse.
2. Creating a reliability model for a system that demonstrates how the components or modules are connected.
3. Determining the probability of success, P (Success) and the probability of failure, P (Failure) components or modules from its failure rate, (A).
4. The state of the system is identified in relation to the states of its components. If the system identifies success, the result is that system up, or system identifies failure, the result is that system down.

Another representation of a structure function of BSS by table could be by a **ternary matrix**. Consider a certain technical system, the correct functioning of which is influenced by certain events, called a set of basic events $D = \{d_1, d_2, ..., d_m\}$. They are mapped by Boolean variables $d_1, d_2, ..., d_m$, which take the value 1 if and only if the related event happens. On the set D of basic events, the critical sets $K_1, K_2, ..., K_t$ are defined, the minimal subsets of D such events, the simultaneous occurrence of which breaks the correct functioning of the system. Let the events forming a certain set of K be realized simultaneously. The system fails if and only if among the given critical sets there is a set K_j of them that is contained in K: $Kj \subseteq K$. From the minimality of critical sets, it follows that they do not absorb each other. Each critical set Kj can be represented as a positive elementary conjunction of k_j variables from the set $D = \{d_1, d_2, ..., d_m\}$. Then the complex event R we expressed by disjunction of these conjunctions:

$$R = k_1 \vee k_2 \vee ... \vee k_t. \tag{7}$$

This formula represents a monotonic Boolean function. It is suitable to represent it by a ternary matrix T whose columns correspond to basic events and the rows represent (by units) distinct critical sets K_j. They can also be interpreted as the corresponding positive elementary conjunctions k_j and their characteristic sets are given by intervals of the space of Boolean variables $d_1, d_2, \ldots, d_m$.

For example, in such an interpretation, the matrix R can be considered as DNF.

Table 5 Interpretation of the matrix R as DNF

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	
	-	-	-	1	-	1	-	1
	-	-	-	1	1	-	-	2
R=	1	1	-	-	-	-	1	3
	-	1	1	1	-	-	-	4
	1	-	1	-	-	1	1	5
	-	1	-	-	1	1	-	6

This DNF contains information that the operability of the technical system in question depends on the events $d_1, d_2, d_3, d_4, d_5, d_6, d_7$ forming critical sets:

$$\{ d_4, d_6 \}, \{ d_4, d_5 \}, \{ d_1, d_2, d_7 \}, \{ d_2, d_3, d_4 \}, \{ d_1, d_3, d_6, d_7 \}, \{ d_2, d_5, d_6 \}. \quad (8)$$

For example, if d_1, d_2, *and* d_3 occur at the same time, the system will continue properly. If d_3, d_4, *and* d_5 occur, the system fails because a critical set $\{d_4, d_5\}$ is performed.

1.2.2 Graphical representation

Graphical representation of the structure function can be as:

- Multi-Valued networks
- Cube-based representation
- Decision trees and diagrams

One of often the used representations of a structure function in Reliability Engineering is a decision diagram. In the case of MSS, it is Multi-Valued Decision Diagram (MDD) and for the BSS structure function is used Binary Decision Diagram (BDD). These representations in Algebra Logic have been introduced by (Akers, 1978). Both BDD and MDD form of representation of structure function is orthogonal.

An MDD is a directed acyclic graph of structure function representation (Zaitseva, 2012). The graph has *m* sink nodes, labelled from 0 to (*m*-1), representing *m* corresponding constant from 0 to (*m*-1). As a rule, these nodes show as a rectangular block in MDD. Each non-sink node is labelled with a structure function variable *x* and has *m* outgoing edges. Variables are in a round block of MDD. In an MSS reliability analysis, the sink node is interpreted as a system reliability state from 0 to (*m*-1) and the non-sink node presents either a system component. Each non-sink node has *m* edges, and the first (left) is labelled the "0" edge and agrees with component fail, and the *m*-th last outgoing edge is labelled "*m*-1" edge and presents the perfect operation state of system component.

Paths from the top non-sink node to zero-sink node are used to analyse an MSS failure. Paths from the top non-sink node to another sink node are considered for system repair by means of MDD. The example of MDD for a laparoscopic surgery procedure Success is shown in Figure 3. The BDD of this system is presented in Figure 4.

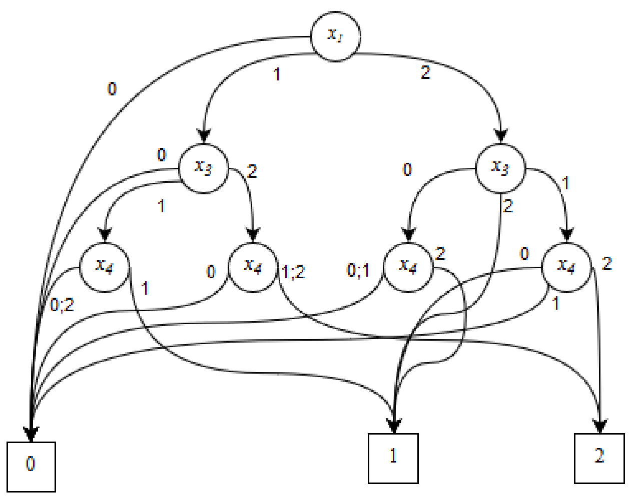


Figure 3 The MDD of the structure function of a laparoscopic surgery procedure success

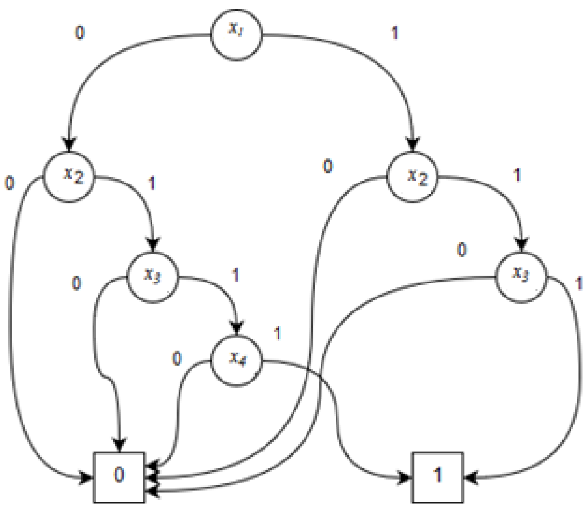


Figure 4 The BDD of the structure function of laparoscopic surgery procedure success

1.2.3 Analytical (formulas) representation

The next type of a structure function representation is the description of the structure function by the different types of formulas:

- Multi-Valued sum-of-product
- Reed-Muller Expansion
- Arithmetic representation

We need to note that the analytical representation of MSS structure function is a complex problem because the dimension of this function increases extremely depending on the number of component states and the number of components. For example, the structure function of laparoscopic surgery procedure success is represented by the logical polynomial form as:

$$\phi(\boldsymbol{x}) = x_2\, x_3\, x_4^2 + 2x_2\, x_3^2 x_4^2 + 2x_2^2 x_3\, x_4 + x_2^2 x_3^2 x_4^2 + 2x_1\, x_2\, x_3 + 2x_1\, x_2\, x_3\, x_4 + 2x_1 x_2 x_3 x_4^2 + 2x_1 x_2 x_3^2 + 2x_1\, x_2\, x_3^2 x_4 + x_1\, x_2^2 x_3^2 x_4^2 + x_1\, x_2^2 x_3 + x_1\, x_2^2 x_3\, x_1\, x_2^2 x_3\, x_4^2 + x_1\, x_2^2 x_3^2 + x_1\, x_2^2 x_3^2 x_4 + 2x_1^2 x_3\, x_4 + 2x_1^2 x_3\, x_4^2 + 2x_1^2 x_3^2 x_4 + 2x_1^2 x_2\, x_3 + 2x_1^2 x_2\, x_3\, x_4 + x_1^2 x_2\, x_3\, x_4^2 + 2x_1^2 x_2\, x_3^2 x_4 + x_1^2 x_2\, x_3^2 x_4^2 + x_1^2 x_2^2 x_3 + 2x_1^2 x_2^2 x_3\, x_4 + 2x_1^2 x_2^2 x_3^2 x_4 + 2x_1^2 x_2^2 x_3^2 x_4^2\ (mod\ 3), \quad (9)$$

and at the same the structure function of BSS for this system in logical polynomial form is:

$$\phi(\boldsymbol{x}) = x_1x_3x_4 + x_1x_2x_4 + x_1x_2x_3x_4 \; (mod\ 2). \tag{10}$$

Therefore, BSS is acceptable for the system representation if the detailed behavior of the components and the system reliability is not required. The analytical form for a BSS structure function must be canonical and orthogonal. The canonical form guarantees a single-valued mathematical representation and the orthogonal form permits us to use a probabilistic evaluation. The orthogonal form is required in order to calculate the reliability.

The BSS structure function is interpreted as a Boolean function (Schneeweiss 2009, Dutuit 2001). Therefore, the background of the Boolean algebra for reliability analysis should be considered in this work.

1.3 Reliability indices calculated based on SF

1.3.1 Availability for BSS

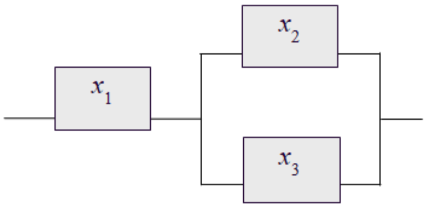
The system availability is a significant reliability measure, characterized as a probability that system works. For a BSS availability can be computed build on structure function as:

$$A = \Pr\{\phi(x) = 1\}. \tag{11}$$

In reverse, a system unavailability is defined as a probability that the system fails. Unavailability of the system can be calculate based on a structure function as:

$$U = 1 - A = \Pr\{\phi(x) = 0\}. \tag{12}$$

Example: We have a system with 3 components x_1, x_2, x_3 , which is defined in Figure 5 by a structure function represented in the form of table.



x_1	x_2	x_3	$\phi(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figure 5 System with 3 components

1. First we need to indicate the structure function values 1: $\phi(\mathbf{x}) = 1$

   In our case, it is $\phi(\mathbf{x}) = 1$: (1, 0, 1), (1, 1, 0) and (1, 1, 1).

2. Then we add the state vectors (sets of variables values) to the sum: $(x_1 \dots x_n)=1$.

   Pr $\{\Sigma(x_1 \dots x_n)=1\}$: Pr{(1,0,1) + (1,1,0) + (1,1,1)}

3. Then we compute the probability of every state vector: $\Pr\{(x_1 \dots x_n)\}$

   $\Pr\{(x_1 \dots x_n)\} = \Pr\{x_1\} \cdot \dots \cdot \Pr\{x_n\}$

   $\Pr\{x_i\} = p_i$, if $x_i=1$ and $\Pr\{x_i\} = q_i$, if $x_i=0$

   In our case, it is: $\Pr\{(1,0,1)\} = p_1 \cdot q_2 \cdot p_3$, $\Pr\{(1,1,0)\} = p_1 \cdot p_2 \cdot q_3$, $\Pr\{(1,1,1)\} = p_1 \cdot p_2 \cdot p_3$

4. Finally, we can computed the availability as:

   $A = p_1 \cdot q_2 \cdot p_3 + p_1 \cdot p_2 \cdot q_3 + p_1 \cdot p_2 \cdot p_3 = p_1 \cdot q_2 \cdot p_3 + p_1 \cdot p_2 \cdot (q_3 + p_3) = p_1 \cdot (q_2 \cdot p_3 + p_2)$

   $A = p_1 \cdot (p_2 + p_3 + p_2 \cdot p_3)$

1.3.2 Availability for MSS

Availability for MSS must be calculated separately for each system performance level. The probability of the system performance level can be computed by a formula (2).

Then, the unavailability of the system can be defined as the probability of the performance level when system fails, it is for $j=0$ and for this reason, the unavailability of the system can be represented as follows:

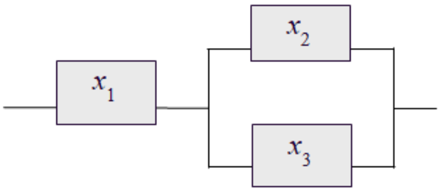
$$U = A_0 = \Pr\{\phi(x) = 0\}. \quad (14)$$

Availability for MSS is the probability that system reliability is greater than or equal to the system performance level j:

$$A_j = \Pr\{\phi(x) \geq j\} = \sum_{w=1}^{j} A_w, \qquad \text{for } j = 1, \ldots, m-1. \qquad (15)$$

Example:

We have an MSS with 3 components x_1, x_2, x_3, which is defined in Figure 6 by a structure function represented in form of a table. We want to calculate the availability A_2 of for the performance level j=2.



x_1	x_2	x_3	$\phi(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	2
2	0	0	0
2	0	1	1
2	1	0	1
2	1	1	2

Figure 6 MSS with 3 components

1. First, we must calculate the probability of system for each performance level it is for P_1 *and* P_2

For instance, P_2 is computed as follows:

1.1 First, we indicate the structure function values j: $\phi(\boldsymbol{x})$ = j

In our case, it is $\phi(\boldsymbol{x})$ = 2: (1, 1, 1) and (2, 1, 1).

1.2 Then we add to sum the state vectors (sets of variables values): $(x_1 \ldots x_n)$ = j.

Pr $\{\Sigma(x_1 \ldots x_n)=2\}$: Pr{(1,1,1) + (2,1,1) }

1.3 Finally, we compute the probability of every state vector: Pr$\{(x_1 \ldots x_n)\}$

$\Pr\{(x_1 \ldots x_n)\} = \Pr\{x_1\} \cdot \ldots \cdot \Pr\{x_n\}$

$\Pr\{x_i\} = p_{is}$, if $x_i = s$, $s = \{0, \ldots, m_i - 1\}$

In our case, it is: $\Pr\{(1,1,1)\} = p_{11} \cdot p_{21} \cdot p_{31}$, $\Pr\{(2,1,1)\} = p_{12} \cdot p_{21} \cdot p_{31}$

1.4 Finally, $P_2 = p_{11} \cdot p_{21} \cdot p_{31} + p_{12} \cdot p_{21} \cdot p_{31} = p_{21} \cdot p_{31} \cdot (p_{11} + p_{12})$

With the same procedure we calculate:

$P_1 = (p_{11} + p_{12}) \cdot (p_{20} \cdot p_{31} + p_{21} \cdot p_{30})$

2. Now we can represent the availability A_1, A_2:

   $A_1 = P_1 = (p_{11} + p_{12}) \cdot (p_{20} \cdot p_{31} + p_{21} \cdot p_{30})$

   $A_2 = P_1 + P_2 = (p_{11} + p_{12}) \cdot (p_{20} \cdot p_{31} + p_{21} \cdot p_{30} + p_{21} \cdot p_{31})$

3. We can also calculate P_0 and then represent the unavailability U:

   $U = P_0 = p_{10} + p_{20} \cdot p_{30} \cdot (p_{11} + p_{12})$

1.3.3 Importance Measures

Every component of the system has various impacts on system performance. Importance analysis analyzes the calculation of the impact of these components on the system performance. Importance analysis could be quantitative and qualitative. The qualitative analysis examines situations which can improve or downgrade of the system performance. Quantitative analysis is focused on evaluating the importance of system components regarding system performance. Determination of components with the highest impact on system performance is significant in the planning the system maintenance or optimization of system availability (Kvassay, 2017). Indices for the estimation of the influence of component states change into the system reliability are named in reliability analysis as Importance Measures (IM). **IMs** are probabilities that characterize how to change system reliability if the i-th system component state changes.

Structural Importance (SI) is one of the simplest measures of component importance, because it ignores any consideration of the individual reliability of components; instead it concentrates topological characteristics of the system based on its structure function. It is used for analyzing systems under design when we do not know the entire structure of the system. SI of the MSS for i-th component state s is the probability that this system performance level j decrements if the component state changes from s_i to s_{i-1} depending on the topological properties of the system:

$$SI_i^{s,j} = \frac{p_i^{s,j}}{p_s}, \tag{16}$$

where $p_i^{s,j}$ is the number of system states when a change of the i-th component state from s to s-1 causes a change of performance level of the system from j to j-1 and p_s is the number of system states for which $\phi(s_i, x) = j$.

Example: We have MSS defined by structure function represents by Table 6:

Table 6 Example of MSS

x_1	x_2	$\phi(x)$
0	0	0
0	1	1
0	2	1
1	0	1
1	1	2
1	2	2

It is expected that the states of components and system can downgrade only by one performance level. For example, if we want to calculate SI for a condition when component 1 downgrade by one level from 1 to 0 and system performance level downgrade from 2 to 1 it can be computed as:

$$SI_1^{1,2} = \frac{p_1^{1,2}}{p_1} = \frac{2}{2} = 1 \tag{17}$$

Another example could be the degradation of component 2 from state 2 to 1 and also system performance level downgrade from 2 to 1:

$$SI_2^{2,2} = \frac{p_1^{2,2}}{p_2} = \frac{0}{0} = 0 \tag{18}$$

It means that the condition when component 2 downgraded by one level has no impact on system performance.

2 Methods for structure function construction

2.1 Boolean Algebra

The Boolean algebra in abstract algebra is defined as a complementary and distributive union. This type of algebraic structure contains the basic properties of both set and logical operations.

Boolean Logic is a form of algebra which is centered around three simple words known as Boolean Operators: “Or,” “And,” and “Not”. In Boolean algebra there are defined operations of conjunction and intersection, and unary complement operation is also defined here (Verma, 2010).

2.1.1 Basic concepts of sets theory and algebra of logic

It is important to investigate and solve many issues that arise in reliability theory, the methods of theory sets and algebra of logic, probability theory and mathematical statistics. In the set theory, the sets are formed from elements that have certain properties and are among themselves or with elements of other sets in some relations. A special section of the general set theory - algebra of sets - considers various operations on sets with any elements (Zaitseva, 2003).

If we want to specify that an object *a* is one of elements of the set A, we use the sign ∈:

$$a \in \mathrm{A}, \tag{19}$$

a is included or contained in the set A. If the object *a* does not occur among the elements of the set A, then write:

$$a \notin \mathrm{A}, \tag{20}$$

a is not included or not contained in the set A. Let two sets A and B be considered. If each element of the set A also belongs to the set B, then A is a part or a subset of the set B. This is written down with a sign of inclusion:

$$\mathrm{A} \subset \mathrm{B}. \tag{21}$$

Equal sets (for example A, B) are called identical sets that are sets consisting of the same elements. It is obvious that for equal sets simultaneously A ⊂ B a B ⊂ A.

Sometimes, when determining a set, you can not know if it contains at least one element.

The notion of an empty set that does not contain a single element is denoted by Ø. An empty set is considered a subset of any set that is the inclusion $Ø \subset A$ and it is valid for any set A.

The union of two sets A and B is called set, denoted by

$$A \cup B, \qquad (22)$$

which consists of all elements entering at least one of the sets A or B. The intersection (or common part) of two sets A and B is called set, denoted by:

$$A \cap B, \qquad (23)$$

consisting of all elements that enter both A and B.

Two sets A and B are said to be disjoint (or incompatible), if they do not have common elements. Incompatibility condition (or orthogonality) of the set A to B symbolically is denoted by $A \cap B = Ø$.

$$A \setminus B \qquad (24)$$

is a subset of the set A consisting of all elements of A that do not occur in B. Moreover, the definition of the difference $A \setminus B$ does not need to be contained in $B \subset A$.

The use of concepts and symbols of set theory allows us to give a clearer and concise mathematical description of the foundations of the theory reliability. The algebra of logic, which includes the calculus of propositions, or Boolean algebra, is a branch of mathematical logic. Mathematical logic operates on statements and studies questions of representation and transformation of binary functions from binary arguments through some logical operations, called logical connections. The function f (x_1, ..., x_n) defined on sets of the form (x_1, ..., x_n), in which the variables x_i can take the values 0 or 1, and variety on these values, is called the function of the algebra of logic. Simple statements through logical connections can be used to make complex statements that have the meaning of truth: "true" (1) or "false" (0) - depending on the truth values of simple statements. Connections between statements can be represented as operations on binary variables. The basic logical operations are defined in the next subchapter. Different dependencies between statements, considered in mathematical logic can be divided into two groups: elementary and complex logical functions. The latter are obtained from the first by their repeated use in a wide variety of combinations. The sequence of elementary logic functions is usually written using parentheses. In the algebra of logic (more precisely, in the algebra of propositions) we consider three basic logical operations: negation, conjunction (multiplication) and disjunction (addition). Subtraction and division in the algebra of logic are absent. Using the algebra of logic equations, we can describe the conditions of the technical system, including digital devices. The equations show which elements and which compounds can create a specific digital device. It's obvious that the latter is directly related to the solution of various tasks from the field reliability of technical systems (Ryabinin I. A., 1981).

