UDC 621.391

J. KOSTOLNY, E. ZAITSEVA

University of Zilina, Zilina, Slovakia

DECISION DIAGRAM AND DIRECT PARTIAL LOGIC DERIVATIVES IN RELIABILITY ANALYSIS OF MULTI-STATE SYSTEM

Direct Partial Logic Derivative (DPLD) is used in the reliability analysis for the calculation of the Multi-State Systems (MSS) importance measures. MSS is mathematical model that permits to define some performance levels (more than two) for the system reliability. This mathematical model causes using of the special methods for analysis because has high dimension as a rule. One of possible methods for the MSS analysis is methods based on Decision Diagram we that can easily analyzes systems with higher dimension. New algorithms for calculating DPLD by Multi-Valued Decision Diagrams of the MSS are proposed in this paper.

Key words*: reliability, decision diagram, direct partial logic derivative, multi state system.*

Introduction

The importance analysis is one of directions in reliability analysis [1, 2]. This analysis allows identifying the relatively most critical components of the system from which alternatives can be identified to improve the system reliability. Here, a component has two aspects: the structural aspect and the reliability aspect [3]. The former refers to the location of the component in the system, and the latter refers to the reliability of the physical unit installed at that location. The structural aspect is relevant in the system design when several components with distinct reliabilities can be arbitrarily assigned to several locations in the system. The reliability aspect is considered when the components are already installed in the system but there is budget to improve the system reliability through the improvement of the reliability of a component. Many *Importance Measures* (IMs) have been proposed for estimation of these aspects.

The IM quantifies the relative importance of a component, in comparison to other components, with respect to the system reliability. Every of IMs allows measuring some aspect of the influence of the system component states changes to the system reliability/availability. Basic IMs were been considered in [1 – 4]. Different mathematical tools and approaches are used for calculation of IMs [5]. There are methods of importance analysis that are based on the mathematical tools of logic algebra [2, 6, 7]. These methods are developed in this paper. In particularly new algorithms are proposed for calculation of IMs by mathematical tools of Multi-Valued Logic as Logic Differential Calculus and Multi-Valued Decision Diagram. Authors of the paper [8] considered approach for calculation of IMs based on Logic Differential Calculus and Decision Diagram. In this paper this approach is developed and new algorithms improves calculation aspects of result in [8].

1. Multi-State Systems

As a rule the initial system in reliability analysis interpreted as the system with two possible states: failure and functioning. Therefore such system presentation permits to investigate the system failure first of all. Different aspects of the system functioning isn't analyzed in this case. There is other interpretation of the system as opposed to system with two states. Multi-State System (MSS) is the mathematical model for the representation of the initial system in reliability analysis, i.e. the set of reliability indices and measures are calculated based on this representation. This model allows defining some system states (more than two). These states can be interpreted as system failure, system partial functioning and system perfect functioning, for example.

1.1. Structure function of the MSS

Consider the MSS that has M performance levels: from zero to (M-1). Each of n system components can be in one of m_i (i = 1, …, n) possible states: from the complete failure (it is 0) to the perfect functioning (it is m_i-1). A structure function is one of typical representations of the MSS [6, 9] and is defined as:

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

where x_i is the i-th component; $\mathbf{x} = (x_1, \ldots, x_n)$ is vector of components states.

The structure function (1) represents the system with two states if $m_i = m_j = M = 2$ ($i \neq j$; i, j =1, …, n).

The i-th (i =1, …, n) system component states x_i is characterized by probability of the performance rate:

$$p_{i,s} = \Pr\{x_i = s_i\}, \quad s = 0, \ldots, m-1. \quad (2)$$

There are some methods of the MSS reliability analysis [9]. Markovian methods, Monte-Carlo simulation, Logic Algebra methods are used for the MSS reliability analysis and estimation. In this paper we use the Multi-Valued Logic (MVL) mathematical tools for the MSS reliability analysis, namely the Multi-Valued Decision Diagram (MDD) for the structure function (1) representation and the Logical Differential Calculus for the MSS behavior analysis.

The Logical Differential Calculus for MSS quantification have been proposed in [8] firstly. The Logical Differential Calculus is mathematical tool that permits to analysis changes in function depending of changes of its variables. Therefore this tool can be used to evaluate influence of every system component state change. The principal disadvantage of the Logical Differential Calculus application in reliability analysis is increase of computational complexity depending on number of system component. In this case the MDD is used for the structure function representation [6].

1.2. Direct Partial Logic Derivation

The Logical Differential Calculus of MVL function includes different methods and algorithms for estimation of influence of variable/variables value change to the function value modification [6]. Direct Partial Logic Derivative (DPLD) is part of Logic Differential Calculus and can be used for analysis of dynamic properties of MVL function or MSS structure function.

The DPLD with respect to variable x_i for MSS structure function (1) permits to analyse the system performance level change from j to $\tilde{j}$ when the i-th component state changes from s to $\tilde{s}$ [6]. This change is defined by the derivative:

$$\frac{\partial\phi(j \to \tilde{j})}{\partial x_i(s \to \tilde{s})} = \begin{cases} 1, & \text{if } \phi(s_i, \boldsymbol{x}) = j \text{ and } \phi(\tilde{s}_i, \boldsymbol{x}) = \tilde{j}; \\ 0, & \text{other.} \end{cases} \quad (3)$$

where $\phi(s_i, x) = \phi(x_1,\ldots, x_{i-1}, s, x_{i+1},\ldots, x_n)$; $\phi(\tilde{s}_i, x) =$ $= \phi(x_1,\ldots, x_{i-1}, \tilde{s}, x_{i+1},\ldots, x_n)$; s, $\tilde{s} \in \{0,\ldots,m_{i-1}\}$ and j, $\tilde{j} \in \{0,\ldots,M-1\}$.

The structure function (1) of the coherent MSS has following assumptions [9]: (a) the structure function is monotone; (b) all components are independent and relevant to the system.

1.3. Multi-Valued Decision Diagram

The MDD is generalization of the Binary Decision Diagram (BDD) that is introduced for Boolean function representation in [10]. The MDD is a directed acyclic graph to represent the MVL-function [11]. For the structure function (1) this graph has M sink nodes, labelled from 0 to (M-1), representing M corresponding constant from 0 to (M-1). Each non-sink node is labelled with a structure function variable x and has m_i outgoing edges. The sink node is interpreted as a system reliability state from 0 to (M-1) and non-sink node presents either a system component. Each non-sink node has m_i edges and the first (left) is labelled the “0” edge and agrees with component fail, and the m_i-th last outgoing edge is labelled “m_i -1” edge and presents the perfect operation state of system component.

The example of the MDD for the MSS with three components (n = 3) is in Fig.1, the structure function of this MSS ($m_1 = m_2 = 2$, $m_3 = 4$ and M = 3) is defined as:

$$\phi(\mathbf{x}) = (x_1 \text{ OR } x_2) \text{ AND } x_3, \quad (4)$$

The MSS structure *The MDD*

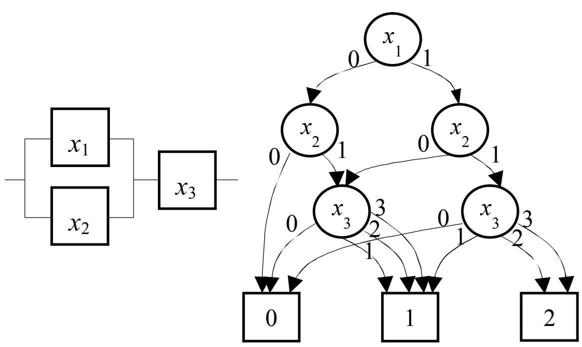

Fig. 1. The MDD example

2. Importance analysis of the MSS

2.1. Importance measures for the MSS

The MSS importance analysis is one of directions for estimation of MSS behavior against the system structure and components states [6]. In papers [12] for MSS define IM such as Structural Importance (SI), Criticality Importance (CI), Birnbaum importance (BI), Fussell-Vesely importance (FVI), Component Dynamic Reliability Indices (CDRI). Short descriptions of these measures are in Table 1.

In paper [8] algorithms for the calculation of the IMs have been considered. This calculation is implemented by the DPLD for the MSS that is presented as the MDD.