2.1.2 Basic logical operations

It is necessary to define certain basic concepts. The union $\mathbb{L}$ is formed by a partially ordered set $\mathbb{L}$**,** where for each two elements *x*, *y* of the set $\mathbb{L}$ is defined a minimum (minimum value) and a maximum (maximum value). We can write (Dunn, 2001):

$$(\forall x, y \in \mathbb{L})(\exists i, s \in \mathbb{L})\big((i = \min(x, y)) \wedge (s = \max(x, y))\big). \tag{25}$$

The minimum operation is also called an intersection and it is marked with a $\wedge$, while the maximum is used to denote $\vee$ and to name the conjunction. The union $\mathbb{L}$ can then be written in the form $(\mathbb{L},\wedge,\vee)$.

A distributive union is such a union in which for all *x*, *y* and *z* belonging to the set $\mathbb{L}$ the following relations apply:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \qquad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \tag{26}$$

The bounded union L = $(\mathbb{L},\wedge,\vee,0,1)$ is a union $(\mathbb{L},\wedge,\vee)$, where for the constants 0,1 belonging to the set $\mathbb{L}$ relations are valid:

$$\forall x \in \mathbb{L}, x \wedge 0 = 0 \text{ and at the same time } x \vee 0 = x \tag{27}$$

$$\forall x \in \mathbb{L}, x \wedge 1 = x \text{ and at the same time } x \vee 1 = 1 \tag{28}$$

Constant 1 is called an upper boundary or maximum of the union $\mathbb{L}$ and constant 0 is called a lower boundary or minimum of union $\mathbb{L}$.

The complementary union is the algebraic structure $(\mathbb{L}, \wedge, \vee, \neg, 0, 1)$, where $(\mathbb{L}, \wedge, \vee, \neg, 0, 1)$ is a bound union and for each element x belonging to the set $\mathbb{L}$ there is a complementary element $\neg x$ belonging to set $\mathbb{L}$, and:

$$x \wedge \neg x = 0 \tag{29}$$

$$x \vee \neg x = 1 \tag{30}$$

Based on all previous definitions, it is possible to define Boolean algebra as follows: The Boolean algebra is arranged by sixths $(\mathbb{A}, \wedge, \vee, \neg, 0, 1)$, where $\mathbb{A}$ is the set on which the binary operations of the intersection are denoted by the symbol $\wedge$ and the conjunction denoted by the symbol $\vee$, the complementary operation of the complement marked with the symbol $\neg$ and the two elements 0 and 1 (called minimum and maximum, also denoted by the symbol $\perp$ or $\top$), that for all elements a and b of the set $\mathbb{A}$ the following axioms apply (Hilary, 2002):

1. Associativity

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z \qquad x \vee (y \vee z) = (x \vee y) \vee z \tag{31}$$

2. Commutativity

$$x \wedge y = y \wedge x \qquad a \vee b = y \vee x \tag{32}$$

3. Law of absorption

$$x \wedge (x \vee y) = x \qquad x \vee (x \wedge y) = x \tag{33}$$

4. Neutral element

$$x \wedge 1 = x \qquad x \vee 0 = x \tag{34}$$

5. Distributivity

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \qquad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \tag{35}$$

6. Complement

$$x \wedge \neg x = 0 \qquad x \vee \neg x = 1 \tag{36}$$

It is clear that Boolean algebra uses binary operations of intersection (also known as AND), conjunction (also called OR) and unary complement operation (also known as NOT). However, in Boolean algebra, other operations are also used, which can be expressed by a combination of three basic operations. These operations are representations that transform the two input variables $x, y \in \{0,1\}$ into the third output variable $z = f(x, y) \in \{0,1\}$, with a total of 16 different possible views because for each combination of input f (0,0), f (0,1), f (1,0) and f (1,1) there are 2 options for the output variable values (Donald, 2008).

Operation NOT, i.e. negation or complement operation, is a unary operation whose output is a complement to the operand, which in Boolean algebra means that negating the value 0 is 1 and negating the value 1 is 0. However, the negation can also be used for other operations (NAND, NOR, ...).

AND operation in Boolean algebras is a binary operation of intersection whose output is the minimum of two operand values. This means that this operation has a value of 1 at output when both operands have a value of 1, otherwise the output is 0.

The OR operation in Boolean algebra is a binary operation of conjunction whose output is the maximum of two operand values. Obviously, this operation has a value of 0 at the output just when both operands have a value of 0, otherwise the operation is a value of 1.

Operation NOR, also known as Pierce's function, is a Boolean algebras binary operation that is a negation of an OR operation. This means that this operation has an output value of 1 just when both operands have a value of 0. Otherwise, it has a value of 1 at the output.

NAND operation, also known as Sheffer's function, is a Boolean algebras operation in the Boolean algebras, which is a negation of the AND operation. Therefore, it is obvious that this operation gives the value 0 at the output only if the value of both operands is equal to 1, otherwise the output is 1.

The operation of equivalence is a binary operation defined in Boolean algebra that compares the value of two operands, and if their values are equal, the output is 1, otherwise the output is 0.

Last, the XOR operation, which is a binary operation defined in the Boolean algebra, represents the opposite of the operation of equivalence. This means that the output of this operation is a value 1 just when the values of its operands are different, otherwise the output is 0.

Boolean algebra finds its use in a number of industry disciplines such as logic circuits in electrical engineering and computer engineering, working with statements in propositional logic, or assessing the state of the system and its components in the reliability analysis (Crama, 2011).

2.1.3 Basic definitions and types

Consider the degree of the argument x, which will be $x_i^{\mathcal{L}_i}$ where $\mathcal{L}_i$ is a binary variable, then we can write:

$$x_i^{\mathcal{L}_i} = \begin{cases} x_i, \text{ if } \mathcal{L}_i = 1, \\ \overline{x_i}, \text{ if } \mathcal{L}_i = 0. \end{cases} \quad (37)$$

We call the variables x_i and their negations $\overline{x_i}$ (i = 1, 2, ..., n) as a letters, and i – as the number or index of a variable.

DEFINITION 1. An expression of the form:

$$x_1^{\mathcal{L}_1} x_2^{\mathcal{L}_2} \dots x_r^{\mathcal{L}_r} \quad (38)$$

is called an elementary conjunction (K) of rank r. Due to the fact that $x_i\overline{x_i} = 0$ and $x_i x_i \dots x_i = x_i$, all the letters in the elementary conjunction are different.

DEFINITION 2. An expression of the form:

$$K_1 \lor K_2 \lor \dots \lor K_j, \quad (39)$$

where K_j are elementary conjunctions of various ranks, is called disjunctive normal form (DNF). For example, the function:

$$f(x_1, \dots, x_4) = x_1 x_2 \lor x_1 x_2 \overline{x_3} \lor \overline{x_1} x_3 x_4 \quad (40)$$

is written in DNF, since all three terms are elementary conjunctions.

DEFINITION 3. If the function $f(x_1, \dots, x_n)$ is written in DNF, and the rank of each elementary conjunction is n, then such a DNF is called a perfect DNF (SDNF), and the conjunction - member of the SDNF.

DEFINITION 4. An expression of the form:

$$x_1^{\mathcal{L}_1} \lor x_2^{\mathcal{L}_2} \lor \dots \lor x_r^{\mathcal{L}_r} \quad (41)$$

is called an elementary disjunction (A) of rank r.

DEFINITION 5. Two elementary conjunctions are called orthogonal if their product is equal to zero. For example, the product of elementary conjunctions $x_1\overline{x_2}$ and $x_1\, x_2 x_3 x_4$ is equal to zero, since one of them contains $\overline{x_2}$ and the other x_2 and consequently, they are orthogonal.

DEFINITION 6. The DNF is called the orthogonal DNF (ODNF) if all its elementary conjunctions are pairwise orthogonal. In accordance with this definition, SDNF is ODNF, since all its terms are pairwise orthogonal. But the SDNF has the longest formula expression of all ODNF, since it contains the maximum number of letters.

DEFINITION 7. A non-repetitive DNF (BDNF) is DNF, in which all letters have different numbers. Letters x_i and $\overline{x_i}$ have the same number, so they cannot simultaneously to enter the BDNF.

DEFINITION 8. A probabilistic function (PF) is called probability of true of DNF or other form of Boolean function

$$Pr\,\{f(x_1, \dots, x_n) = 1\}. \quad (42)$$

When we take the theory of probability and mathematical logic, these events in logic are independent and random, and therefore we can make a transition between them (Ryabinin I. A., 1981). This request for undependability of variables is truth for orthogonal forms.

The transformation of DNF and other forms of Boolean function into the orthogonal form is an important problem. Below one of the possible algorithms for this transformation is considered.

The mathematical logic tool for BSS processing is well developed but does not allow to analyze MSS. Therefore we propose to use MVL for an MSS analysis.

2.2 MVL

Boolean algebra is defined on a set of two elements, $M = \{0, 1\}$. The operations of Boolean algebra must adhere to certain properties, called laws, or axioms used to prove more general laws about Boolean expressions to simplify expressions.

Multivalued algebra is a generalization of Boolean algebra, based upon a set of m elements $M = \{0, 1, 2, \dots, m\}$. The primary advantage of a multivalued system is the ability to encode more information per variable than a binary system is capable of doing (Yanushkevich, 2006).

Multiple-Valued logics (Multi-Valued Logics, Many-Valued Logics) are 'logical calculi' in which there are more than two truth values.

Those most popular in the literature is three-valued logic (e.g., Łukasiewicz's and Kleene's), which accept the values "true", "false", and "unknown“.

The MVL definition:

•The alphabet {0, 1, …, m-1};

•As a minimum two operation: "*" and "+"

•The constant "0": 0*X = 0 and 0+0=0, X ∈ {0, 1, …, m-1}

The operations of Boolean algebra have their analogs in multivalued algebra. The multivalued counterparts of binary AND, OR and NOT are the multivalued *conjunction, disjunction* and *cycling* operators introduced by Post in 1920.

MVL-function

An m-valued MVL-function of n arguments (variables) is a mapping:

$$f(x_1, x_1, \dots, x_n) = f(x): (0, 1, \dots, m-1)^n \rightarrow (0,1, \dots, m-1) \quad (43)$$

Example Function

$f: \{0, 1, 2\}^n$ → {0, 1, 2} is called a ternary logic function,

$f: \{0, 1, 2, 3\}^n$→ {0, 1, 2, 3} is called a quaternary logic function,

$f: \{0, 1, 2\}^n$→ {0, 1} is called multivalued input binary valued output logic functions.

2.2.1 Operations of multivalued logic

Below we list some of the implementation-oriented m-valued logic operations.

The MAX operation is defined as:

$$\text{MAX}(x_1, x_2) = x_1 \text{ if } x_1 \geq x_2, \quad x_2 \text{ otherwise.} \quad (44)$$

When m = 2, this operation turns into an OR operation. A MAX function of n variables is written:

$$\text{MAX}(x_1, x_2, \dots, x_n) = x_1 \lor x_2 \lor \dots \lor x_n. \quad (45)$$

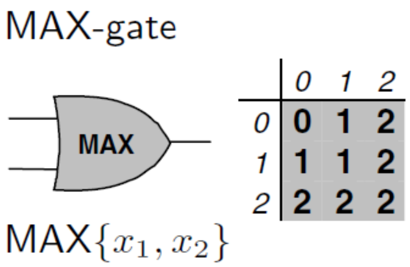


Figure 7 Operation MAX for ternary (m = 3) two-variable elementary functions.

The MIN operation of *x*1 and *x*2 is defined as:

$$\text{MIN}(x_1, x_2) = x_2 \text{ if } x_1 \geq x_2, \quad x_1 \text{ otherwise.} \tag{46}$$

and for *n* variables is written:

$$\text{MIN}(x_1, x_2, \ldots, x_n) = x_1 \wedge x_2 \wedge \ldots \wedge x_n. \tag{47}$$

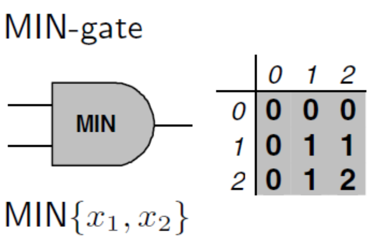


Figure 8 Operation MIN for ternary (m = 3) two-variable elementary functions.

The modulo *m* product operation is defined by:

$$\text{MOD-PROD}(x_1, x_2, \ldots, x_n) = x_1 x_2 \ \ldots, x_n \ mod\ (m). \tag{48}$$

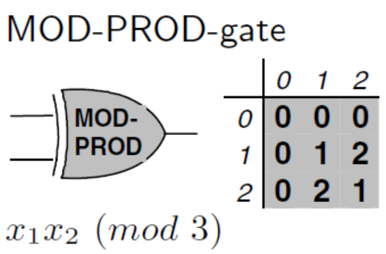


Figure 9 Operation MOD-PROD for ternary (m = 3) two-variable elementary functions.

The modulo *m* sum operation is defined below as

$$\text{MODSUM}(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n \; mod \; (m). \tag{49}$$

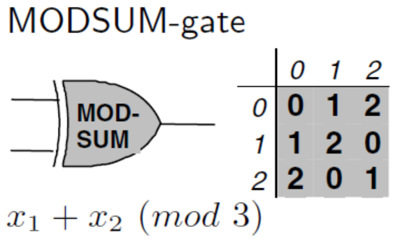


Figure 10 Operation MODSUM for ternary (m = 3) two-variable elementary functions.

The truncated sum operation of *n* variables is specified by

$$\text{TSUM}(x_1, x_2, \ldots, x_n) = \text{MIN}(x_1 \vee x_2 \vee \ldots \vee x_n, m - 1). \tag{50}$$

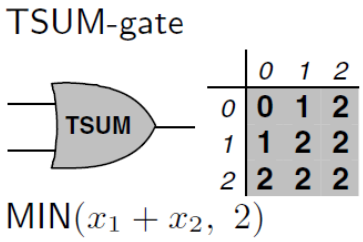


Figure 11 Operation TSUM for ternary (m = 3) two-variable elementary functions.

The truncated product operation is defined by

$$\text{TPROD}(x_1, x_2, \ldots, x_n) = \text{MIN}(x_1 \wedge x_2 \wedge \ldots \wedge x_n, (m - 1)). \tag{51}$$

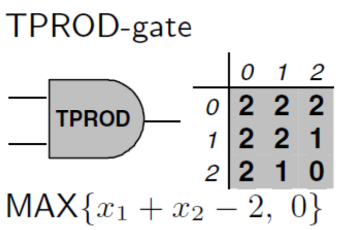


Figure 12 Operation TPROD for ternary (m = 3) two-variable elementary functions.

The complement operation is specified by:

$$\bar{x} = (m - 1) - x, \tag{52}$$

where $x \in M$ is a unary operation. For example, in ternary logic, $\bar{x} = 2 - x$. Notice that the property $\bar{\bar{x}} = x$ can be used in multivalued logic. This is because $(m - 1) - \bar{x} = (m - 1) - ((m - 1) - x) = x$.

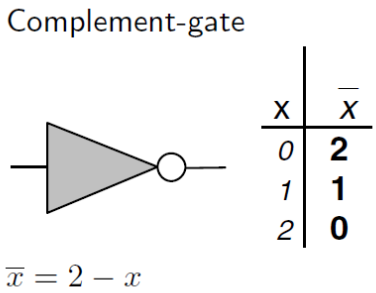


Figure 13 Operation Complement for ternary (m = 3) two-variable elementary functions.

Typical problems in MVL are:

- Optimization,
- Minimization,
- Classification,
- System function representation and analysis,
- Event Driven Analysis.

In this work we focus mainly on problems of optimization, minimization and system function representation and analysis. In computer science optimization is the process of modifying objects to make some it aspect more efficiently or use fewer resources.

The MVL function may be optimized so that it is capable of operating with less memory storage or other resources.

The typical approach for MSS structure function analysis is a generalization of methods for BSS structure function analysis that are based on Boolean logic. However, this approach does not allow using all details of MSS. Another approach is based on the application of MVL mathematical methods for MSS structure function analysis. According to this approach, the structure function is explained as MVL function. *Multi-Valued Decision Diagrams* (MDDs) are often used for efficient manipulation of MSS structure function. MDDs have been proposed for the analysis of large dimension MVL function in (Miller, 2002). MDDs have been used in reliability computation due to their compact and easy representation of structure function and some the first investigations of MDD application in reliability analysis of MSS are considered in (Amari 2010, Zaitseva 2008, Mo 2014). Need to note that the construction of MDD of minimal complexity (minimal numbers of nodes) is an actual problem (Kuo 2007, Mo 2014).

The conception of orthogonalization in MVL has been considered in some investigations (Perkowski 1992, Perkowski 1991).

In this work, the conception of orthogonalization is considered for the truth table of an MSS structure function (MVL function) or variables vectors of the structure function. Two types of orthogonalization are considered: complete orthogonal variables vectors and partly orthogonal variables vectors.

Similar to Boolean logic, the two variables vector of MSS structure function (MVL function) are orthogonal if they are disjoint to one another. Consider two variables vectors $a_1 \ldots a_j \ldots a_n$ and $b_1 \ldots b_j \ldots b_n$ where a_i, $b_i \in \{0, \ldots, m_i\text{-}1\}$. These vectors are orthogonal if at least one pair of variables satisfies the condition $a_i \neq b_i$.

The set of complete orthogonal variables vectors consists of m_i variables vectors for which there is one variable that has different values in each vector, and other variables values are equal. The variable is named orthogonal variable if it has different values for these vectors.

For example, let us consider the MSS structure function with $m_i = m_s = M = 3$ (all inputs and outputs variables can be in 3 states - the structure function (1) is homogenous if $M = m_i = m_j$ for $i \neq j$) and $n = 4$. The variables vectors 0102, 0112, 0122 are complete orthogonal, because x_1, x_2 and x_4 have equal values, and x_3 has different values from 0 to 2.

The set of partly orthogonal variables vectors consists of s variables vectors ($2 \leq s \leq m_i\text{-}1$) for which there is one variable that has different values in each vector, and other variables values are equal. The variable is named orthogonal variable if it has different values for these vectors.

Two variables vectors 0112 are 0122 are partly orthogonal for MSS structure function of $n = 4$ components and for $m_i = m_s = M = 3$. The variables x_1, x_2 and x_4 have equal values, and x_3 has different values from 1 and 2.

The conceptions of complete orthogonalization and partly orthogonalization are used in algorithm for the minimization of MSS structure function truth table.

2.3 Orthogonalization

Orthogonalization is very important for the processing of a structure function in the reliability analysis. It expects independent events. By using it, it can easily go to the probability form.

Use of the orthogonal form in the theory of Reliability Engineering

DNF and transform Boolean formula to it.

The complexity of solving a number of problems formulated in terms of DNF systems decreases if DNF is reduced to a form in which all elementary conjunctions are mutually orthogonal. Such forms include tasks from the domain of decomposition of Boolean functions, the synthesis of logical networks, the reliability of technical systems (Zakrevskij, 2008).

Applications in the theory of Reliability Engineering

The applications will be shown in Reliability Engineering on determining the operability of a technical system. This system is represented by a ternary matrix described in section 1.2.1 Table representation in section Ternary matrix.

Suppose that all the basis events d_i are mutually independent and the probability of their realization for some instant of time is known: $\Pr(d_1)$, $\Pr(d_2)$, etc. Let us set the problem of determining the probability of failure of the system for the same instant of time.

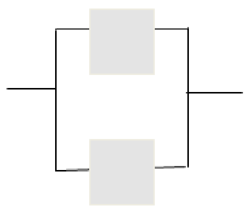
As usual, two theorems from the probability theory (4) and (5) are used in Reliability Engineering. The first (4) of these theorems makes it easy to calculate the probability of realizing any critical set of basis events. However, further calculations of the probability of a complex event *R* are difficult, since the events represented by different critical sets can be, in general, dependent and compatible.

The solution can be found by first transforming the original DNF representing the event *R* into an equivalent DNF composed of mutually orthogonal conjunctive terms corresponding to incompatible events. Then the probability of the event *R* can be expressed as the sum of the probabilities of events represented by the members of the obtained orthogonal DNF. Thus, the calculation of the probability of a complex event R reduces to the orthogonalization of the DNF that defines this event (Zakrevskij, 2005).

2.3.1 Boolean function

The conception of orthogonalization is well known and often used in the Boolean algebra for variable vectors (Schneeweiss 2009, Hudson 1983). This conception is defined for the Boolean function and for product terms of the function. According to (Hudson 1983), two product terms of a Boolean function are orthogonal if their product is zero or they are disjointed to one another. The conception of orthogonal product term can be generalized for the variables vectors: two variables vectors are orthogonal if their product is zero. The mathematical description of the Boolean function is orthogonal if all product terms are orthogonal. According to (Barlow 1978, Barlow 1975) the orthogonal form of the structure function (1) is transformed into probabilistic form by the substitution of structure function variables by the probabilities of appropriate components states. For example, for the Boolean function orthogonal form this substitution is implemented as the replacements of an inverse variable by the probability of appropriate component failure and non-inverse variable by the probability of appropriate component functioning.

Consider the example of the parallel system that has two performance levels, and its components can be in two possible states. It is BSS (Figure 14).



x_1	x_2	$\phi(\boldsymbol{x})$
0	0	0
0	1	1
1	0	1
1	1	1

Figure 14 Parallel system with its truth table

The structure function of this system can be represented as:

$$\phi(\boldsymbol{x}) = \phi(\boldsymbol{x}) = x_1 \vee x_2 = \mathrm{OR}(x_1, x_2) \tag{53}$$

$$\phi(\boldsymbol{x}) = \overline{x_1}\, x_2 \vee x_1\, \overline{x_2} \vee x_1\, x_2 \tag{54}$$

The mathematical representation (53) is not orthogonal. This representation can be transformed to probabilistic form by special rules only. At the same time the representation (54) is orthogonal, and the probabilistic form (system availability (2)) is developed by substitute of probabilities of two components state instead of the structure function variables:

$$A = q_1p_2 + p_1q_2 + p_1p_2 = (1-p_1)\cdot p_2 + p_1\cdot(1-p_2) + p_1p_2 = p_2 - p_1\cdot p_2 + p_1 - p_1\cdot p_2 + p_1p_2 = \mathbf{p_1+p_2-p_1p_2} \quad (55)$$

The system could be described using the Boolean function. The function may be incompletely and completely defined. This function describes the logical linking of elements in the system, but does not allow us to analyze probability conditions. It is important for us to move from logical through orthogonal form to probability form and get to the reliability of the system.

2.3.2 MVL function

The orthogonalization of DNF considered in this work is based on the disjunctive expansion of an elementary conjunction into many other conjunctions that became orthogonal to conjunctions from defined family or absorbed by one of them. For reducing further calculations, the absorbed conjunction is removed from the result and the remaining conjunctions are minimized. The orthogonalization algorithm described in Zakrevskij and Pottosin (Zakrevskij, 2005) is adapted for orthogonalization in this work.

Orthogonal DNF - DNF that contains orthogonal elementary conjunctions and that means multiplying these conjunctions gives 0. The algorithm works on the principle of operation expanding k_i over k_j, where k_i and k_j are certain non-orthogonal elementary conjunctions.

The algorithm is shown on example, a pair of vectors representing the considered conjunctions k_i and k_j.

In the first step, we select all variables which are found in k_i but not in k_j. The number of these variables is t. In our case t = 3 and the variables are x_2, x_5, x_9 (Figure 15).

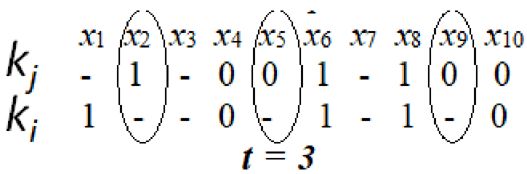


Figure 15 First step of expansion

In the second step, we expanded disjunctively k_i by the first of these variables. One of the products of this expansion will be orthogonal to the conjunction k_j and the other will be expanded in the second variable of the selected set, etc.

Variable x_2 in conjunction k_i is expanded by 1 and 0 to conjunctions in the Figure 16. One of the conjunctions which is orthogonal on k_j (it's a second conjunction of expansion in Figure with value of variable x_2=0 because variable x_2 in k_j have value of variable x_2=1) is inserted into result and the second will be expanded by a second variable x_5 .

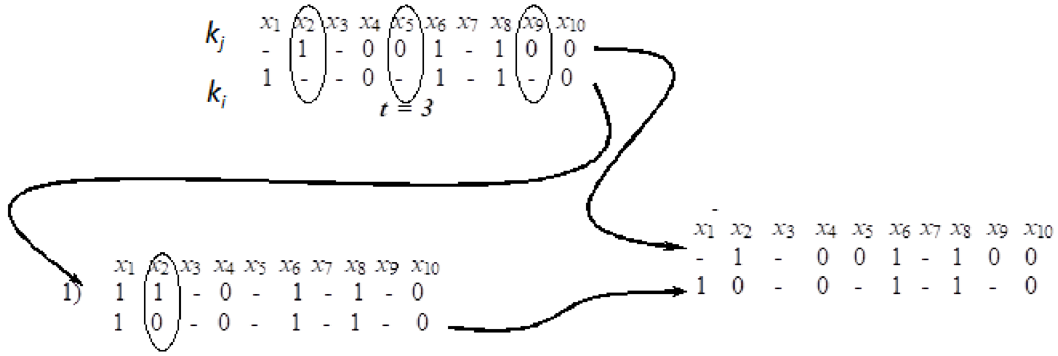


Figure 16 Second step of expansion

The same procedure is performed with variable x_5, which is expanded in conjunction k_i by 1 and 0 to conjunctions in the Figure 17. One of the conjunctions which is orthogonal on k_j (it's the first conjunction of expansion in Figure 16 with value of variable x_5=1 because variable x_5 in k_j have value of variable x_5=0) is inserted into result and the second will be expanded by the third variable x_9.

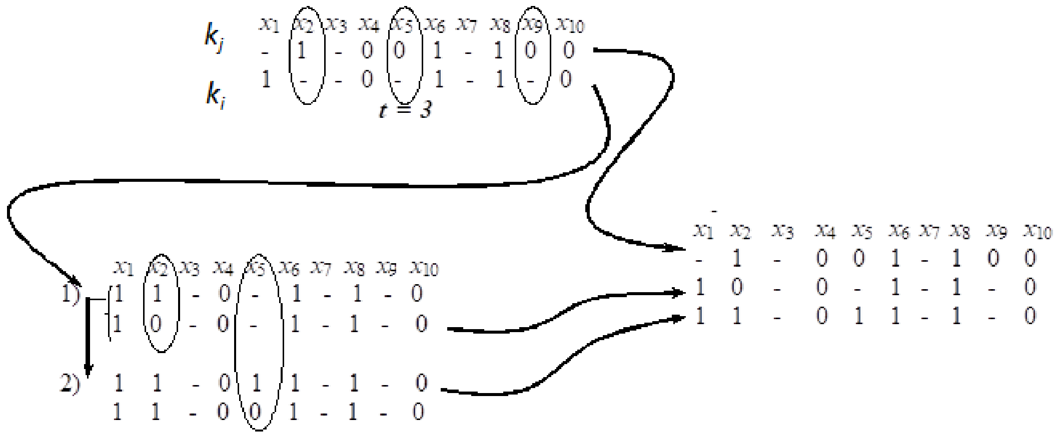


Figure 17 Process of expansion

In the expansion with respect to the last variable, one of the products will be orthogonal to the conjunction k_j, and the other will be absorbed by it.

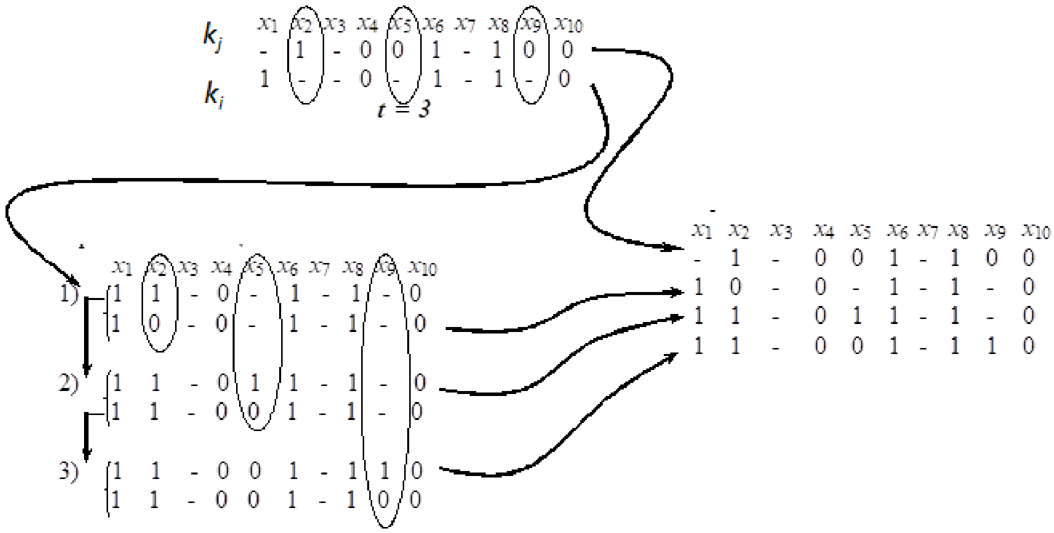


Figure 18 Final look of expansion of ternary vector

The result is that the conjunction k_i comes out to be replaced by t conjunctions which are orthogonal to k_j (in our example conjunction k_i is replaced by 3 conjunction orthogonal to conjunction k_j in Figure 18).

The operation of disjunctive expansion is easily realized when the conjunctions are represented by ternary vectors, the components which correspond with variables and take the value 1 if the variable enters the conjunction without inversion, the value 0 if variable is with inversion and the value "-" if not included (Figure 19).

$$T = \begin{array}{c} \begin{matrix} a & b & c & d & e & f & g \end{matrix} \\ \begin{bmatrix} 0 & 1 & - & 0 & 1 & 1 & - \\ 1 & 0 & 1 & 0 & - & 1 & 1 \\ 1 & 1 & - & 0 & 1 & 1 & 0 \\ 1 & 1 & - & 0 & - & - & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ - & 1 & 1 & 1 & 0 & 0 & - \\ - & - & - & 1 & 1 & 0 & - \\ - & - & - & 1 & - & 1 & - \end{bmatrix} \end{array}$$

$$\bar{a}b\bar{d}ef \vee a\bar{b}c\bar{d}fg \vee .....$$

Figure 19 Non-orthogonal form: matrix of ternary vectors

If the ranks of the conjunctions k_i and k_j are not equal, then it is more profitable to expand the conjunction of a larger rank by a conjunction of smaller rank - the number of products of the expansion will be smaller in this case. This principle is used in the orthogonalization of DNF in the algorithm below. Consider the procedure for orthogonalization of the DNF given by the ternary matrix T.

1. Before the beginning of the orthogonalization, the rows of the matrix *T* are partially ordered in order of non-decreasing the ranks, measured by the number of ones in the row. It's described with matrix *T* the initial DNF which represents a complex event *R*.
2. We create a matrix T^+ using the ternary vectors that are included in it. This is done by sequentially selecting the rows from the matrix by using the above-defined disjunctive expansion of the corresponding elementary conjunctions.

The first row of the matrix *T* is transferred to the generated matrix T^+, which will specify the required orthogonal DNF. In this process, the selected row is compared successfully with the rows that are already included in T^+ and is expanded as to the first of them which is non-orthogonal to it. Then next row is selected from matrix *T* and compared it with each of the rows that are included in matrix T^+. If the compared rows are non-orthogonal, then the row is expanded as to the first of them. The products of the expansion not absorbed by the rows of T^+ are added at the bottom of matrix T^+ and we repeat the same process until the selected row becomes orthogonal to all rows of matrix T^+. Then it can be selected next row from matrix *T* if any still exists.

After each iteration, connected with the selection and expansion of the next row from the matrix *T*, all the rows of the matrix T^+ are mutually orthogonal. After selecting all the rows from *T*, the matrix T^+ will represent the required orthogonal DNF.

For example, the following two matrices show the execution of the algorithm where Table 7 is the initial matrix and T^+ Table 8 is the result of its orthogonalization. It is orthogonalized the DNF of the Boolean function of the system, represented by the ternary matrix *T*, which is the initial matrix. Matrix T^+ is the product of orthogonalization described with algorithm above. The algorithm is suitable for any DNF and is suitable for the monotone boolean function.

Table 7 Initial matrix T

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
	1	0	-	-	-	0	-	1
$T =$	-	0	1	1	-	-	0	2
	1	-	1	1	1	-	-	3
	0	0	1	-	1	0	1	4

The products of the expansion of the rows of the matrix *T* are labeled in T^+ by the numbers of these rows in *T*. The components of the expansion corresponding to the variables, which the original variable was expanded, are indicated in bold.

Table 8 Result of orthogonalization of initial matrix T

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
	1	0	-	-	-	0	-	1
	0	0	1	1	-	-	0	2
$T^+ =$	**1**	0	1	1	-	**1**	0	2
	1	**1**	1	1	1	-	-	3
	1	**0**	1	1	1	**1**	**1**	3
	0	0	1	-	1	0	1	4

x_1	x_2	x_3	x_4	x_5	x_6	x_7		
1	0	-	-	-	0	-	1	expansion
-	0	1	1	-	-	0	2	expansion
0	0	1	1	-	-	0	2	
1	0	1	1	-	-	0	2	← absorbed
1	0	1	1	-	**1**	0	2	
1	0	1	1	-	**0**	0	2	← absorbed

Figure 20 Process of orthogonalization for second conjunction

The first conjunction is added to the result matrix. Then we can see in Figure 20 expansion of the second conjunction where only rows highlighted in blue are added to the result matrix and the remaining rows are absorbed.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7		
	1	0	-	-	-	0	-	1	
	0	0	1	1	-	-	0	2	
$T^+ =$	**1**	0	1	1	-	**1**	0	2	
	1	**1**	1	1	1	-	-	3	
	1	**0**	1	1	1	-	-	3	← absorbed
	1	**0**	1	1	1	**1**	1	3	
	1	**0**	1	1	1	**0**	-	3	← absorbed

Figure 21 Process of orthogonalization for third conjunction

The third conjunction is expanded in Figure 21. As we can notice the second expansion in component x_6 is orthogonal in the first row in result matrix however it is not in the next 2 rows and it is the reason why the component x_7 is set on state 1 (marked in red) to be orthogonal also on conjunctions in rows 2 and 3.

The calculation of the probability of the complex event R is no longer difficult. For example, suppose that all basis events x_i are independent and the probability of occurrence of each of them is 1/3. Then the probability of $\overline{x_i}$:

$$\Pr(\overline{x_i}) = 1 - \Pr(x_1) = 1 - 1/3 = 2/3, \tag{56}$$

the probability of the product of the events $x_1, \overline{x_2}, \overline{x_6}$ represented by the first row of the matrix T^+ is:

$$\Pr(x_1)\ \Pr(\overline{x_2})\ \Pr(\overline{x_6}) = (1/3)\ (2/3)\ (2/3) = 2^2 / 3^3, \tag{57}$$

and, finally, the probability of a complex event R:

$$\Pr(R) = 2^2 / 3^3 + 2^3 / 3^5 + 2^2 / 3^6 + 2^0 / 3^5 + 2^1 / 3^7 + 2^3 / 3^6 \approx 0{,}203. \tag{58}$$

For example system considered in section 2 (Structure Function of Laparoscopic Surgery Procedure Success) in the Table 6 for BSS this orthogonal form was obtained:

$$x_1 x_2 x_3 \overline{x_4} \ \vee \ \overline{x_1}\, x_2 x_3 x_4 \ \vee \ x_1 x_2 x_3 x_4 \tag{59}$$

The orthogonal representation (46) of the structure function of Laparoscopic Surgery Procedure Success can be transformed into the probabilistic form according to the definition 8 (section 3):

$$A = p_{10} \cdot p_{21} \cdot p_{31} \cdot p_{41} + p_{11} \cdot p_{21} \cdot p_{31} \cdot p_{40} + p_{11} \cdot p_{21} \cdot p_{31} \cdot p_{41} = p_{21} \cdot p_{31} \cdot (p_{11} + p_{41} - p_{11} \cdot p_{41}). \tag{60}$$

The probabilistic form (60) is the system availability in terms of reliability analysis.

2.4 Minimalization

2.4.1 Boolean function

Simplification of Boolean expressions

A product term (conjunction of literals) is said to be an **implicant** of a complete function if the product term implies the function. Therefore, each of the product terms in a sum of product form describing a complete Boolean function is an implicant of the function. In other words we can say that minterms of the functions are its implicants.

An implicant of a Boolean function is said to be a **prime implicant** if the implicant does not subsume any other implicant with fewer literals of the same function. For example, consider a Boolean function:

$$f(A, B, C) = A\bar{B}C + \bar{B}C. \quad (61)$$

In this function $A\bar{B}C$ is an implicant of the function since it is the minterm (the inputs of function that evaluate to 1 are called 'minterms') that describes the function. On the other hand, consider the implicant $\bar{B}C$. There is no other term in the function which is subsumed by $\bar{B}C$ and hence it is a prime implicant. It is important to note that, if the Boolean expression is the sum of prime implicants then it corresponds to one of the minimal sum of product (disjunctive normal) formula.

2.4.2 MVL

The truth table of MSS structure function is an orthogonal form of the function representation. It means that all variables vectors of this table are orthogonal. However, the truth table has dimensional $m_1 \times m_2 \times \ldots \times m_n$ that complicates this table analysis and evaluation. The minimization of the truth table is the constructing truth table with less dimension and without loss of information about the system behavior. One of the ways for the minimization of the truth table is replacing the variables vectors with variables that do not have an influence to the function value. As a rule, it is orthogonal variables vector with respect to one variable for equal value of the function. The formed variables vector has less number of variables than initial variables vectors. Therefore the formed truth table includes fewer variables vectors, and these vectors include fewer variables. The complete and partial orthogonalizations of variables can be taken into account. Similar process is used in the algorithm of Quine–McCluskey for the minimization of the Boolean function (McCluskey, 1956).

Quine-McCluskey method used in many – valued logic

The main goal of the algorithm is to find a minimum subset of the implicants similar such that the standard disjunction of the implicants would be functionally equivalent to the function.

Input of algorithm is a n-valued logical function *f* with *m* input components, defined by a truth table of values.

The output of an algorithm is a simplified normal form functionally equivalent to the function *f*.

The algorithm is based on 2 basic steps:

The first step is to find a **prime implicants**.

1. The rows of the true table of values represent the set of n^m implicants. For example, the row 1 from Table 1 represent implicant [1: 0 0 0] where the first value defined the output value of a function and the next three values defined the input values of the components x, y, z.
2. At first we select rows (implicants) with a non-zero output value of the function from table 1 and we create a lookup table. This table is of $n - 1$ dimensional hypercube shape and each side of the hypercube contains $m + 1$ cells. The implicants are organized in the table according to the number of input logical values they contain and every axis of the hypercube corresponds to one logical value.

   For example, if we have a five-valued logical function with 8 input components the implicants with three 1(state of 3 input components has value 1), no 2, five 3, and no 4 are sorted to the cell with coordinates [3, 0, 5, 0].
3. In the lookup table we find such implicants that differ in value only in one place, only in the state of one component.
4. When we find such a group of implicants differing only in one place, we will join all the implicants in this group and replace them with one new implicant. At the position where these implicates differ, we mark as a dash -. The output value of the new implicant is the lowest value of the output values of every implicant of the group of implicants. The important fact is that when searching for implicants belonging to a given group, they can only be found in adjacent cells of the truth table of values. As a result, we don't have to seek through all the cells in the table, which simplifies the computational complexity.
5. Then with * we mark all implicants from the group of implicants differing only in one place which output value of this implicants is the smallest value of all implicants occurring in this group. These implicants marked with * will not occur as a result because we have just covered them with other implicant.

   For example if we have logical function with four possible states 0, 1, 2, 3 and with 3 components then group of implicants { [1: 0 0 3], [3: 0 1 3], [1: 0 2 3], [2: 0 3 3] } can be reduced to only one implicant [1: 0 - 3]. The smallest output value of the group is 1 so the output value of new implicant is also 1. And finally we marked with * implicants [1: 0 0 3] [1: 0 2 3] because their output value is the smallest of all implicants of this group. Another example is group of implicants { [1: 0 0 3], [1: 0 1 3], [1: 0 2 3], [1: 0 3 3] } that can be reduced to only one implicant [1: 0 - 3]. The smallest output value of the group is 1 so the output value of new implicant is also 1. And finally, we marked with * all implicants.