2.2. Calculation of the DPLD by the MDD

In article [8] was presented two algorithms for calculation of IMs for the MSS that is defined by the MDD. The DPLD (3) is principal tools in this calculation. These algorithms locate two sets of paths in the MDD. The first set consists of the paths that satisfy condition $\phi(s_i, x) = j$ that is path from the top-node to the sink node labelled j which includes the non-sink node of the i-th variable with value s only. The second set includes paths $\phi(\tilde{s}_i, x) = \tilde{j}$. The comparison of these sets permits to determine paths in the MDD that conform to non-zero values of DPLD (3).

Table 1

Importance Measures of MSS

Short name	Description
SI	SI concentrates on the topological structure of the system and determines the proportion of working states of system in which the working of the i-th component makes the difference between system failure and working state.
BI	BI of a given component is defined as the probability that such component is critical to MSS functioning and represents loss in the MSS when the i-th component was fails.
CI	CI measure is the probability that the i-th system component is relevant to MSS and is functioning in the specified time
CDRI	CDRI estimates the influence of the i-th component state change to MSS and is probability of MSS performance change depending on the i-th component state change.

The modification of the algorithms in [8] permits to obtain two new algorithms. Both new algorithms have identical basic principle that locates the path from the top non-sink to the sink node of the MDD that satisfy condition for the DPLD (3) calculation. This path is united in the special structure that is named "Tree of paths".

The tree of paths is formed for the conditions of the DPLD calculation: $\phi(s_i, x) = j$ or $\phi(\tilde{s}_i, x) = \tilde{j}$. Therefore the tree of paths for the condition $\phi(s_i, x) = j$ unites all paths from the root to the sink node j that include out coming edges of the non-sink node x_i labelled s. The tree of paths for the condition $\phi(\tilde{s}_i, x) = \tilde{j}$ is formed similarly.

$\phi(0_1, \mathbf{x}) = 0$

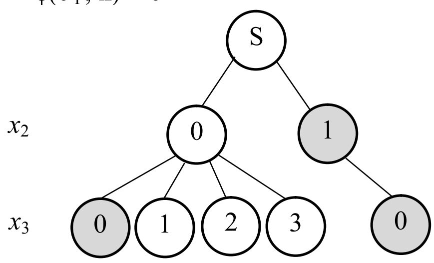

Fig. 2. The tree of paths for the MDD in Fig.1

For example, consider the tree of paths of the DPLD $\partial\phi(0\to1)/\partial x_1(0\to1)$ for the condition $\phi(0_1, \mathbf{x}) = 0$ (Fig.2). This tree starts from root labelled "S" that conforms to the variable x_1, because it is variable of the DPLD. The variable, on which the derivative is calculated, isn't included in the tree of paths. Consider other values of variables x_2 and x_3 between $x_1 = 0$ and $\phi(\mathbf{x}) = 0$ for the MDD in Fig.1. If the variable $x_2 = 0$, the variable x_3 is absent in this path. Therefore the variable x_3 can has any value. So, the tree of paths includes node with value 0 for the variable x_2 and nodes with values 0, 1, 2, 3 for the variable x_3 (it is "white part" of the tree in Fig.2). If the variable $x_2 = 1$, the variable $x_3 = 0$ (it is "gray part" of the tree in Fig.2). Therefore the tree of paths in Fig.2 locates all paths for condition $\phi(0_1, \mathbf{x}) = 0$.

2.3. New Algorithms for the DPLD calculation by the "tree of path"

Consider two algorithms for the DPLD (3) calculation by the MDD based on the application of the trees of paths below.

The Algorithm 1 has three steps. The tree of paths for the condition $\phi(s_i, \mathbf{x}) = j$ is formed at the first step. The tree of paths for the condition $\phi(\tilde{s}_i, \mathbf{x}) = \tilde{j}$ is obtained at the second step. The last step of the algorithm is comparing these two trees. The general part of these trees is decision that is non-zero values of the DPLD (3).

For example, Fig. 3 illustrates the calculation of the DPLD $\partial\phi(0\to1)/\partial x_1(0\to1)$ for MDD in Fig. 1 based on the Algorithm 1. The first step of this algorithm is forming of the tree of paths for condition $\phi(0_1, \mathbf{x}) = 0$. This tree building is considered in detail above (Fig. 2). The second step of the algorithm permits to obtain the tree of paths for condition $\phi(1_1, \mathbf{x}) = 1$. This tree includes all paths from the out-coming edge labelled 1 of the first variable x_1 to the sink node 1 of the MDD in Fig. 1. The third step of algorithm is comparing of two trees that satisfy condition $\phi(0_1, \mathbf{x}) = 0$ and $\phi(1_1, \mathbf{x}) = 1$ accordingly. The resultant tree of paths (the algorithm result) includes paths that are identical for two initial trees. So, the paths with $x_2 = 0$ and $x_3 = \{1, 2, 3\}$ are appended in the resultant tree, because they are in both trees. The path $x_2 = 0$ and $x_3 = 0$ is only in the first tree for the condition $\phi(0_1, \mathbf{x}) = 0$. The path $x_2 = 1$ and $x_3 = 0$ in this tree differ from the path $x_2 = 1$ and $x_3 = 1$ in the tree for the condition $\phi(1_1, \mathbf{x}) = 1$. Therefore the non-zero valued of the DPLD $\partial\phi(0\to1)/\partial x_1(0\to1)$ are $(x_1, x_2, x_3) = \{(\underline{0\to1}, 0, 1), (\underline{0\to1}, 0, 2), (\underline{0\to1}, 0, 3)\}$.

The Algorithm 2 is modification of the Algorithm 1 and includes 2 steps. The first step is identical to the first step of the algorithm 1. The result of this step is the tree of paths for the condition $\phi(s_i, \mathbf{x}) = j$. The second phase of this algorithm unites the second and third steps of the algorithm 1.