6. We repeat the searching for such a group of implicants until we find all possible groups and replace them with new implicants and marked with * those that will cover with other implicant in the result.
7. Those implicants that were not marked with * are marked alphabetically (sequentially A, B, C,..). These implicants are prime implicants and may be part of the resulting simplified normal form, but it must be verified, so they are processed in the second part of the algorithm in the table of covering.
8. All the reduced implicants are sorted to a new lookup table and then we continue by step 3.
9. The number of repeating of the steps of the algorithm will not be higher than the number of the input components. The reason is that the implicants in the new lookup table contain one more "–" than the implicants in the previous lookup table. Thus the edges of the new lookup table are shorter by one.

The second step is to build a **table of covering**.

Table of covering is a two - dimensional table where each column represents one prime implicant given by step 1 of the algorithm and each row represents one row from the truth table of values - non-zero implicants. Each non-zero value in the table of covering represents the output value of the prime implicant for that column, with the fact that the prime implicant covers the implicants (rows) for that row where the value is assigned. The aim of the second step is to find a subset of the prime implicants that cover all non-zero implicants from the truth table of values.

1. We find the unique largest value for each row and mark it if such a value exists. The prime implicant (column) where the value is marked must be found in the resulting normal form because the row (implicant) can be covered only by this corresponding prime implicant. Then we can remove from the table of covering column that represents this prime implicant and also all of the rows (implicants)

corresponding to the prime implicants (where there exist some non-zero values). We repeat this step until no more reduction is possible.

2. Row dominance: If we find two rows (implicants) r1 and r2 where one is a subset of the other, r1 ⊆ r2, we remove r2. The reason for this removal is that if row one is covered by prime implicant, row two is also covered and row two does not provide any new information for the table of covering.
3. Column dominance: If we find two columns c1 and c2 where one is a subset of the other, c1 ⊆ c2, we remove c1. The reason for this removal is that if column one which represents prime implicant covers implicants (rows) associated with this column then this column covers also implicants associated with column two. Therefore c1 is redundant and we can remove c1.
4. We repeat Step 2 and Step 3 until no more reduction is possible.
5. In step one we have selected those prime implicants that must be found in the resulting normal form and in step 3 we removed those that are redundant. After these steps, we should have prime implicants for which we are not able to decide which are redundant. Thus we find the redundant prime implicants in this remaining set of prime implicants by choosing any suitable method.
6. The prime implicants given by Step 1 and the prime implicants which are not redundant (step 3 and step 5) are then aggregated by the standard disjunction which gives us the resulting simplified normal form.

Example

First we describe algorithm on an example for completely defined three-valued logical function f (x,y,z). The function is defined by 3 input components x, y, z, that can be in 3 states together with a function output value also defined by 3 states. The function is defined by a truth table of values (Table 9).

Table 9 Example of structure function defined by a truth table

	x	y	z	f (x,y,z)
1.	0	0	0	1
2.	0	0	1	1
3.	0	0	2	2
4.	0	1	0	0
5.	0	1	1	0
6.	0	1	2	1
7.	0	2	0	0
8.	0	2	1	0
9.	0	2	2	1
10.	1	0	0	1
11.	1	0	1	1
12.	1	0	2	2
13.	1	1	0	0
14.	1	1	1	0
15.	1	1	2	1
16.	1	2	0	0
17.	1	2	1	0
18.	1	2	2	1
19.	2	0	0	1
20.	2	0	1	2
21.	2	0	2	2
22.	2	1	0	1
23.	2	1	1	1
24.	2	1	2	1
25.	2	2	0	1
26.	2	2	1	1
27.	2	2	2	0

Solution

1. First, we remove the rows (implicants) whose output value of logical function is 0 from the original table (Table 9). Then we create a new lookup table (Table 10)

from the remaining rows. Lookup table consists of sorted implicants according to numbers of input states they could have.

Table 10 Lookup table

2 / 1	0	1	2
0	1: 0 0 0 *	1: 1 0 0 * 1: 0 0 1 *	1: 1 0 1 *
1	1: 2 0 0 * 2: 0 0 2 *	1: 0 1 2 * 2: 1 0 2 * 2: 2 0 1 A 1: 2 1 0 *	1: 1 1 2 * 1: 2 1 1 *
2	1: 0 2 2 * 2: 2 0 2 * 1: 2 2 0 *	1: 1 2 2 * 1: 2 1 2 * 1: 2 2 1 *	

2. In a lookup table we look for three implicants that differ in the component state in only one place. We replace these 3 implicants with one and leave a character of dash (-) at the point where they differ. For example in our lookup table we can replace the implicants [1: 0 0 0], [1: 2 0 0] and [1: 1 0 0] with an implicant [1: - 0 0]. We mark all the implicants in a lookup table by *, because all implicants have the smallest output value of logical function. Another case that may occur is when we want to replace for example implicants [1: 0 0 0], [2: 0 0 2] and [1: 0 0 1] by implicant [1: 0 0 -]. As you may have noticed for the new implicant the smallest output value of the original implicants is selected as the new output value and only implicants with this value in the lookup table are marked by *. In Table 11 all implicants that continue processing in the new lookup table are defined.

Table 11 Implicants from the previous lookup table

No.	x	y	z	f (x,y,z)
1.	-	0	1	1
2.	1	0	-	1
3.	2	1	-	1
4.	0	0	-	1
5.	-	0	0	1
6.	2	-	1	1
7.	-	1	2	1
8.	1	-	2	1
9.	2	0	-	1
10.	2	-	0	1
11.	-	0	2	2
12.	0	-	2	1

3. Any implicants that remain unmarked after the previous step must be part of the result. In our lookup table it is the implicant [2: 2 0 1]. This implicant is marked by letter A.
4. Now we create new lookup table (Table 12) from implicants which we obtained in step 2 from Table 11. With all of them we perform the same procedure until we cannot find any other implicants which can be reduced (Table 13).

Table 12 New lookup table

2 / 1	0	1
0	1: 0 0 - * 1: - 0 0 *	1: 1 0 - * 1: - 0 1 *
1	1: 2 0 - * 1: 2 - 0 B 2: - 0 2 C 1: 0 - 2 D	1: 2 1 - E 1: 2 - 1 F 1: - 1 2 G 1: 1 - 2 H

Table 13 New lookup table in the last step of the procedure of searching of prime implicants

2 / 1	0
0	1: - 0 - I

5. Now we have implicants marked by letters (Table 14), which are prime implicants and we can now create a table of covering (Table 15).

Table 14 Table of implicants

A	2: 2 0 1
B	1: 2 - 0
C	2: - 0 2
D	1: 0 - 2
E	1: 2 1 -
F	1: 2 - 1
G	1: - 1 2
H	1: 1 - 2
I	1: - 0 -

6. The columns of the table of covering represent the found prime implicants from Table 14 and non-zero implicants of the Table 9 represent rows. The task of the covering table is to find, if possible, a minimum subset of prime implicants that cover all implicants in the rows of the table.
7. In covering table, we mark the value of each row that is only the largest and underline it. The corresponding value in a given column indicates a prime implicant that will be among the resulting subset of prime implicants. Then we remove implicants covered by this prime implicant from the table of covering. We also remove the column with this prime implicant. Finally we use a row and column dominance rules for possible reduction of the table of covering.
8. After reducing the rows and columns of the table, we get the Table 14. In this table, both remaining prime implicants cover all remaining rows (implicants) so that we can select one of them (H) and include it in the resulting implicant Table 16.

Table 15 Table of covering

x, y, z	A	B	C	D	E	F	G	H	I
0,0,0									1
0,0,1									1
0,0,2			2	1					1
0,1,2				1			1		
0,2,2				1					
1,0,0									1
1,0,1									1
1,0,2			2					1	1
1,1,2							1	1	
1,2,2								1	
2,0,0		1							1
2,0,1	2					1			1
2,0,2			2						1
2,1,0		1			1				
2,1,1					1	1			
2,1,2					1		1		
2,2,0		1							
2,2,1						1			

9. Finally we have a subset of prime implicants that cover all possible implicants(rows in Table 15). After applying the maximum operation to these prime implicants, we get a simplified normal form of function f (x,y,z) (62).

Table 16 Table of resulting implicants

A	2: 2 0 1
B	1: 2 - 0
C	2: - 0 2
D	1: 0 - 2
E	1: 2 1 -
F	1: 2 - 1
H	1: 1 - 2
I	1: - 0 -

$$f(x,y,z) = (2 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=0] \wedge [\mathrm{z}=1])$$
$$\vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{z}=0])$$
$$\vee (2 \wedge [\mathrm{y}=0] \wedge [\mathrm{z}=2])$$
$$\vee (1 \wedge [\mathrm{x}=0] \wedge [\mathrm{z}=2])$$
$$\vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=1])$$
$$\vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{z}=1])$$
$$\vee (1 \wedge [\mathrm{x}=1] \wedge [\mathrm{z}=2])$$
$$\vee (1 \wedge [\mathrm{y}=0]) \quad (62)$$

2.4.3 Incompletely specified function

Example 2:

Now we will describe algorithm on an example for incompletely defined three-valued logical function f (x,y,z). The function is defined by 3 input components x, y, z, that can be in 3 states together with a function output value also defined by 3 states. The function is defined by a truth table of values (Table 17). The incompletely defined states of components are marked by question mark.

Solution

1. As well as for a completely defined logic function, first we remove the rows whose output value is 0. Rows whose component state is incompletely defined continue to be processed in the algorithm.
2. For the incompletely defined input values of components we defined states because we need to include these implicants with undefined states to the construction of lookup table. For example for implicant [1: 0 1 ?] we defined that component *c* will be in state 2 so this implicant will look like [1: 0 1 2]. The result of defined states is in Table 18, where all newly defined values are listed in red cells.

Table 17 Example of structure function incompletely defined by a truth table

No.	x	y	z	f (x,y,z)
1.	0	0	0	0
2.	0	0	1	0
3.	0	0	2	0
4.	0	1	**?**	0
5.	0	2	0	0
6.	1	0	0	0
7.	2	0	0	0
8.	**?**	1	0	0
9.	0	1	1	1
10.	0	1	**?**	1
11.	0	2	1	1
12.	**?**	0	1	1
13.	1	1	0	1
14.	1	1	1	1
15.	2	**?**	2	1
16.	2	2	0	1
17.	2	0	1	1
18.	1	0	2	2
19.	1	1	2	2
20.	1	2	0	2
21.	**?**	2	1	2
22.	0	2	2	2
23.	1	2	2	2
24.	**?**	2	2	2
25.	2	1	1	2
26.	2	1	2	2
27.	2	**?**	1	2

Table 18 Redefined states in red cells

No.	a	b	c	f (x,y,z)
1.	0	1	2	1
2.	1	0	1	1
3.	2	0	2	1
4.	1	2	1	2
5.	2	2	2	2
6.	2	2	1	2

3. Now we create a lookup table (Table 19) with implicants that remained after reduction of rows in step 1 and after redefined missing states in step 2.

Table 19 Lookup table

1 / 2	0	1	2	3
0			1: 2 0 2 * 2: 0 2 2 * 1: 2 2 0 *	2: 2 2 2 *
1		1: 0 1 2 * 2: 1 0 2 * 1: 0 2 1 * 2: 1 2 0 * 1: 2 0 1 *	2: 1 2 2 * 2: 2 1 2 A 2: 2 2 1 B	
2	1: 0 1 1 * 1: 1 1 0 * 1: 1 0 1 *	2: 1 1 2 * 2: 1 2 1 * 2: 2 1 1 C		
3	1: 1 1 1 *			

4. In the same way as a completely defined function we look for three implicants that differ in the component state in only one place in a lookup table. The reason why we defined in the previous step all the undefined states with some of the possible states 0, 1, 2 is that now we can join these implicants to the process of replacing of implicants that differ only in one value. We replace these 3 implicants with one and leave a character of dash (-) at the point where they differ. For example in our lookup table we can replace the implicants [2: 0 2 2], [2: 1 2 2] and [2: 2 2 2] with implicant [2: - 2 2].We mark all the implicants in lookup table by *, because all

implicants have the smallest output value of logical function. In Table 20 all implicants that continue processing in the new lookup table is defined.

Table 20 Implicants from the previous lookup table

No.	x	y	z	f (x,y,z)
1.	-	2	2	2
2.	2	-	2	1
3.	2	2	-	1
4.	-	1	2	1
5.	-	2	1	1
6.	1	-	2	2
7.	1	2	-	2
8.	2	-	1	1
9.	-	1	1	1
10.	1	1	-	1
11.	1	-	1	1

5. Any implicants that remain unmarked after the previous step must be part of the result. In our lookup table it is an implicants [2: 2 1 2], [2,2,1], [2,1,1]. These implicants are marked by letters A, B, C.

Table 21 New lookup table

2 / 1	0	1	2
0			2: - 2 2 D 1: 2 2 - E 1: 2 - 2 F
1		1: - 1 2 G 1: - 2 1 I 2: 1 - 2 J 2: 1 2 - K 1: 2 - 1 L	
2	1: - 1 1 M 1: 1 1 - N 1: 1 - 1 O		

6. Now we create new lookup table from implicants which we obtained in step 4 from the implicants of table 20. With all of them we perform the same procedure until we cannot find any other implicants which can be reduced.
7. Now we have implicants marked by letters (Table 22), which are prime implicants and we can now create a table of covering (Table 23).

Table 22 Table of resulting implicants

A	2: 2 1 2
B	2: 2 2 1
C	2: 2 1 1
D	2: - 2 2
E	1: 2 - 2
F	1: 2 2 -
G	1: - 1 2
H	1: - 2 1
I	2: 1 - 2
J	2: 1 2 -
K	1: 2 - 1
L	1: - 1 1
M	1: 1 1 -
N	1: 1 - 1

8. The task of the covering table is to find, if possible, a minimum subset of prime implicants that cover all implicants in the rows of the table.
9. In covering table, we mark the value of each row that is only the largest and underline it. After this step, we can see that in the table of covering we cannot reduce any prime implicants.
10. Finally we have a subset of prime implicants that cover all possible implicants(rows in Table 23). After applying the maximum operation to these prime implicants, we get a simplified normal form of function f (x,y,z) (63).

Table 23 Table of covering

x, y, z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
0,1,1												1		
0,1,2							1							
0,2,1								1						
1,0,1														1
1,1,0				1									1	
1,1,1												1	1	1
2,0,2					1									
2,2,0						1								
2,0,1											1			
1,0,2									2					
1,1,2							1		2				1	
1,2,0										2				
1,2,1								1		2				1
0,2,2				2										
1,2,2				2					2	2				
2,2,2				2	1	1								
2,1,1			2								1	1		
2,1,2	2				1		1							
2,2,1		2				1		1			1			

$$
\begin{aligned}
f(x,y,z) = {} & (2 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=1] \wedge [\mathrm{z}=2]) \\
& \vee (2 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=2] \wedge [\mathrm{z}=1]) \\
& \vee (2 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=1] \wedge [\mathrm{z}=1]) \\
& \vee (2 \wedge [\mathrm{y}=2] \wedge [\mathrm{z}=2]) \\
& \vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{z}=2]) \\
& \vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=2]) \\
& \vee (1 \wedge [\mathrm{y}=1] \wedge [\mathrm{z}=2]) \\
& \vee (1 \wedge [\mathrm{y}=2] \wedge [\mathrm{z}=1]) \\
& \vee (2 \wedge [\mathrm{x}=1] \wedge [\mathrm{z}=2]) \vee (2 \wedge [\mathrm{x}=1] \wedge [\mathrm{y}=2]) \\
& \vee (1 \wedge [\mathrm{x}=2] \wedge [\mathrm{z}=1]) \vee (1 \wedge [\mathrm{y}=1] \wedge [\mathrm{z}=1]) \\
& \vee (1 \wedge [\mathrm{x}=1] \wedge [\mathrm{y}=1]) \vee (2 \wedge [\mathrm{x}=2] \wedge [\mathrm{y}=2] \wedge [\mathrm{z}=1])
\end{aligned}
\tag{63}
$$

3 Experiments

3.1 Minimization of MSS structure function

The proposed algorithm for the structure function truth table (minimization) is started from the division of the MSS structure function truth table into M truth sub-tables for each value of the structure function. Next, M steps are analysis of each of truth sub-table that is implemented as:

- *The choice of completely orthogonal variables vectors*. The variables vectors (rows) differed by one variable are chosen in the sub-table. The chosen variables vectors are completely orthogonal to the variable with different values if their number is equal m_i (the number of possible variable states). The variable with different values does not have any influence on an analyzed function value if other variable's values are equal. Therefore this variable can be removed. The different values of the variable are replaced by dash and a new vector is included in the result table. This process is repeated for remaining variables vectors in the truth sub-table. If the completely orthogonal variables vectors are not found, the analysis of partly orthogonal vector is started.
- *The choice of partly orthogonal variables vectors.* The variables vectors (rows) differed by one variable are chosen in the sub-table. The chosen variables vectors are partly orthogonal to the variable with different values if their number is less than m_i. The variable with different values has restricted influence for analyzed function value if other variable's values are equal. Therefore this variable can be removed for chosen values. The different values of the variable are replaced by a set of chosen values and a new vector is included in the result table (chose values of the variable are written separated by a comma). This process is repeated for remaining variables vectors in the truth sub-table. If the partially orthogonal variables vectors are not found the remaining variables vectors (rows) are copied to the resulting table.
- The variables vectors (rows) in the resulting truth sub-table must be mutually orthogonal. If the variables vectors in the resulting sub-table are not orthogonal, one of them is replaced by an initial set of variables vectors.

The algorithm of MSS structure function is illustrated by the example. Consider the MSS of three components (n = 3). The system and its components have three states (m_i = m_s =M = 3). The structure function of this system is shown in Figure 22.

Structure function of system

x_1	x_2	x_3	$\phi(\mathbf{x})$
0	0	0	0
0	0	1	0
0	0	2	0
0	1	0	0
0	2	0	0
1	0	0	0
2	0	0	0
2	1	0	0
0	1	1	1
0	1	2	1
0	2	1	1
1	0	1	1
1	1	0	1
1	1	1	1
2	0	2	1
2	2	0	1
2	0	1	1
1	0	2	2
1	1	2	2
1	2	0	2
1	2	1	2
0	2	2	2
1	2	2	2
2	2	2	2
2	1	1	2
2	1	2	2
2	2	1	2

3 truth sub-tables

x_1	x_2	x_3	$\phi(\mathbf{x})$
0	0	0	0
0	0	1	0
0	0	2	0
0	1	0	0
0	2	0	0
1	0	0	0
2	0	0	0
2	1	0	0

x_1	x_2	x_3	$\phi(\mathbf{x})$
0	1	1	1
0	1	2	1
0	2	1	1
1	0	1	1
1	1	0	1
1	1	1	1
2	0	2	1
2	2	0	1
2	0	1	1

x_1	x_2	x_3	$\phi(\mathbf{x})$
1	0	2	2
1	1	2	2
1	2	0	2
1	2	1	2
0	2	2	2
1	2	2	2
2	2	2	2
2	1	1	2
2	1	2	2
2	2	1	2

Figure 22 Structure function of a system with 3 truth sub-tables

According to the first step of the proposed algorithm, the initial truth table is divided into three sub-tables (M = 3) (Figure 22).

Consider the minimization of the truth sub-table for the structure function value 0 (Fig.4). The first procedure is the choice of completely orthogonal variables vectors. The first three rows ((0, 0, 0), (0, 0, 1) and (0, 0, 2)) differ only by the variable x_3 since they contain all possible states in the variable x_3. Therefore these variables vectors are completely orthogonal and the variable x_3 can be removed. It is replaced by the dash and vector (0, 0, -) is included in the result table (Figure 23). There are no other completely orthogonal variables vectors in this table. Therefore the analysis is continued for partially orthogonal variables vectors.

The second procedure is the choice of partially orthogonal variables vectors. The two variables vectors (0, 1, 0) and (0, 2, 0) differ by the variable x_2. These variables vectors are partially orthogonal and the variable x_2 can be merged for values 1 and 2. The variables vector (0, (1,2), 0) are included into the result sub-table. If only two rows match, as is the case for the next two rows in the table, we write both states in the variable that do not match in the resulting table. The variables vectors (2, 0, 0) and (2, 1, 0) are transformed to the vector (2, (0,1), 0) in a similar way.