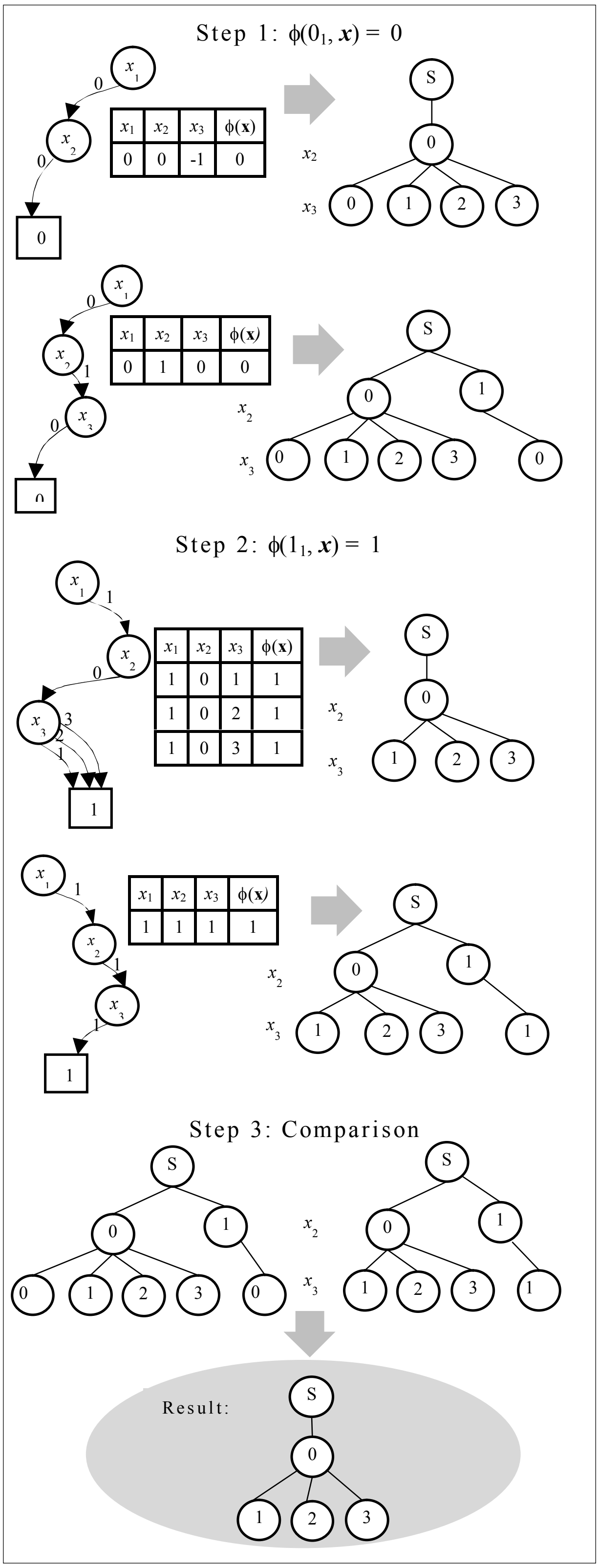

Fig. 3. Example of the Algorithm 1

The tree of paths for the condition $\phi(s_i, \mathbf{x}) = j$ is verified and transformed to the resultant tree of paths that is defined non-zero valued of the DPLD. The one of paths of the tree is selected and transformed taking into account next rules: (a) the value of the variable x_i is changed from s to $\tilde{s}$; (b) the value of the sink node is define as $\tilde{j}$. This transformed path is compared with the MDD. If this path is in the MDD, it is included into the resultant tree.

For example, in the Fig. 5 is presented the Algorithm 2 for calculation of the DPLD $\partial\phi(0\rightarrow1)/\partial x_1(0\rightarrow1)$ for the MDD (Fig. 1). The first step of this algorithm is identical of the first step of the Algorithm 2. The resultant tree is formed at the second step of the Algorithm 2. The building of this tree starts from analysis of the path $(x_1, x_2, x_3) = (1, 0, 0)$ (Fig. 4). This path doesn't agree with any of paths of the MDD from the top node to the sink node labelled 1. Therefore this path isn't included to the resultant tree. Next tree paths from the root to the sink node labelled 1 $(x_1, x_2, x_3) = (1, 0, 1)$, $(x_1, x_2, x_3) = (1, 0, 2)$ and $(x_1, x_2, x_3) = (1, 0, 3)$ are in the MDD. Therefore these paths are included in the resultant tree. And last path $(x_1, x_2, x_3) = (1, 1, 1)$ (Fig.4) isn't in the MDD. The result of this algorithm (Fig.4) is identical to the result of the Algorithm 1 (Fig. 3).

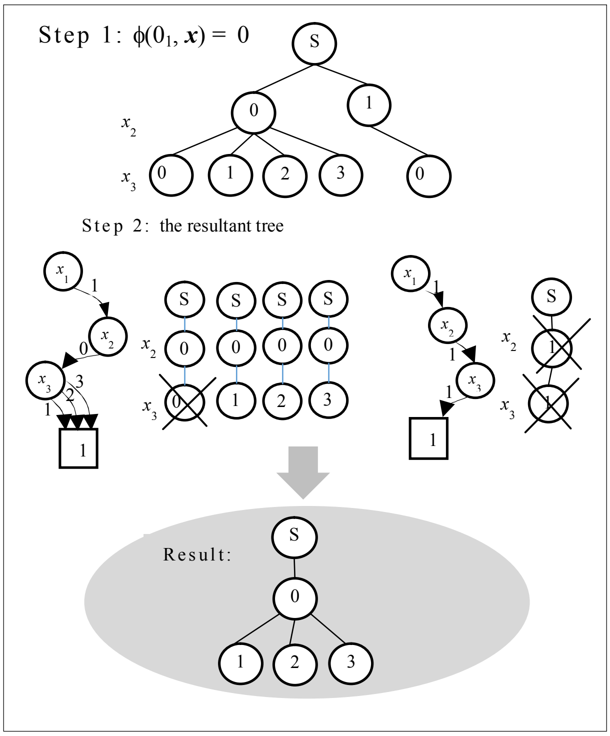

Fig. 4. Example of the Algorithm 2

These algorithms have similar computation complexity and the pseudo-code of these algorithms is in Fig. 5.

3 Experimental researches

The Algorithm 1 and the Algorithm 2 are tested by sets of the benchmark's LGSynth91. With tool ABC (A System for Sequential Synthesis and Verification developed by Berkeley Verification and Synthesis Research Center) [13].

```
Search_Diagram(state_1){
   Construct_TreeOfPaths_1();
}

if(algorithm_1) {
    Search_Diagram(state_2){
         Construct_TreeOfPaths_2();
    }

     Compare_Trees(TreeOfPaths_1,
    TreeOfPaths_2){
View_Result();
     }
}
else {   // algorithm 2
     Compare_Trees(TreeOfPaths_1,
    Diagram(state_2)){
    View_Result();
     }
}
```

Fig. 5. Pseudo-code of algorithms

These benchmarks are transformed to set on Decision Diagram. 18 Decision Diagrams are builder based on the benchmarks. One characteristic is considered in the testing and it is scanning duration in CPU ticks. The computer with Windows 7 Professional 64-bit with two cores CPU Intel i5-2430M 2.40 GHz was used for testing.