One variable vector is in the initial truth sub-table after the forming of two new variables vectors according to the procedure of the choice of partly orthogonal variables vectors. It is the variable vector (1, 0, 0) that is copied in the result sub-table. All variables vectors in result sub-table are orthogonal. Therefore another transformation is not needed.

Input table

x_1	x_2	x_3	$\phi(x)$
0	0	0	0
0	0	1	0
0	0	2	0
0	1	0	0
0	2	0	0
1	0	0	0
2	0	0	0
2	1	0	0

Result table

x_1	x_2	x_3	$\phi(x)$
0	0	-	0
0	1,2	0	0
1	0	0	0
2	0,1	0	0

Figure 23 Minimization of the truth sub-table for the structure function value 0

Because the result table is orthogonal, it can be used for the probability representation of the considered MSS structure function. In particular, according to (2) the probability of this MSS failure (performance level 0) is:

$$A_0 = p_{1,0}p_{2,0} + p_{1,0}(p_{1,1}+p_{2,2})p_{3,0} + p_{1,1}p_{2,0}p_{3,0} + p_{1,2}(p_{2,0} + p_{2,1})p_{3,0} \quad (64)$$

The truth sub-table for the structure function value 1 (Figure 24) and the truth sub-table for the structure function value 2 (Figure 25) are minimized similarly. Based on the result sub-tables probabilities of the system performance levels 1 and 2 according to (2) can be presented and calculated as:

$$A_1 = p_{1,0}p_{2,1}(p_{3,1} + p_{3,2}) + p_{1,0}p_{2,2}p_{3,1} + p_{1,1}p_{2,0}p_{3,1} + p_{1,1}p_{2,1}(p_{3,0+}p_{3,1}) + p_{1,2}p_{2,0}(p_{3,1} + p_{3,2}) + p_{1,2}p_{2,2}p_{3,0} \tag{65}$$

and

$$A_2 = p_{1,1}(p_{2,0} + p_{2,1})p_{3,2} + p_{1,1}p_{2,2} + p_{1,0}p_{2,2}p_{3,2} + p_{1,2}p_{2,2}(p_{3,1}p_{3,2}) + p_{1,2}p_{2,1}(p_{3,1} + p_{3,2}) \tag{66}$$

Input table

x_1	x_2	x_3	$\phi(x)$
0	1	1	1
0	1	2	1
0	2	1	1
1	0	1	1
1	1	0	1
1	1	1	1
2	0	2	1
2	2	0	1
2	0	1	1

Result table

x_1	x_2	x_3	$\phi(x)$
0	1	1,2	1
0	2	1	1
1	0	1	1
1	1	0,1	1
2	0	1,2	1
2	2	0	1

Figure 24 Minimization of the truth sub-table for the structure function value 1

Input table

x_1	x_2	x_3	$\phi(x)$
1	0	2	2
1	1	2	2
1	2	0	2
1	2	1	2
0	2	2	2
1	2	2	2
2	2	2	2
2	1	1	2
2	1	2	2
2	2	1	2

Result table

x_1	x_2	x_3	$\phi(x)$
1	0,1	2	2
1	2	-	2
0	2	2	2
2	2	1,2	2
2	1	1,2	2

Figure 25 Minimization of the truth sub-table for the structure function value 2

For the comparison, the probabilities of the system performance levels 2 based on the initial truth sub-table is represented as:

$$A_2 = p_{1,1}p_{2,0}p_{3,2} + p_{1,1}p_{2,1}p_{3,2} + p_{1,1}p_{2,2}p_{3,0} + p_{1,1}p_{2,2}p_{3,1} + p_{1,0}p_{2,2}p_{3,2} + p_{1,1}p_{2,2}p_{3,2} + p_{1,2}p_{2,2}p_{3,2} + p_{1,2}p_{2,1}p_{3,1} + p_{1,2}p_{2,1}p_{3,2} + p_{1,2}p_{2,2}p_{3,1} \quad (67)$$

The comparison of (66) and (67) shows that the calculation of the probability of the system performance level based on minimized truth sub-table has less computational complexity.Moreover the verification of the sum of all probabilities of the system performance levels is equal 1:

$$A_0 + A_1 + A_2 = p_{1,0}p_{2,0} + p_{1,0}\big(p_{1,1}p_{2,2}\big)p_{3,0} + p_{1,1}p_{2,0}p_{3,0} + p_{1,2}(p_{2,0} + p_{2,1})p_{3,0} + p_{1,0}p_{2,1}\big(p_{3,1} + p_{3,2}\big) + p_{1,0}p_{2,2}p_{3,1} + p_{1,1}p_{2,0}p_{3,1} + p_{1,1}p_{2,1}(p_{3,0}p_{3,1}) + p_{1,2}p_{2,0}(p_{3,1} + p_{3,2}) + p_{1,2}p_{2,2}p_{3,0} + p_{1,1}\big(p_{2,0} + p_{2,1}\big)p_{3,2} + p_{1,1}p_{2,2} + p_{1,0}p_{2,2}p_{3,2} + p_{1,2}p_{2,2}(p_{3,1}p_{3,2}) + p_{1,2}p_{2,1}(p_{3,1} + p_{3,2}) = p_{1,0}p_{2,1} + p_{1,0}p_{2,2} + p_{1,1}p_{2,0} + p_{1,2}p_{2,0} + p_{1,1}p_{2,1} + p_{1,2}p_{2,1} + p_{1,2}p_{2,2} + p_{1,1}p_{2,2} + p_{1,0}p_{2,0} = p_{1,0} + p_{1,1} + p_{1,2} = 1 \quad (68)$$

The minimalized truth-table of MSS structure function can be used to construct other forms for MSS analysis. One of them is MDD. MDD for the considered example constructed according to typical rules presented in (Zaitseva, 2012, Miller, 2002) is shown in Figure 26.

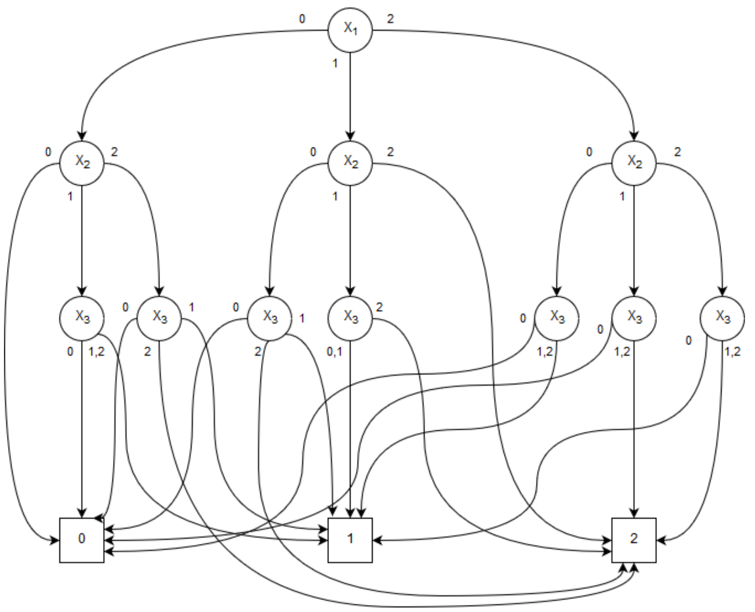


Figure 26 Multi-valued Decision Diagram

3.2 Results for minimalization of generate functions

The evaluation of the efficiency of the logical function minimization was implemented based on the set of the generated monotone functions. The monotone logical functions agree with the definition of coherent structure functions, which allow representing most of the real system. The generated set includes the structure function of MSS with 3 performance levels of system and 3 states of every of the system components. The number of the function is 50. These functions had been minimized by the developed algorithm for MSS structure function minimization (this algorithm is considered in section 2.4.2).

We got the results which are shown in Figure 27. From the graph we can see that maximum decrease of implicants for generate functions was about 33% and the minimum was about 3%. The most common value of decline was about 18% and this value was measured for 13 functions.

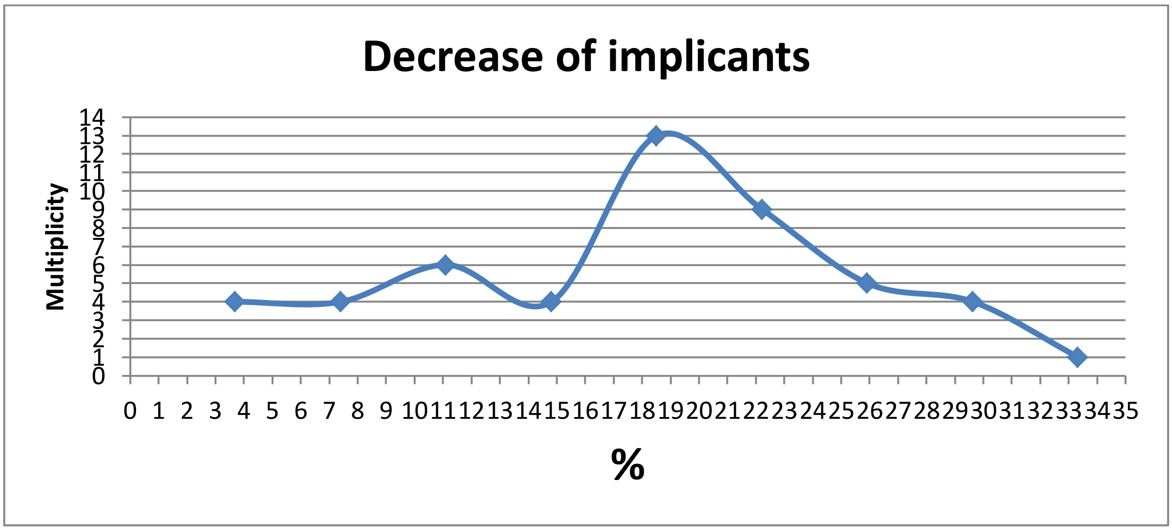


Figure 27 Decrease of implicants for generate functions

The processing procedure is shown in Figure 28. We have defined system described by structure function and first we use algorithms for minimalization to reduce or simplify this structure function, which can be very complex in real systems. Thanks to the minimization process, we can significantly reduce this function which leads to simpler computational complexity. If there is for instance fifth power of a number 3 there are 243 possible defined implicants for this structure function. If this function can be simplified by minimalization to 150 implicants, this form can be already orthogonal but it is not guaranteed thus as a next process orthogonalization is used. After Orthogonalization, on the other hand, the number of implicants may increase, and therefore it is more important for us transformation efficiency into an orthogonal form which may or may not be simplified.

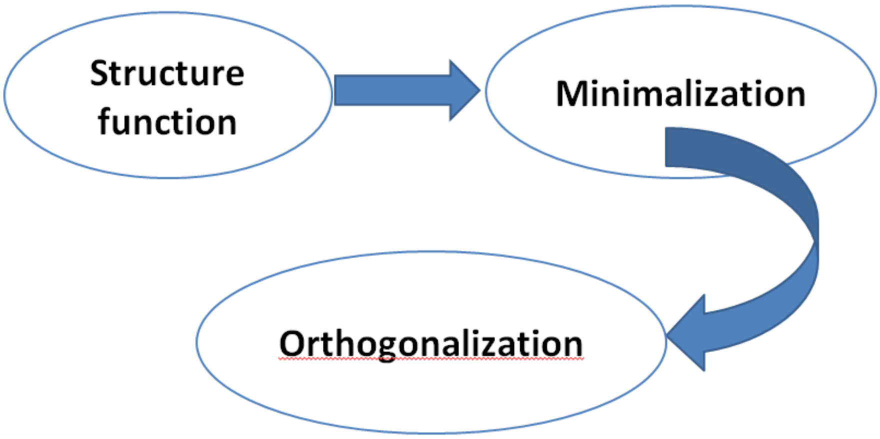


Figure 28 Processing of the system defined by structure function

3.3 Orthogonalization of a real system

3.3.1 K-out-of-n configuration

The components topology in the analyzed system is in most cases some common reliability-wise component topologies or their combination (Tortorella, 2015). One of the most common types of system topology in reliability analysis is k out of n configuration and our drone fleet is analyzed with this type of topology. K out of n configuration is functional if at least k components out of all n components are functional (Tortorella, 2015). The reliability block diagram of this configuration is shown in Figure 29.

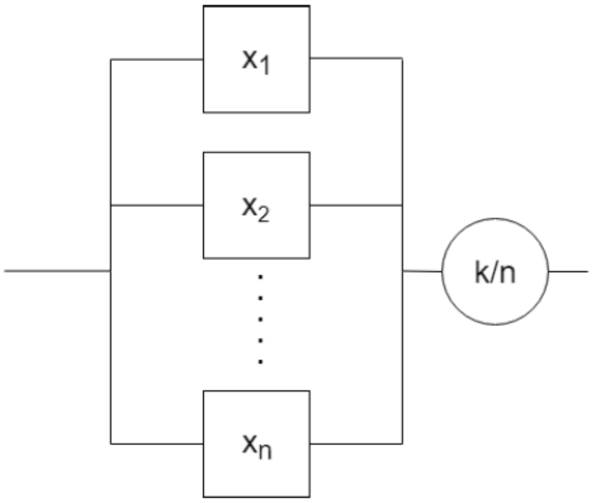


Figure 29 RBD for k-out-of-n configuration

This configuration can be interpreted by structure function following its definition as the following notation (Tortorella 2015, Rausand 2004):

$$\phi(x_1, x_2, \dots, x_n) = \bigvee_{S_i \in S_{k/n}} \bigwedge_{j \in S_i} x_j, \tag{68}$$

where $S_{k/n}$ represents sets of all k combinations of all n components indexes and S_i is a ith selected combination from $S_{k/n}$.

3.3.2 Complex monitoring system with drone fleet

The whole algorithm and configuration is applied on complex monitoring system with a drone fleet (Figure 30). In this drone fleet system 3 types of components exists. The first type of components is the control unit (CU) and its task is to control and manage all drones in the fleet. The control unit is the most important element of the drone fleet since the other drones are dependent on its working and cannot do their tasks without it. The second type of component is the main drone (MD) which represents all *kd* main drones. MD executes scheduled tasks that are set by CU. The last type of components is *nd*-*kd* redundant drone (RD). RD represents back up if some of the main drones fails. RD assumes responsibility for this failed main drone and performs its tasks. This complex monitoring system with a drone fleet can be in two states. The first state is when the system can be considered as a functional and it means that CU and at least *kd* drones are functional. In this state it can perform all scheduled tasks in the drone fleet. The second state is when the system does not meet previous conditions and the system can be considered as faulty.

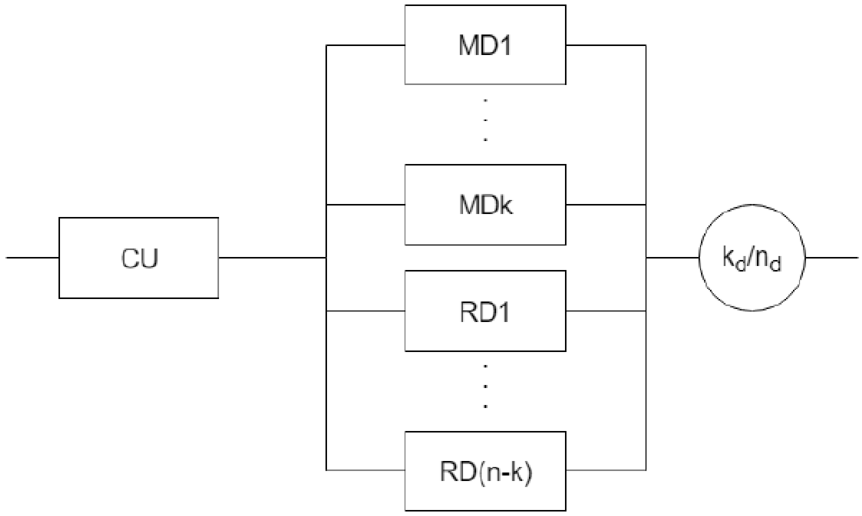


Figure 30 RBD for drone fleet

As it is known, system reliability is a probability that system performs its functions during defined time assuming that it worked at the beginning, we will suppose that CU and all drones are irreparable and all components of drone flight are working at the beginning.

3.3.3 Orthogonalization of system for drone fleet

In (Rusnak, 2019), author considers drone fleet where CU can control and manage up to 5 drones at a time (*nd*=5). There is also condition to have redundant drones in the fleet and at least 2 drones must working and it means that *kd* can have value 2, 3 or 4.

This drone fleet system can have 2 modules, the first module is CU and the second is kd-out-of-5 system representing all drones in the fleet for $kd \in$ (2,3,4). The structure function for such a system has this form:

$$\phi(x_1, x_2, x_3, x_4, x_5, x_6) = x_1 \wedge \bigvee_{S_i \in S_{\frac{k_d}{5}}} \bigwedge_{j \in S_i} x_j \quad (69)$$

Structure function can be defined for $kd \in$ (2, 3, and 4). In our case, if kd=3, then the structure function (13) has the following form:

$$\phi(x_1, x_2, x_3, x_4, x_5, x_6) = x_1(x_2\ x_3\ x_4 \vee x_2\ x_3\ x_5 \vee x_2\ x_3\ x_6 \vee x_2\ x_4\ x_5 \vee x_2\ x_4\ x_6 \vee x_2\ x_5\ x_6 \vee x_3\ x_4\ x_5 \vee x_3\ x_4\ x_6 \vee x_3\ x_5\ x_6 \vee x_4\ x_5\ x_6) \quad (70)$$

This formula is not orthogonal and to efficiently compute reliability from structure function, the orthogonal form of structure function for this configuration is used and it has the following form (Rausand 2004, Tortorella 2015):

$$\phi(x_1, x_2, \ldots, x_n) = \bigvee_{x_i \in X_{k/n}} \bigwedge_{j \in S_{x_i}} x_j \bigwedge_{j \in N - S_{x_i}} \overline{x_j} \tag{71}$$

where Xk represents a set of all state vectors in which at least k variables are not negated, N represent set of all components indexes and S_{x_i} represents a set of components indexes that are not negated for i-th state vector.

$$\begin{aligned}\phi(x_1, x_2, x_3, x_4, x_5, x_6) = (&x_1\ x_2\ x_3\ x_4\ \overline{x_5}\ \overline{x_6} \vee x_1\, x_2\ x_3\ \overline{x_4}\ x_5\ \overline{x_6} \vee \\ &x_1\, x_2\ x_3\ \overline{x_4}\ \overline{x_5}\ x_6 \vee x_1\, x_2\, \overline{x_3}\ x_4\ x_5\ \overline{x_6} \vee x_1\, x_2\ \overline{x_3}\ x_4\ \overline{x_5}\, x_6 \vee \\ &x_1\, x_2\, \overline{x_3}\ \overline{x_4}\ x_5\ x_6 \vee x_1\, \overline{x_2}\ x_3\ x_4\ x_5\ \overline{x_6} \vee x_1\, \overline{x_2}\ x_3\ x_4\ \overline{x_5}\ x_6 \vee \\ &x_1\, \overline{x_2}\ x_3\, \overline{x_4}\ x_5\ x_6 \vee x_1\, \overline{x_2}\ \overline{x_3}\, x_4\ x_5\ x_6 \vee x_1\, \overline{x_2}\ x_3\ x_4\ x_5\ x_6 \vee \\ &x_1\, x_2\ \overline{x_3}\ x_4\ x_5\ x_6 \vee x_1\, x_2\ x_3\ \overline{x_4}\ x_5\ x_6 \vee x_1\ x_2\ x_3\ x_4\ \overline{x_5}\ x_6 \vee \\ &x_1\ x_2\ x_3\ x_4\ x_5\ \overline{x_6} \vee x_1\ x_2\ x_3\ x_4\ x_5\ x_6\,)\end{aligned} \tag{72}$$

According to the algorithm from section 2.3.2 we use orthogonalization to form (70) and we obtain Table 24.

Table 24 Final orthogonal conjunctions

No	x_1	x_2	x_3	x_4	x_5	x_6
1	1	1	1	1	-	-
2	1	1	1	0	1	-
3	1	1	1	0	0	1
4	1	1	0	1	1	-
5	1	1	0	1	0	1
6	1	1	0	0	1	1
7	1	0	1	1	1	-
8	1	0	1	1	0	1
9	1	0	1	0	1	1
10	1	0	0	1	1	1

In table 24 component x_1 represents the first module of drone fleet and it is CU and therefore it has always a state of 1 (functioning). According to the algorithm we sequentially added rows to the table and we control to add a new row orthogonal to the other rows already added in the resulting table.

For example when we add the row 2 in the table then the component x_4 in conjunction (row) 2 is expanded by 1 and 0 to conjunctions in Table 25. However, in result Table 24 only row with component x_4 in state 0 is included because the second row where component x_4 is in state 1 is already absorbed by the first row in the resulting table.

Table 25 Example of expansion

No	x_1	x_2	x_3	x_4	x_5	x_6
1	1	1	1	1	-	-
2	1	1	1	**0**	1	-
2	1	1	1	**1**	1	-

Another example is adding row 5 where we must change component x_3 to state 0 to be orthogonal to rows 1, 2 and 3 and we also change the component x_5 from the state - to state 0 to be orthogonal to row 4 (marked in red). Therefore in Table 26 we use only one row (marked in blue), the remaining row is absorbed.

Table 26 Example of expansion

No	x_1	x_2	x_3	x_4	x_5	x_6
1	1	1	1	1	-	-
2	1	1	1	0	1	-
3	1	1	1	0	0	1
4	1	1	0	1	1	-
5	1	1	**0**	1	**0**	1
5	1	1	**1**	1	-	1

For comparison algorithm of orthogonalization from section 2.3.2 give us a lower number of conjunctions, in which the formula 70 can be represented in orthogonal form than the formula (71) described by (Rausand 2004, Tortorella 2015). While the first mentioned described the system with 10 conjunctions, second described it with up to 16.

Conclusion

The reliability analysis is valuable research for various systems. The analysis of any system starts by defining the number of states and then proceeding to the development of a mathematical description of the system. Therefore these 2 steps are the most important for us and only after their processing we can do further analysis of complex systems. There are two approaches for system representation that are based on MSS and BSS. BSS allows representing the initial system as a mathematical model with two possible states that are complete failure and perfect working. MSS permits to consider more than only two states in the behavior of system reliability or availability.

When we select a structure function as a mathematical representation of the system, this function does not always have the form as needed and therefore some processing and changes of structure function are necessary. It causes the development of algorithms and methods for the construction of orthogonal and minimal representation of MSS structure function. Such a form of representation allows moving from a logical to a probabilistic description and then gets to the reliability of the system

We introduced a processing model of a real system that was a drone fleet. We defined this system by structure function and then we showed that thanks to the minimization process, we can significantly reduce this function which leads to simpler computational complexity. Form of function after minimalization can be already orthogonal but it is not guaranteed so as the next process we have to to use the process of orthogonalization. After orthogonalization, on the other hand, the number of implicants may increase, and therefore it is more important for us the transformation efficiency into an orthogonal form which may or may not be simplified.

According to the principal goal, next task were decided:

- the method proposed in (Zakrevskij & Pottosin 2005) was modified to the application for the BSS structure function orthogonalization;
- the conception of orthogonal form in the Multiple-Valued Logic was considered and conception of the orthogonal MSS structure function had been proposed;
- the proposed conception of the orthogonal MSS structure was used for the development of algorithms of MSS structure function minimization and orthogonalization;

- the developed algorithms were approbated and evaluated on selected systems (structure functions of BSS and MSS).

Resume

1 Predmet výskumu

Súčasný stav a úroveň technológií spôsobujú nové trendy a podmienky vo vývoji teórie spoľahlivosti. Existuje široká škála úloh, ktoré sa zvyčajne netýkajú teórie spoľahlivosti, o ktorých sa nedá rozhodnúť použitím tradičných metód. Takéto úlohy napríklad hodnotia riziko teroristického útoku (Levitin 2009), spoľahlivosť podnikovej analýzy (Solojentsev 2009), odhadujú riziká a dôsledky technologických havárií (Zio 2009) a mnoho ďalších. Moderné technológie zároveň umožňujú takmer bezporuchovú prevádzku technickej časti zložitých systémov. Táto situácia spôsobuje zmenu tradičných prístupov a podnecuje vývoj nových metód v teórii spoľahlivosti (Birolini 2014, Ushakov 2006, Zio 2009). E. Zio v prehľade o teórii spoľahlivosti (Zio 2009) definoval teóriu spoľahlivosti ako „presne ohraničený multidisciplinárny vedecký odbor, ktorého cieľom je poskytnúť súbor formálnych metód na výskum neurčitých hraníc medzi fungovaním a zlyhaním systému, riešením nasledujúcich otázok:

- Prečo zlyhávajú systémy, napr. použitím konceptov spoľahlivostnej fyziky na zistenie príčin a mechanizmov zlyhania a identifikáciu následkov;
- Ako vyvinúť spoľahlivé systémy, napr. návrhom založeným na spoľahlivosti;
- Ako merať a testovať spoľahlivosť pri návrhu, prevádzke a riadení;
- Ako udržiavať spoľahlivosť systémov pomocou údržby, diagnostiky a prognózy porúch. “

Podľa analýzy (Zio 2009) známe problémy teórie spoľahlivosti ako:

- matematický popis systému;
- kvantitatívna analýza systému;
- reprezentovanie, znázornenie a kvantifikácia neistoty v správaní systému,

by sa mali brať do úvahy v rámci nových výziev v teórii spoľahlivosti. Na základe týchto skutočností je možné konštatovať, že skúmanie matematickej reprezentácie systému je relevantným problémom teórie spoľahlivosti. Hlavné kroky pre vývoj matematického znázornenia systému v teórii spoľahlivosti sú (Bris 2014, Aven 2017):

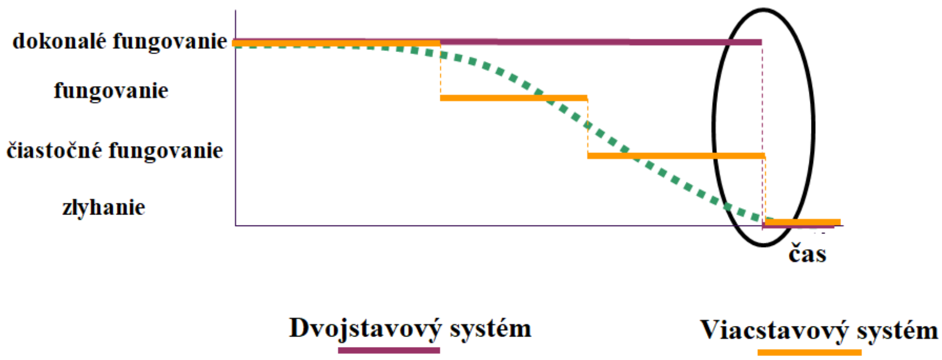
1. definícia počtu úrovní výkonnosti systému;
2. matematické znázornenie modelu systému;

3. kvantifikácia modelu systému (výpočet indexov);
4. meranie správania sa systému.

Prvý a druhý krok v analýze spoľahlivosti má koreláciu s počiatočnými údajmi. Cieľom týchto krokov je zostrojiť matematický model na hodnotenie spoľahlivosti. Preto sa v tejto práci venujeme v prvom rade týmto dvom krokom.

Prvý krok je definícia prístupu pre všeobecnú reprezentáciu systému. Existujú dva hlavné prístupy (Obrázok 1) na reprezentáciu systému pri analýze spoľahlivosti konkrétne viacstavový systém (MSS) (Barlow, 1978) a binárny systém (BSS) (Barlow, 1975). BSS umožňuje reprezentovať skúmaný systém ako matematický model s dvoma možnými stavmi, ktoré sú úplné zlyhanie a perfektné fungovanie. MSS, na rozdiel od BSS, umožňuje definovať aj viac ako dva stavy správania sa systému.

Podľa (Lisnianski, 2003) koncepcie ako práceschopnosť, spoľahlivosť a stavy systému môžu byť vyjadrené ako „úroveň výkonnosti“ MSS. Použitie MSS umožňuje podrobnejšie analyzovať spoľahlivosť systému avšak táto analýza je komplikovanejšia (Natvig 2010, Lisnianski 2003).



Obrázok 1 Dvojstavový a viacstavový systém

MSS nie je často používaný v analýze spoľahlivosti, pretože má dve hlavné obmedzenia. Prvým z nich je výpočtová zložitosť (Lisnianski, 2003). Uvedenie do analýzy ďalších úrovní výkonnosti systému a stavov komponentov spôsobuje značné zväčšenie dimenzie tejto matematickej reprezentácie. Druhým nedostatkom je málo účinných metód a algoritmov kvalitatívnej a kvantitatívnej analýzy pre MSS (Aven 2014, Birolini 2014, Zio 2009). Preto je výskum a vývoj v analýze spoľahlivosti MSS aktuálnym problémom v teórii spoľahlivosti.

Algoritmy pre hodnotenie MSS závisia od matematických metód použitých pri analýze systému. V (Lisnianski, 2003) autori uviedli štyri hlavné skupiny matematických metód analýzy MSS, ktoré reprezentujú správanie MSS formou štruktúrnej funkcie, Markovho modelu, univerzálne vytvárajúcej funkcii a matematického modelu založeného na simulácii Monte Carlo. Každý z týchto typov MSS má určité výhody. Dôležitými výhodami štruktúrnej funkcie sú jednoduchosť konštrukcie, možnosť aplikácie pre systém s akoukoľvek štrukturálnou zložitosťou a jednoduché metódy výpočtu indexov spoľahlivosti založené na metódach algebry logiky.

Typickým prístupom pri analýze štruktúrnych funkcií MSS je zovšeobecnenie metód pre analýzu štruktúrnych funkcií BSS, ktoré sú spravidla založené na booleovskej logike (Barlow 1975, Barlow 1978, Birolini 2014). Tento prístup má však obmedzenia, ktoré ho robia neefektívnym pre MSS. Ďalší prístup je založený na aplikovaní matematických metód viachodnotovej logiky (MVL) pri analýze štruktúrnej funkcie MSS (Zaitseva 2017, Zaitseva 2012, Kvassay 2017). Podľa tohto prístupu je štruktúrna funkcia vyjadrená ako funkcia MVL (Zaitseva 2017). Prístupy založené na matematických metódach MVL je možné využiť pre spracovanie a analýzu štruktúrnych funkcií MSS čo je ukázané pre výpočet práceschopnosti systému v (Zaitseva 2017, Zaitseva 2015), analýzu kritických stavov systému v (Kvassay 2017, Kvassay 2014) a analýzu dôležitosti v (Zaitseva 2012) , Zaitseva 2015).

Presná matematická reprezentácia je vytvorená v druhom kroku analýzy spoľahlivosti, ktorá vyplýva z matematických metód, ktoré sa použijú na vyhodnotenie skúmaného objektu / systému. Metódy analýzy spoľahlivosti MSS a BSS reprezentovaných štruktúrnou funkciou sú známe a bežne sa používajú v inžinierskej praxi a rôznych aplikáciách (Barlow 1978, Murchland 1975). Dôležité výhody štruktúrnej funkcie sú (Kolowrocki 2014, Lisnianski 2018, Natvig 2010):

- definíciu univalentnej korelácie úrovne výkonnosti systému a stavov komponentov;
- zobrazenie systému akejkoľvek štrukturálnej zložitosti;
- zložitosť reprezentácie systému nezávisí od jeho štruktúry.

Jedným z hlavných problémov pre ďalší vývoj a používanie MSS je nedostatočný matematický základ pre jeho analýzu.

Štruktúrna funkcia je jednou zo základných reprezentácií MSS. Dimenzia štruktúrnej funkcie sa však značne zvyšuje s narastajúcim počtom komponentov systému (Zaitseva, 2003). Vývoj metód štruktúrnej funkcie by mal byť založený na ortogonalizácii a minimalizácii. Aby bolo možné využívať štruktúrnu funkciu bez problémov v teórii pravdepodobnosti, musí byť logická forma reprezentácie štruktúrnej funkcie ortogonálna a minimálna (Solojentsev, 2009).

Typ matematickej reprezentácie závisí od detailnosti vyhodnotenia systému a od matematického prístupu, ktorý sa používa na výpočet indexov pri analýze spoľahlivosti.

Kvantifikácia systému v treťom kroku predpokladá výpočet indexov pri analýze spoľahlivosti ako sú napríklad funkcia spoľahlivosti, miera zlyhania, priemerný čas do zlyhania, priemerný čas na opravu, priemerný čas medzi poruchami, pokrytie porúch, dostupnosť, nedostupnosť, index dôležitosti atď. (Lisnianski, 2010). Matematická reprezentácia systému a vybraných metód v druhom kroku určuje algoritmy a metódy výpočtu týchto indexov. Algoritmy a metódy na výpočet indexov taktiež závisia od reprezentácie štruktúrnej funkcie (Murchland, 1975; Barlow 1978; Natvig 2010).

Po vypočítaní indexov je možné vykonať analýzu ich hodnôt. Meranie a zlepšovanie spoľahlivosti systému sa vykonáva vo štvrtom kroku pri vývoji stratégií pre zvýšenie spoľahlivosti systému, udržiavateľnosti a ďalších vlastností spoľahlivosti systému.

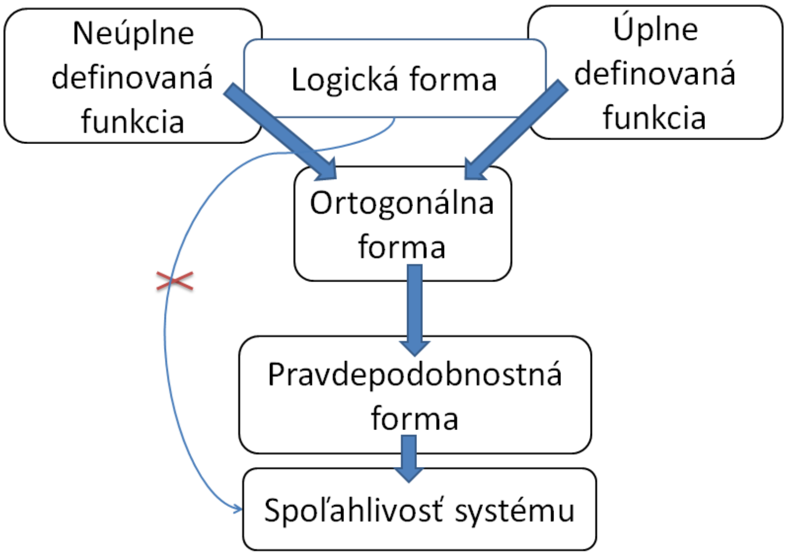
Ako vyplýva z analýzy základných krokov vytvorenia matematickej reprezentácie, matematická reprezentácia ľubovoľného systému sa začína definovaním počtu stavov a vývojom matematického opisu systému, ktorý úzko súvisí s matematickou metódou použitou pre vyhodnotenie systému. Preto sú tieto dva kroky pre nás najdôležitejšie a až po ich spracovaní môžeme vykonať ďalšiu analýzu zložitých systémov.

V tejto práci sa rozoberá analýza BSS a MSS. Analýza možného matematického opisu navrhnutá vyššie nám umožňuje zvoliť si štruktúrnu funkciu, pretože tento matematický opis sa dá skonštruovať pre systém akejkoľvek štrukturálnej zložitosti (Griffith 1980, Lisnianski & Levitin 2003). Štruktúrna funkcia definuje univalentnú koreláciu úrovne výkonnosti systému a stavov komponentov. Metódy založené na štruktúrnej funkcii boli vyvíjané a rozširované mnohými výskumami, napríklad v (Levitin, 2009, Zio 2019, Ushakov 2006).

Matematické metódy na vyhodnotenie štruktúrnej funkcie sú často založené na metódach algebry logiky. V prípade BSS sa tieto metódy vyvíjajú s použitím booleovskej logiky (Wood 1985, Schneeweiss 2009, Ryabinin 1981). Vyhodnotenie štruktúrnej funkcie MSS sa vykonáva pomocou viachodnotovej logiky (Zaitseva 2017, Rauzy, 2001). Dôležitou podmienkou väčšiny metód založených na štruktúrnej funkcii je reprezentácia štruktúrnej funkcie v ortogonálnej forme (Schneeweiss 2009, Ryabinin 1981, Rauzy, 2001). Táto forma je dôležitá pre reprezentáciu štruktúrnej funkcie, pretože umožňuje veľmi jednoduchú transformáciu logickej interpretácie štruktúrnej funkcie do pravdepodobnostnej formy (Ryabinin 1981, Griffith 1980, Reinske & Ushakov1988, Schneeweiss 2009, Sellers & Singpurwalla 2008). Väčšinu indexov (spoľahlivosť, nedostupnosť, indexy dôležitosti a iné) možno vypočítať iba na základe pravdepodobnostnej formy (Obrázok 2). Preto výpočet takýchto indexov vyžaduje pravdepodobnostnú formu štruktúrnej funkcie, ktorú je možné získať na základe logického ortogonálneho tvaru štruktúrnej funkcie. To vyžaduje vývoj algoritmov na ortogonalizáciu pôvodnej štruktúrnej funkcie (Ryabinin 1981).

Problém ortogonalizácie logických funkcií je typický problém v algebre logiky (Miller & Aaron 2008, Stankovic, Astola & Moraga 2012). Existujú viaceré ortogonálne formy pre logické funkcie. Jednou zo známych foriem je úplna disjunktívna normálna forma. Dôležitou nevýhodou tejto formy je veľká dimenzia, ktorá súvisí s počtom nenulových hodnôt booleovskej funkcie (Ryabinin 1981, Smirnov & Gajdamovich 2001, Rausand & Hoyland 2007). Preto je logická funkcia zvyčajne minimalizovaná a potom je pre túto funkciu implementovaná ortogonalizácia (Ryabinin 1981, Wood 1985). Existuje niekoľko metód na ortogonalizáciu booleovských funkcií, ktoré môžu byť efektívne použité na vytvorenie ortogonálnej štruktúrnej funkcie v analýze spoľahlivosti BSS. Ide najmä o metódu, ktorú navrhol prof. A. Ryabinin v (Ryabinin 1981) na základe vytvorenia špeciálnej maticovej transformácie. Avšak táto metóda nemôže byť vhodne použitá pre funkciu s veľkou dimenziou. Podľa vyhodnotenia v (Ryabinin 1981) sa táto metóda môže použiť pre funkciu s 20 premennými, čo znamená analýzu BSS iba s 20 komponentmi. Analýza ďalších skúmaní logických funkcií v ortogonalizácii ukázala, že prístup, ktorý navrhli prof. A. Zakrevskij a prof. Yu. Pottosín v (Zakrevskij a Pottosin 2005) sa môže použiť na funkciu s veľkou dimenziou a môže byť použitý na analýzu spoľahlivosti štruktúrnych funkcií. Táto metóda však bola vyvinutá iba pre booleovskú funkciu. Je potrebné poznamenať, že problém ortogonalizácie vo viachodnotovej logike nie je jednoznačne definovaná.

Problém ortogonalizácie vo viachodnotovej logike úzko súvisí s problémom minimalizácie logických funkcií, pretože funkcie vo viachodnotovej logike majú veľkú dimenziu (Petrik 2008). Preto by ortogonalizácia štruktúrnej funkcie MSS mala zahŕňať minimalizáciu tejto funkcie v prípade, že táto funkcia je vytvorená ako disjunktívna normálna forma. Jednu z možných adaptácii a interpretácii ortogonalizačného problému booleovskej logiky navrhol prof. M. Perkowski (Perkowski 1992). Tento výskum by sa mal rozpracovať na použitie pri analýze spoľahlivosti MSS. Na základe spomenutých faktov je ortogonalizácia podstatným problémom, ktorý by sa mal brať do úvahy pri aktuálnom výskume v teórii spoľahlivosti. Tento problém by sa mal obzvlášť zohľadniť pri analýze MSS (Sellers & Singpurwalla 2008).



Obrázok 2 Prechod z logickej formy na spoľahlivosť systému

Na základe vyššie spomenutých skutočností je preto hlavným cieľom práce vývoj a zdokonaľovanie matematického prístupu analýzy spoľahlivosti MSS pri vytváraní matematickej interpretácie skúmaného systému vo forme štruktúrnej funkcie s uplatnením matematického prístupu viachodnotovej logiky. Tento cieľ nás vedie ku skúmaniu vývoja metód na zostrojenie ortogonálnej formy štruktúrnej funkcie BSS a MSS. Vývoj takýchto metód vedie k nasledovným úlohám:

- pokračovanie výskumu z (Zakrevskij a Pottosin 2005) a vývoj algoritmu pre ortogonalizáciu štruktúrnej funkcie BSS na základe metódy navrhnutej autormi v (Zakrevskij a Pottosin 2005);
- analýza koncepcie ortogonálnej formy pre funkciu viachodnotovej logiky a definícia koncepcie ortogonalizácie pre štruktúrnu funkciu MSS;
- vývoj algoritmov pre minimalizáciu a ortogonalizáciu štruktúrnej funkcie MSS;
- validácia vyvinutých algoritmov pre ortogonalizáciu BSS a MSS na vybraných systémoch (štruktúrne funkcie BSS);
- analýza efektívnosti navrhovaných algoritmov.