The comparison of both algorithms according to calculating time given the number of variables and numbers of nodes (ordered by number of nodes) is in Fig. 6. For both algorithms it is possible to see the relationship between the number of variables and number of nodes in the BDD benchmarks. Numbers at the left part of the graphs indicate the number of the CPU ticks and the numbers in the right are number of the nodes in the benchmarks.

The computation times of the Algorithm 1 and Algorithms 2 are indicated as "NEW1-2T TIME" and "NEW2-1T TIME" respectively

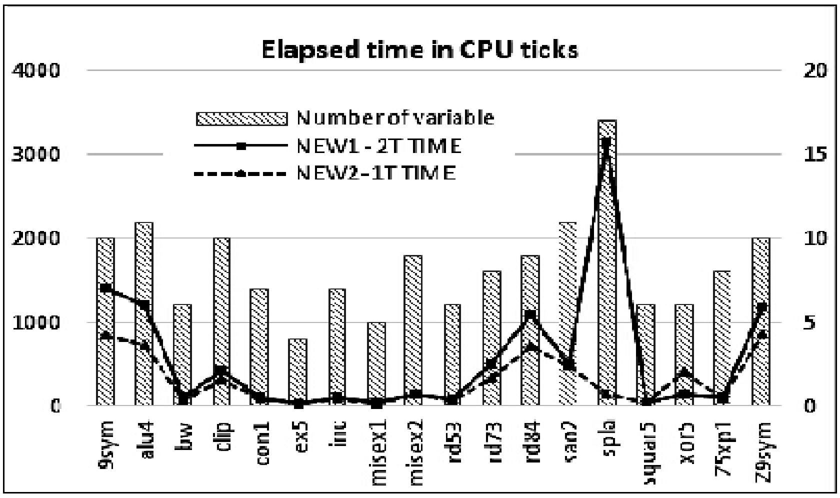

Fig. 6. Test 1 – elapsed time

The comparison of the computation time for the Algorithm1 and Algorithm 2 show that the Algorithm has lesser computational complexity.

Conclusion

MDD is well-suited for representation of MVL function with large number of variables. MSS structure function has a lot of variables that are agree with system components. Therefore a structure function has a large dimension and MDD is useful for representation of MSS structure function. But most of algorithms in MSS reliability analysis are proposed for system representation by Truth Table or equation, so new algorithms for MSS estimation based on MDD representation of system structure function development are necessary.

New algorithms for calculation of the DPLD by the MDD are presented in this paper. Therefore DPLD is mathematical background in algorithms for calculation of importance measures. The computational complexity of these algorithms is less than the previously proposed algorithms in [8].

References

1. Barlow, R. Importance of System Components and Fault Tree Events [Text] / R. Barlow, F. Proschan // Stochastic Processes and their Applications. – Vol. 3. – P. 153 – 173.

2. Meng, F.C. On Some Structural Importance of System Components [Text] / F.C. Meng // Journal of Data Science. – 2009. – Vol. 7. – P. 277 – 283.

3. Hwang, F.K. Anew index of component importance [Text] / F.K. Hwang // Operations Research Letters. – 2001. – 28. – P. 75 – 79.

4. Levitin, G. Generalised Importance Measures for Multi-State Elements Based on Performance Level Restric-

tions [Text] / G. Levitin et al. // Reliability Engineering and System Safety. – 2003. – Vol. 82(3). – P. 287 – 298.

5. Kuo, Way. Importance Measures in Reliability, Risk, and Optimization: Principles and Applications [Text] / Way Kuo, Xiaoyan Zhu // Wiley, 2012, – P. 472.

6. Zaitseva, E. Importance Analysis of a Multi-State System Based on Multiple-Valued Logic Methods [Text] / E. Zaitseva, A. Lisnianski, I. Frenkel // Recent Advances in System Reliability: Signatures, Multi-state Systems and Statistical Inference Springer: London, 2012. – P. 113 – 134.