2 Metódy založené na štruktúrnych funkciách

Kvantitatívne hodnotenie spoľahlivosti akéhokoľvek systému je možné na základe matematického znázornenia skúmaného systému.

2.1 Štruktúrna funkcia

Štruktúrna funkcia je jedna z možných matematických modelov reprezentujúcich skutočný systém v teórií spoľahlivosti. Štruktúrna funkcia udáva úroveň výkonu systému (spoľahlivosť/práceschopnosť) v závislosti od jeho stavov komponentov (Natvig 2010, Zio 2009):

$$\phi(\boldsymbol{x})=\phi(x_1,\ldots, x_n)\colon \{0,\ldots,m_1 -1\}\times\ldots\times\{0,\ldots,m_n -1\}\rightarrow\{0,\ldots,M -1\}, \quad (1)$$

kde $\phi(\boldsymbol{x})$ je stav systému od jeho zlyhania ($\phi(\boldsymbol{x}) = 0$) po dokonalú funkčnosť ($\phi(\boldsymbol{x}) = M - 1$); $\boldsymbol{x} = (x_1,\ldots, x_n)$je stavový vektor; x_i je stav komponentu, ktorý sa mení od stavu zlyhania ($x_i = 0$) po dokonalú funkčnosť ($x_i = m_i -1$).

Systém so štruktúrnou funkciou (1) je viacstavový (MSS) a umožňuje nám reprezentovať a skúmať niektoré úrovne výkonnosti systému. Ak $M = m_i = 2$ štruktúrna funkcia (1) reprezentuje dvojstavový systém (BSS) , ktorý nám umožňuje analyzovať 2 systémové stavy: zlyhanie a bezchybné fungovanie. Štruktúrna funkcia (1) môže byť reprezentovaná ako klasifikačný model. Vzhľadom na túto reprezentáciu sú všetky vektory stavov systému (x_1, ..., x_n) rozdelené do M tried (Zaitseva, 2016).

Vzhľadom na matematickú definíciu (1) sa premenná štruktúrnej funkcie interpretuje ako komponent systému. Štruktúrna funkcia umožňuje reprezentáciu rôznych systémov.

Štruktúrna funkcia má rôzne vlastnosti v závislosti od typu skúmaného systému. V tejto práci sa uvažuje o koherentných systémoch, to znamená:

- štruktúrna funkcia je monotónna: $\phi((s -1)_i, \boldsymbol{x}) \leq \phi(s_i, \boldsymbol{x})$ pre každé $i \in \{1,\ldots, n\}$ a $s \in \{1,\ldots, m_i -1\}$;
- komponenty obsiahnuté v systéme nie sú irelevantné, kde $\phi(s_\mathrm{i}, \boldsymbol{x})=(x_1,\ldots,x_{\mathrm{i}-1},s,x_{\mathrm{i}+1},\ldots,x_\mathrm{n})$.

Hodnotenie MSS na základe štruktúrnej funkcie predpokladá vyjadrenie pravdepodobnosti jednotlivých stavov pre každú zložku systému.

Metódy posudzovania spoľahlivosti systému založené na reprezentácii štruktúrnych funkcií sú pevne stanovené. Tieto metódy sú deterministické a používajú sa pri kvantitatívnej a kvalitatívnej analýze. Štruktúrnu funkciu je možné vytvoriť na základe úplne špecifikovaných údajov, ktoré indikujú korelácie všetkých komponentov a ich stavov. Takéto údaje pre väčšinu systémov v reálnom svete sú neúplné a neisté. Typickým príkladom je analýza a hodnotenie ľudského faktora.

Štruktúrna funkciu BSS sa interpretuje ako logická funkcia. Táto funkcia popisuje logické prepojenie prvkov v systéme, ale neumožňuje analyzovať pravdepodobnostné podmienky - je to logická funkcia - neumožňuje nám povedať nič o spoľahlivosti systému - teda o pravdepodobnosti, že systém vykonáva svoje funkcie počas definovaného času za predpokladu, že to fungoval na začiatku. Je dôležité prejsť od logickej cez ortogonálnu formu k pravdepodobnostnej forme a tak prejsť k spoľahlivosti systému. MSS sa bude interpretovať ako funkcia viachodnotovej logiky.

Pravdepodobnosť úrovne výkonnosti systému je definovaná pre každú úroveň výkonnosti ako:

$$A_j = \Pr\{\phi(x) = j\}, j = 1, \ldots, M - 1. \quad (2)$$

V prácach (Barlow 1978, Hudson 1983, Lisnianski 2003) autori ukázali, že akýkoľvek stav systému s (j = 1,…, M -1) pre pevne stanovené komponenty koherentného MSS podľa predpokladu možno vypočítať ako súčet pravdepodobnosti stavov komponentov:

$$p_{is} = \Pr\{x_i = s\}, s = 0, \ldots, m_i - 1. \quad (3)$$

Ako bolo ukázané v prácach (Barlow 1978, Hudson 1983), štrukturálna funkcia (1) sa môže použiť na výpočet práceschopnosti systému (2), ak premenné štruktúrnej funkcie opisujú nezávislé udalosti. Toto je možné, ak štruktúrna funkcia je v kanonickej a ortogonálnej forme. Na výpočet práceschopnosti systému sa používajú dve vety z teórie pravdepodobnosti (2):

3. Pravdepodobnosť súčinu nezávislých udalostí a a b (súbežná udalosť) sa rovná súčinu pravdepodobností týchto udalostí:

$$Pr(ab) = Pr(a)Pr(b). \quad (4)$$

4. Pravdepodobnosť súčtu nezlučiteľných udalostí a a b (najmenej jedna z nich nastane) sa rovná súčtu pravdepodobností týchto udalostí:

$$Pr(a + b) = Pr(a) + Pr(b). \quad (5)$$

Praktické použitie dvoch viet (4) a (5) predpokladá zmenu premenných x_i (i = 1, …, n) štruktúrnej funkcie (1) pravdepodobnosťou stavov komponentov systému (3), ak je štruktúrna funkcia opísaná kanonickou a ortogonálnou formou. V (Caldarola, 1980) je ukázané, že pri interpretácii koherentných MSS možno pravdepodobnosť stavu systému j

(j = 0, ..., $m - 1$) vypočítať pre vektor vo fixnom stave $\boldsymbol{x} = (x_1,\ldots,x_n)$ ako súčin pravdepodobností $\Pr\{x_i = s\}$stavov komponentov, kde $s = 0, \ldots, m_i-1$ definuje možné stavy komponentu i. Jednou z podmienok nekoherentného systému je to, že premenné sú vzájomne nezávislé a preto môžeme na ich analýzu použiť pravidlo (4). To znamená, že ak pracujeme s 2 premennými a vypočítame pravdepodobnosť stavu, keď prvá premenná zlyhala a druhá premenná je v stave 1 (funkčná), použijeme iba pravdepodobnosť prvej premennej (zlyhanie) a pravdepodobnosť druhej premennej (fungovanie) a aplikujeme násobenie, pretože tieto premenné sú nezávislé udalosti. Ak sa uvažuje o všetkých možných stavov, pri ktorých zlyhá systém, musia byť tieto stavy navzájom nekompatibilné (5), čo znamená, že jedna premenná nemôže byť v rovnakom stave pre funkčný a zlyhaný systém.

Dôležitým aspektom pri výpočte práceschopnosti systému je preto konštrukcia kanonickej a ortogonálnej formy štruktúrnej funkcie. Tento aspekt možno skúmať na základe metód booleovskej logiky pre BSS a na základe viachodnotovej logiky pre MSS.

3 Metódy tvorby štruktúrnych funkcií

3.1 Booleovská algebra

Boolovská algebra v abstraktnej algebre je definovaná ako komplementárne a distribučné zjednotenie a tento typ algebraickej štruktúry obsahuje základné vlastnosti množinových a logických operácií.

Booleovská logika je forma algebry, ktorá je sústredená okolo troch základných booleovských operácií OR(alebo), AND (a) a NOT(negácia).

Booleovská algebra je definovaná na množine dvoch prvkov, M = {0, 1}. Operácie booleovskej algebry dodržiavajú určité vlastnosti, nazývané zákony alebo axiómy, ktoré sa používajú na preukázanie všeobecnejších zákonov o booleovských výrazoch, aby sa napríklad výrazy zjednodušili.

3.2 Viachodnotová algebra

Viachodnotová algebra je zovšeobecnením booleovskej algebry založenej na súbore m prvkov $M = \{0, 1, 2, . . . ,m\}$. Primárnou výhodou viachodnotového systému je schopnosť kódovať viac informácií na premennú, ako dokáže binárny systém (Yanushkevich, 2006).

Definícia viachodnotovej logiky:

- Abeceda {0, 1, ..., m-1};
- Minimálne dve operácie: „*“ a „+“
- Konštanta „0“: 0 * X = 0 a 0 + 0 = 0, X ∈ {0, 1,…, m-1}

Operácie booleovskej algebry majú svoje analógie vo viachodnotovej algebre. Viachodnotové náprotivky binárnych operátorov „Alebo“, „A“ sú viachodnotovými konjunkciami, disjunkciami.

Typickým prístupom pre analýzu štruktúrnej funkcie MSS je zovšeobecnenie metód pre analýzu štruktúrnej funkcie BSS, ktoré sú založené na booleovskej logike. Tento prístup však neumožňuje použitie všetkých detailov MSS. Ďalší prístup je založený na použití matematických metód MVL pre analýzu štruktúrnych funkcií MSS. Podľa tohto prístupu je štruktúrna funkcia interpretovaná ako funkcia MVL.

3.3 Ortogonalizácia

Ortogonalizácia je veľmi dôležitá pre spracovanie štruktúrnej funkcie pri analýze spoľahlivosti. Očakáva nezávislé udalosti. Jej použitím sa môže ľahko prejsť z logickej na pravdepodobnostnú formu.

V niektorých vedeckých článkoch sa uvažovalo o koncepcii ortogonalizácie v MVL (Perkowski 1992, Perkowski 1991).

V tejto práci je koncepcia ortogonalizácie braná do úvahy pre tabuľku pravdivosti štruktúrnej funkcie MSS (funkcia MVL) alebo premenných vektorov štruktúrnej funkcie. Zvažujú sa dva typy ortogonalizácie: úplné ortogonálne premenné vektory a čiastočne ortogonálne premenné vektory.

Podobne ako booleovská logika, dva vektory premenných štruktúrnej funkcie MSS (funkcia MVL) sú ortogonálne, ak sú vzájomne disjunktné. Majme dva vektory premenných $a_1…a_j…a_n$ a $b_1…b_j…b_n$ kde a_i, b_i ∈ {0, …, m_i-1}. Tieto vektory sú ortogonálne, ak aspoň jedna dvojica premenných spĺňa podmienku $a_i \neq b_i$.

Sada úplne ortogonálnych vektorov premenných pozostáva z m_i vektorov premenných pre ktoré existuje jedna premenná, ktorá má v každom vektore rôzne hodnoty, a ostatné hodnoty premenných sú rovnaké. Premenná sa nazýva ortogonálna premenná, ak má pre tieto vektory rôzne hodnoty.

Ak máme napríklad štruktúrnu funkciu MSS s m_i = m_s =M =3 (všetky vstupné a výstupné premenné môžu byť v 3 stavoch - štruktúrna funkcia (1) je homogénna, ak M =

$m_i = m_j$ for $i \neq j$) a n = 4. Vektory premenných 0102, 0112, 0122 sú úplné ortogonálne, pretože x_1, x_2 a x_4 majú rovnaké hodnoty a x_3 má rôzne hodnoty od 0 do 2.

Sada čiastočne ortogonálnych vektorov premenných pozostáva z s vektorov premenných ($2 \leq s \leq m_i$-1) pre ktoré existuje jedna premenná, ktorá má v každom vektore rôzne hodnoty, a ostatné hodnoty premenných sú rovnaké. Premenná sa nazýva ortogonálna premenná, ak má pre tieto vektory rôzne hodnoty.

Dva vektory premenných 0112 a 0122 sú čiastočne ortogonálne pre štruktúrnu funkciu MSS s n = 4 komponentmi a pre $m_i = m_s = M = 3$. Premenné x_1, x_2 a x_4 majú rovnaké hodnoty a x_3 má rôzne hodnoty od 1 a 2.

Koncepcie úplnej a čiastočne ortogonalizácie sa používajú v algoritme na minimalizáciu tabuľky pravdivosti štruktúrnej funkcie MSS.

3.3.1 Boolovská funkcia

Pojem ortogonalizácie je dobre známy a často sa používa v booleovskej algebre pre vektory premenných (Schneeweiss 2009, Hudson 1983). Podľa (Hudson 1983) sú dva konjuktívne členy booleovskej funkcie ortogonálne, ak je ich súčin nula alebo ak sú navzájom disjunktné. Pojem ortogonálny konjuktívny člen môže byť zovšeobecnený pre vektory premenných: dva vektory premenných sú ortogonálne, ak je ich súčin nula. Matematický opis booleovskej funkcie je ortogonálny, ak sú všetky konjuktívne členy ortogonálne. Podľa (Barlow 1978, Barlow 1975) sa ortogonálna forma štruktúrnej funkcie (1) transformuje na pravdepodobnostnú formu substitúciou premenných štruktúrnych funkcií pravdepodobnosťou stavov príslušných komponentov.

3.3.2 MVL funkcia

Ortogonalizácia DNF v tejto práci je založená na disjunktívnom rozširovaní elementárnych konjunkcií do mnohých ďalších konjunkcií, ktoré sa stali ortogonálnymi z definovanej skupiny alebo boli absorbované. Na zníženie ďalších výpočtov sa absorbovaná konjunkcia z výsledku odstráni a zostávajúce spojenia sú minimalizované. Ortogonalizačný algoritmus opísaný v Zakrevskij a Pottosin (Zakrevskij, 2005) je v tejto práci adaptovaný na proces ortogonalizácie.

Ortogonálne DNF - je také DNF, ktoré obsahuje ortogonálne elementárne konjunkcie a to znamená násobenie týchto konjunkcií, dáva 0. Algoritmus pracuje na princípe operácie rozširujúcej k_i nad k_j, kde k_i a k_j sú neortogonálne elementárne konjunkcie.

3.4 Minimalizácia

Proces minimalizácie sa používa na zníženie alebo zjednodušenie štruktúrnej funkcie, ktorá môže byť v reálnych systémoch veľmi zložitá. Vďaka procesu minimalizácie môžeme výrazne znížiť štruktúrnu funkciu, čo vedie k jednoduchšej výpočtovej zložitosti.

3.4.1 Booleovská funkcia

Konjuktívny člen sa považuje za implikant úplnej funkcie, ak implikuje funkciu. Preto každý z konjuktívnych členov opisujúcich úplnú booleovskú funkciu je implikantom pre túto funkciu. Inými slovami, môžeme povedať, že elementárne konjuktívne členy funkcií sú ich implikantmi.

Implikant booleovskej funkcie sa nazýva je prvoimplikant, ak tento implikant nezahrňuje iný implikant s menším počtom literálov tej istej funkcie. Napríklad, majme booleovskú funkciu:

$$f(A,B,C) = A\bar{B}C + \bar{B}C. \tag{6}$$

V tejto funkcii je $A\bar{B}C$ implikantom funkcie. Na druhej implikant $\bar{B}C$ vo funkcii nezahrňuje iný implikant a preto je prvoimplikantom. Je dôležité si uvedomiť, že ak je booleovský výraz vyjadrený ako súčet prvoimplikantov, potom zodpovedá minimálnej disjunktívnej normálnej forme.

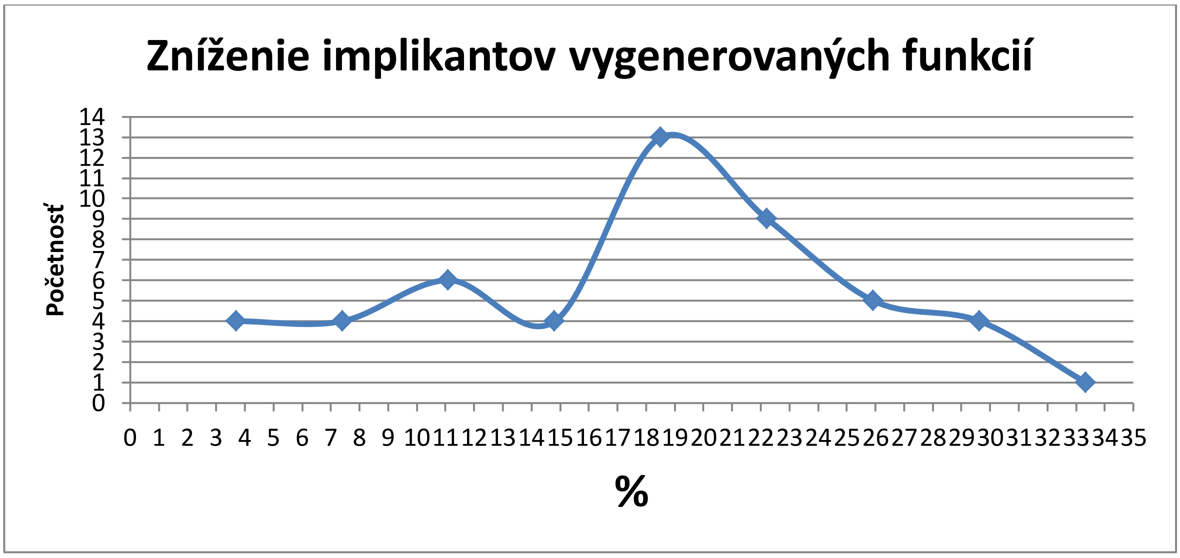
3.4.2 MVL funkcia

Pravdivostná tabuľka štruktúrnej funkcie MSS je ortogonálna forma znázornenia funkcie. To znamená, že všetky vektory premenných v tejto tabuľke sú ortogonálne. Pravdivostná tabuľka má však dimenziu $m_1 \times m_2 \times \ldots \times m_n$, čo komplikuje túto tabuľkovú analýzu a hodnotenie. Minimalizácia pravdivostnej tabuľky je zostrojenie pravdivostnej tabuľky s menšou dimenziou a bez straty informácií o správaní systému. Jedným zo spôsobov minimalizovania pravdivostnej tabuľky je odstránenie vektorov premenných, ktoré nemajú vplyv na hodnotu funkcie. Spravidla je to ortogonálny vektor premenných vzhľadom na jednu premennú pre rovnakú hodnotu funkcie. Vytvorený vektor premenných má menší počet premenných ako počiatočné vektory premenných. Preto vytvorená pravdivostná tabuľka obsahuje menší počet vektorov premenných a tieto vektory obsahujú menší počet premenných. Do úvahy sa môže vziať úplna a čiastočná ortogonalizácia premenných. Podobný princíp sa používa v algoritme Quine – McCluskey na minimalizáciu booleovských funkcií (McCluskey, 1956).

4 Experimenty

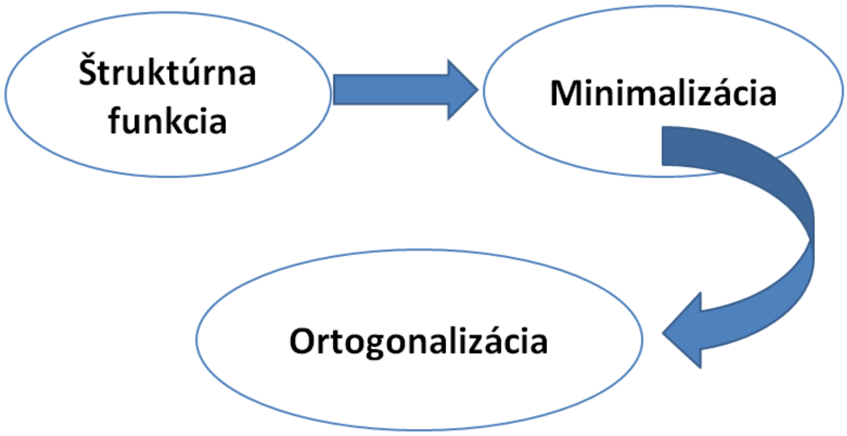
Vyhodnotenie efektívnosti minimalizácie logických funkcií bolo implementované na základe súboru generovaných monotónnych funkcií. Monotónne logické funkcie sa zhodujú s definíciou koherentných štruktúrnych funkcií, ktoré umožňujú reprezentáciu väčšiny skutočných systémov. Generovaná množina obsahuje štrukturálnu funkciu MSS s 3 výkonnostnými úrovňami systému a 3 stavmi všetkých systémových komponentov. Počet funkcií je 50. Tieto funkcie boli minimalizované vyvinutým algoritmom na minimalizáciu štruktúrnych funkcií MSS (tento algoritmus je uvedený v oddiele 2.4.2).

Získali sme výsledky, ktoré sú znázornené na Obrázku 3. Z grafu vidíme, že maximum poklesu implikantov pre vygenerované funkcie bolo okolo 33% a minimum okolo 3%. Najbežnejšia hodnota poklesu bola asi 18% a táto hodnota bola meraná pre 13 funkcií.



Obrázok 3 Zníženie implikantov vygenerovaných funkcií

Postup spracovania je znázornený na Obrázku 4. Definovali sme systém opísaný štruktúrnou funkciou a najskôr využívame algoritmy na minimalizáciu na zjednodušenie tejto štruktúrnej funkcie, ktorá môže byť v reálnych systémoch veľmi zložitá. Vďaka procesu minimalizácie môžeme výrazne znížiť túto funkciu, čo vedie k jednoduchšej výpočtovej zložitosti. Ak existuje napríklad piata mocnina čísla 3, existuje 243 možných definovaných implikantov pre takúto štruktúrnu funkciu. Ak sa táto funkcia dá zjednodušiť minimalizáciou napríklad na 150 implikantov, táto minimálna forma môže byť už ortogonálna, ale nie je to zaručené a preto sa použije ďalší proces ortogonalizácie. Na druhej strane po orthogonalizácii sa môže zvýšiť počet implikantov, a preto je pre nás dôležitejšie, ako dlho trvá proces na ortogonálnu formu, ktorá sa môže alebo nemusí zjednodušiť.



Obrázok 4 Spracovanie systému definovaného štruktúrnou funkciou

Záver

Analýza spoľahlivosti je hodnotným výskumom pre rôzne systémy. Analýza ľubovoľného systému začína definovaním počtu stavov a potom pokračuje vývojom matematického opisu systému. Preto sú tieto dva kroky pre nás najdôležitejšie a až po ich spracovaní môžeme vykonať ďalšiu analýzu zložitých systémov. Existujú dva prístupy k reprezentácii systému, ktorých popis je založený na BSS a MSS. BSS umožňuje reprezentovať počiatočný systém ako matematický model s dvoma možnými stavmi, ktoré sú úplné zlyhanie a perfektné fungovanie. MSS umožňuje brať do úvahy viac ako iba dva stavy v správaní sa práceschopnosti alebo spoľahlivosti systému.

Keď sme ako matematickú reprezentáciu systému vybrali štruktúrnu funkciu, táto funkcia nemá vždy formu akú potrebujeme, a preto sú potrebné určité zapracovania a zmeny v tejto funkcii. To nás vedie k vývoju algoritmov a metód na konštrukciu ortogonálnej a minimálnej reprezentácie štruktúrnej funkcie MSS. Takáto forma reprezentácie umožňuje prechod od logického k pravdepodobnostnému opisu a potom sa dostane k spoľahlivosti systému.

Predstavili sme model spracovania skutočného systému, ktorým bola dronová letka. Tento systém sme definovali štruktúrnou funkciou a potom sme ukázali, že vďaka minimalizácii môžeme túto funkciu výrazne zjednodušiť, čo vedie k jednoduchšej výpočtovej zložitosti. Štruktúrna funkcia po minimalizácii môže byť už v ortogonálnej forme, ale to nie je zaručené, preto ďalším procesom bolo použitie procesu ortogonalizácie. Na druhej strane po ortogonalizácii sa môže zvýšiť počet implikantov, ktoré boli znížené po procese minmalizácie, a preto je pre nás dôležitejšie, ako dlho trvá proces premeny na ortogonálnu formu, ktorá sa môže alebo nemusí zjednodušiť štruktúrnu funkciu.

Podľa hlavného cieľa boli spracované ďalšie úlohy:

- metóda navrhnutá v (Zakrevskij & Pottosin 2005) bola upravená na aplikáciu na ortogonalizáciu štruktúrnej funkcie BSS;
- zvážila sa koncepcia ortogonálnej formy vo viachodnotovej logike a navrhla sa koncepcia ortogonálnej štruktúrnej funkcie MSS;
- navrhovaná koncepcia ortogonálnej štruktúrnej funkcie MSS sa použila na vývoj algoritmov minimalizácie a ortogonalizácie štruktúrnej funkcie MSS;
- vytvorené algoritmy boli overené a vyhodnotené na vybraných systémoch (štruktúrne funkcie BSS a MSS).

References

1. Akers, S. B.: Binary Decision Diagrams, IEEE Transactions on Computers, C-27(6):509–516, 1978
2. Amari S. V., Xing L., Shrestha, A., Akers, J., Trivedi K. S., "Performability analysis of multistate computing systems using multivalued decision diagrams." IEEE Trans. Comput. 59, 1419–1433 (2010).
3. Andrews, J., & Dunnett, S. (2000). Event-Tree Analysis Using Binary Decision Diagrams. *IEEE Transactions on Reliability*, 230-238.
4. Aven T., Heide B. (2009), Reliability and validity of risk analysis, *Reliability Engineering and System Safety*, Vol. 94 (11), pp.1862–1868
5. Aven, T., Zio, E., Baraldi, P., Flage, R., Uncertainty in Risk Assessment: The Representation and Treatment of Uncertainties by Probabilistic and Non-Probabilistic Methods, Wiley, 2014
6. Aven, T. (2017). Improving the foundation and practice of reliability engineering. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *231*(3), 295–305.
7. Barlow R.E., Wu A.S. (1978) Coherent System with Multi-State component, *Math. Operations Research*, 3(11), pp.275-281.
8. Barlow, R. & Proschan, F. (1975). Statistical Theory of Reliability and Life Testing: Probability Models. New York: Holt, Rinehart and Winston, Inc.
9. Barlow R. E. (1998) *Engineering reliability*. ASA and SIAM, Philadelphia, PA.
10. Baybutt, P. (2015). A critique of the Hazard and Operability (HAZOP) study. *Journal of Loss Prevention in the Process Industries*, 52-58.
11. Birolini, A. (2014) Reliability engineering: Theory and practice. Seventh edition. Springer, 626 p.
12. Bris, R., Domesova S. (2014). New Computing Technology in Reliability Engineering, Mathmatical Problem in Engineering, Article ID 187362, 2014.
13. Caldarola, "Coherent System with Multi-State Components," *Nucl. Eng. Des.*, vol. 58, pp. 127–139, 1980.
14. Crama, Y., Hammer, P. L.,: *BOOLEAN FUNCTIONS - Theory, Algorithms, and Applications*.: Cambridge University Press, 2011.
15. Dhillon B.S. (2003), *Human Reliability and Error in Medicine*, World Scientific.

16. Donald E. Knuth, *The Art of Computer Programming, Volume 4 Fascicle 0, Introduction to Combinatorial Algorithms and Boolean Functions*.: Addison-Wesley Professional, 2008.
17. Dunn, J. M., Hardegree, G. M.,: *Algebraic Methods in Philosophical Logic*.: Oxford University Press US, 2001.
18. Dutuit, Y., Rauzy, A., Efficient algorithms to assess component and gate importance in fault tree analysis, Reliability Engineering & System Safety, Vol.72, N.2, 2001, pp 213-222
19. Griffith W.S. (1980). Multistate reliability models, *J Appl Probab*, 17, pp. 735-744
20. Hilary A. Priestley and Brian A. Davey, *Introduction to Lattices and Order*.: Cambridge University Press, 2002.
21. Hudson, J.C., Kapur, K.C.: Modules in Coherent Multistate Systems. IEEE Transactions on Reliability. 32, 183–185 (1983)
22. Hyun, K.-C., Min, S., Choi, H., Park, J., & Lee, I.-M. (2015). Risk analysis using fault-tree analysis (FTA) and analytic hierarchy process (AHP) applicable to shield TBM tunnels. *Tunnelling and Underground Space Technology*, 121-129.
23. Kolowrocki K. (2014) *Reliability of Large and Complex Systems, 2nd Edition*, Elsevier, 460 p.
24. Kuo, S.-Y., Yeh, F.-M., Lin, H.-Y.: "Efficient and exact reliability evaluation for networks with imperfect vertices," IEEE Transactions on Reliability 56 (2) (2007) 288-300.
25. Kvassay, M. Zaitseva, E. and Levashenko, V. "Importance analysis of multi-state systems based on tools of logical differential calculus," *Reliab. Eng. Syst. Saf.*, vol. 165, pp. 302–316, Sep. 2017.
26. Levashenko, V., Zaitseva, E., Kvassay, M., Deserno, T.M.: Reliability estimation of healthcare systems using Fuzzy Decision Trees. In: Proc. of the 2016 Federated Conf.e on Computer Science and Information Systems, FedCSIS 2016 (2016).
27. Levitin G. (2009) Optimizing defense strategies for complex multi-state systems. In: Bier VM, Azaiez MN, editors. *Game theoretic risk analysis of security threats*, Springer, pp. 33–64.
28. Lisnianski A., Levitin G. (2003), *Multi-state System Reliability. Assessment, Optimization and Applications*. Singapore, SG: World Scientific.

29. Lisnianski, A., Frenkel, I., Karagrigoriou, A., Recent Advances in Multi-state Systems Reliability, Springer, 2018
30. Lisnianski, A., Frenkel, I. and Ding, Y. “Multi-state system reliability analysis and optimization for engineers and industrial managers,” *Multi-State Syst. Reliab. Anal. Optim. Eng. Ind. Manag.*, pp. 1–393, 2010.
31. Lyons M., Adams S., Woloshynowych M. and Vincent C.,(2004) Human reliability analysis in healthcare: A review of techniques, *Int. J. of Risk and Safety in Medicine*, Vol. 16 (4), pp. 223–237.
32. Merle, G., Roussel, J.M., Lesage, J.J.: Dynamic Fault Tree Analysis Based On The Structure Function. Annual Reliability and Maintainability Symposium 2011 (RAMS 2011), Jan 2011, Orlando, FL, United States. pp. 462-467.
33. Miller D. M., Drechsler R., “On the Construction of Multiple-Valued Decision Diagrams.” Proc. 32nd IEEE Int. Symp. Mult. Log. 245–253 (2002).
34. Miller D.M., Aaron M. (2008). Multiple Valued Logic: Concepts and Representations, Morgan&Claypool
35. Mo Y., “A multiple-valued decision-diagram-based approach to solve dynamic fault trees.” IEEE Trans. Reliab. 63, 81–93 (2014).
36. Mo, Y., Xing, L., Zhong, F., Pan, Z. Chen, Z. “Choosing a heuristic and root node for edge ordering in BDD-based network reliability analysis,” Reliability Engineering & System Safety 131 (2014) 83-93.
37. Mohd K. R., J. S. (2006). Reliability Estimation of Uninterruptible Power Supply Systems: Boolean Truth Table Method. *INTELEC 06 - Twenty-Eighth International Telecommunications Energy Conference*.
38. Murchland J.D. (1975) Fundamental Concepts and relations for Reliability Analysis of Multistate System, *Reliability and Fault Tree Analysis, Theoretical and Applied Aspects of System Reliability*. SIAM, pp.581-618.
39. Natvig, B., “Multistate systems reliability theory with applications,” *Multistate Syst. Reliab. Theory with Appl.*, pp. 1–232, 2010.
40. Patel, H.R., Linares, A., Joseph, J.V.: Robotic and laparoscopic surgery: cost and training. Surg Oncol, vol.18, no.3, pp. 242-246 (2009).
41. Perkowski, M.: „The generalized orthonormal expansion of functions with multiple-valued inputs and some of its applications.“ pp. 442 - 450. (1992).

42. Perkowski, M., Johnson, P. D.: “Canonical Multi-Valued Input Reed-Muller Trees and Forms.” 3rd NASA Symposium on VLSI Design, 11.3.1- 11.3.13 (1991).
43. Petrik M. (2008). Quine–McCluskey method for many-valued logical functions, Soft Computing 12(4), pp. 393-402
44. Philippart, M. (2018). Human reliability analysis methods and tools. In *Space Safety and Human Performance* (s. 501-568). Oxford ; Cambridge, MA: Butterworth-Heinemann .
45. Reinske, K., Ushakov, I. (1988). Application of Graph Theory for Reliability Analysis. Radio i Sviaz, Moscow, USSR (in Russian)
46. Ramzali, N., Lavasani, M., & Ghodousi, J. (2015). Safety barriers analysis of offshore drilling system by employing Fuzzy Event Tree Analysis. *Safety Science*, 49-59.
47. Rausand M. (2011), Risk Assessment: Theory, Methods, and Applications, Wiley, Hoboken, NJ.
48. Rausand M. and Hoyland A. ”System Reliability Theory,” 2nd ed. Wiley, Hobboken, 2004.
49. Rauzy, A. (2001), “Mathematical foundations of minimal cutsets”, IEEE Transaction on Reliability, Vol.50 No. 4, pp. 389-396.
50. Rusnak, P., Kvassay, M., Zaitseva, E., Kharchenko, V. and Fesenko, H., "Reliability Assessment of Heterogeneous Drone Fleet With Sliding Redundancy," *2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Leeds, United Kingdom, 2019, pp. 19-24.
51. Ryabinin I. A., C. G. (1981). *LOGIC-ESSENTIAL METHODS RESEARCHES RELIABILITY STRUCTURAL SYSTEMS.*
52. Shalev, D. M., & Tiran, J. (2007). Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations. *Reliability Engineering and System Safety* , 1231-1241.
53. Schneeweiss W.G. A short Boolean derivation of mean failure frequency for any (also non-coherent) system, Reliability Engineering & System Safety, Vol.94, N 8, 2009, pp 1363-1367
54. Sellers K.F., Singpurwalla N.D. (2008), Many-valued Logic in Multistate and Vague Stochastic Systems, International Statistical Review vol.76, N. 2, pp.247–267

55. Solojentsev E. (2009) *Scenario Logic and Probabilistic Management of Risk in Business and Engineering*, Springer.
56. Smirnov, A.S., Gajdamovich, D.O. (2001). An analysis of reliability of structural complicated electric circuits taking into account two types of faults, *Elektrichestvo*, vol.2, pp. 50-56
57. Stankovic R. S., J. T. Astola, C. Moraga (2012). Representations of Multiple-Valued Logic Functions, Morgan &Claypool Publishers
58. Tortorella, M. "Reliability, Maintainability, and Supportability: Best Practices for Systems Engineers," John Wiley & Sons, 2015.
59. Ushakov, I. (2006) Is the Reliability Theory Still Alive, *Proc. of the 6th Int. Conf. on Reliability and Statistics in Transportation and Communication (RelStat 2006)*, Riga, Latvia, pp.188-199.
60. Verma, A., Ajit, S. and Karanki, D. (2010). *Reliability and safety engineering*. Springer, pp.15-70.
61. Wood A.P., (1985), Multistate Block Diagrams and Fault Trees, IEEE Trans on Reliability, VOL. R-34, NO. 3, pp.236-240
62. Xiao, N., Huang, H.-Z., Li, Y., He, L., & Jin, T. (2011). Multiple failure modes analysis and weighted risk priority number evaluation in FMEA. *Engineering Failure Analysis*, 1162-1170.
63. Yanushkevich, S., Miller, D., Shmerko, V. and Stanković, R. (2006). *Decision diagram techniques for micro- and nanoelectronic design handbook*. 6000 Broken Sound Parkway NW, Suite 300: CRC Press Taylor & Francis Group, pp.581-601.
64. Zaitseva E., Levashenko V., "Decision diagrams for reliability analysis of multi-state system. In: Proc. of Int. Conf. on Dependability of Computer Systems," DepCoS - RELCOMEX 2008. pp. 55–62 (2008).
65. Zaitseva, E., Levashenko, V., (2017). Reliability analysis of Multi-State System with application of Multiple-Valued Logic. International Journal of Quality & Reliability Management, Vol. 34 Issue:6, pp.862-878.
66. Zaitseva, E. (2003). Logic Algebra in reliability analysis of Multi-State System. Dynamical Systems and Geometric Theories.
67. Zaitseva E., Levashenko (2016), Construction of a Reliability Structure Function Based on Uncertain Data, *IEEE Trans on Reliability*, vol.65(4), pp.1710-1723

68. Zaitseva, E., Levashenko, V., Kostolny, J.: Multi-state system importance analysis based on direct partial logic derivative. In: Proc. of 2012 Int. Conf. on Quality, Reliability, Risk, Maintenance, and Safety Engineering, ICQR2MSE 2012. pp. 1514–1519 (2012).
69. Zaitseva, E., Kvassay, M., Levashenko, V., Deserno, T. M., Voski, V., Herrler, A.: "Qualitative evaluation of faults (mathematical incorrectness) in anatomical model for Regional Anaesthesia Simulator," *2016 International Conference on Information and Digital Technologies (IDT)*, Rzeszow, 2016, pp. 311-318.
70. Zakrevskij, A. D., Pottosin, Yu. V. 2005. International Scientific Conference: "Probability Theory, Random Processes, Mathematical Statistics and Applications": Calculating the probability of a complex event presented by a given set of critical sets. pp. 350-357.
71. Zakrevskij, A., Pottosin, Y., Cheremisinova, L.: Combinatorial algorithms of discrete mathematics ; Tallinn : Tallinn University Press, 2008.
72. Zio, E. (2007). *Introduction to the Basics of Reliability and Risk Analysis (Series on quality, reliability and engineering statistics ; v. 13)*. Singapore: World Scientific, p.222.
73. Zio E. (2009) Reliability engineering: Old problems and new challenges. *Reliability Engineering and System Safety*. 94(2), pp.125-141.
74. McCluskey E. J. (1956), "Minimization of Boolean functions," in *The Bell System Technical Journal*, vol. 35, no. 6, pp. 1417-1444.

List of Publications:

1. Rusnak, P., Kvassay, M., Forgac, A., Zaitseva, E.: "*Logic differential calculus in time-dependent analysis of a pair of system components.*" Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018. - pp. 467-472.
2. Rusnak, P., Sedlacek, P., Forgac, A., Illiashenko O., Kharchenko V.: "*Structure function based methods in evaluation of availability of healthcare system.*" In: DESSERT´2019 = Dependable systems, services & technologies. - Institute of Electrical and Electronics Engineers, 2019. - pp. 13-18.
3. Forgac, A., Rabcan J., Zaitseva E., Lukyanchuk, I.: "*Construction of the structure function of multi-state system based on incompletely specified data*," In: Contemporary complex systems and their dependability: proceedings of the thirteenth international conference on dependability and complex systems DepCoS-RELCOMEX. - Cham: Springer International Publishing AG, 2019. - pp. 184-194.
4. Forgac, A., Rusnak P.: "*Teaching module of mathematical methods in Reliability Engineering*." In: ICETA 2018: 16th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings. - New Jersey: Institute of Electrical and Electronics Engineers, 2018. - pp. 163-171.
5. Stankevich, S. A., Lubskyi, M.S., Forgac A.: "*Thermal infrared satellite imagery resolution enhancement with fuzzy logic bandpass filtering*." In: Information and digital technologies 2019: proceedings of the international conference. - Danvers: Institute of Electrical and Electronics Engineers, 2019. - pp. 428-432.
6. Forgac, A., Lukyanchuk I.: "*New algorithm for multi-valued decision diagram construction*." In: Information and digital technologies 2019 : proceedings of the international conference. - Danvers: Institute of Electrical and Electronics Engineers, 2019. - pp. 142-148.
7. Forgac, A.: "*Methods of Human Reliability Analysis*." In: Central European Researchers Journal. - (2018), pp. 17-21.
8. Kvassay, M., Rusnak, P., Stankovic, R.S., Forgac, A. : "Use of Binary Decision Diagrams in Importance Analysis Based on Minimal Cut Vectors. " In: 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications, TELSIKS 2019 - Proceedings