7. Chang, Y.-R. Computing System Failure Frequencies and Reliability Importance Measures Using OBDD [Text] / Y.-R. Chang IEEE Transactions on Computers, Vol. 53(1), 2004. –, P. 54 – 68.

8. Multi-valued decision diagram in importance analysis [Text] / E. Zaitseva, V. Levashenko, J. Kostolny, V. Kharchenko // Radioelectronic and Computer systems: scientific and technical magazine. –2012. – Vol. 6(58). – P. 95 – 102.

9. Lisnianski, A. Multi-State System Reliability. Assessment, optimization and applications [Text] / A. Lisnianski, G. Levitin // World scientific, 2003.

10. Miller, M. On the construction of multiple-valued decision diagrams [Text] / M. Miller, R. Drechsler // Symp. on Multiple-Valued Logic, Calgary, Canada, 2002. – P. 264 – 269.

11. Sasao, T. Schwitching Theory for Logic Synthesis [Text] / T. Sasao // Kluwer Academic Publishers, Norwell, 1999.

12. Ramirez-Marquez, J.E. Composite Importance Measures for Multi-State Systems with Multi-State Components [Text] / J.E. Ramirez-Marquez, D.W. Coit // IEEE Trans. on Reliability. – 2005. – Vol. 54(3). – P. 517 – 529.

13. ABC: A System for Sequential Synthesis and Verification at Berkley University [Electronic resource]. – Available to: http://www.eecs.berkeley.edu /~alanmi/abc/ - 13.03.2013 y.

Поступила в редакцию 27.02.2013, рассмотрена на редколлегии 20.03.2013

Рецензент: д-р техн. наук, доцент, проф. каф. А.В. Горбенко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

ДІАГРАМИ РІШЕНЬ І ЛОГІЧНІ НАПРАВЛЕНІ ПОХІДНІ ДЛЯ АНАЛІЗУ НАДІЙНОСТІ СИСТЕМ З ДЕКІЛЬКОМА РІВНЯМИ НАДІЙНОСТІ

Й. Костолни, О. Зайцева

Логічні направлені похідні використовуються в аналізі надійності систем з декількома рівнями працездатності для обчислення оцінок значущості елементів. Системи з декількома рівнями працездатності є математичною моделлю, яка дозволяє описати декілька станів надійності системи. Проте така математична модель обумовлює використання спеціальних методів для аналізу, що обумовлене великою розмірністю обчислень. Один з можливих підходів до аналізу таких систем полягає в описі досліджуваної системи у вигляді діаграми рішень. У даній статті пропонуються нові алгоритми розрахунку логічних направлених похідних для обчислення індексів надійності систем з декількома рівнями працездатності заданих у вигляді багатозначних діаграм рішень.

Ключові слова: надійність, діаграми рішень, логічні направлені похідні, системи з декількома рівнями працездатності.

ДИАГРАММЫ РЕШЕНИЙ И ЛОГИЧЕСКИЕ НАПРАВЛЕННЫЕ ПРОИЗВОДНЫЕ ДЛЯ АНАЛИЗА НАДЕЖНОСТИ СИСТЕМ С НЕСКОЛЬКИМИ УРОВНЯМИ НАДЕЖНОСТИ

Й. Костолны, Е. Зайцева

Логические направленные производные используются в анализе надежности систем с несколькими уровнями работоспособности для вычисления оценок значимости элементов. Системы с несколькими уровнями работоспособности представляют собой математическую модель, которая позволяет описать несколько состояний надежности системы. Однако такая математическая модель обуславливает использование специальных методов для анализа, что обусловлено большой размерностью вычислений. Один из возможных подходов к анализу таких систем состоит в описании исследуемой системы в виде диаграммы решений. В данной статье предлагаются новые алгоритмы расчета логических направленных производных для вычисления индексов надежности систем с несколькими уровнями работоспособности заданных в виде многозначных диаграмм решений.

Ключевые слова: надежность, диаграммы решений, логические направленные производные, системы с несколькими уровнями работоспособности.

Костолны Йозеф – аспирант кафедры информатики Жилинского Университета, Жилина, Словакия, e-mail: jkostolny@gmail.com.

Зайцева Елена – канд. техн. наук, доцент кафедры информатики Жилинского Университета, Жилина, Словакия, e-mail: elena.zaitseva@fri.uniza.sk.