

DOI: <https://doi.org/10.15276/ict.03.2026.83>

UDC 004.05

Series-parallel composition of graph-logical models of fault-tolerant multiprocessor systems

Vitaliy A. Romankevich¹⁾ORCID: <https://orcid.org/0000-0003-4696-5935>; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058**Kostiantyn V. Morozov¹⁾**ORCID: <https://orcid.org/0000-0003-0978-6292>; mcng@ukr.net. Scopus Author ID: 57222509251**Alexei M. Romankevich¹⁾**ORCID: <https://orcid.org/0000-0001-5634-8469>; romankev@scs.kpi.ua. Scopus Author ID: 6602114176**Daniil V. Halytsky¹⁾**ORCID: <https://orcid.org/0009-0004-4421-3443>; zipper135401@gmail.com. Scopus Author ID: 58553487600**Lefteris Zacharioudakis²⁾**ORCID: <https://orcid.org/0000-0002-9658-3073>; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200¹⁾ National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”,
37, Peremogy Ave. Kyiv, 03056, Ukraine²⁾ Neapolis University Pafos, 2, Danais Ave. Pafos, 8042, Cyprus

ABSTRACT

The paper addresses the construction of composite graph-logical models describing the failure behavior of fault-tolerant multiprocessor systems. Such models can be used to evaluate system reliability measures through statistical experiments. Directly constructing a model of a complex system from its operability conditions can be labor-intensive; therefore, the initial problem may be decomposed into simpler constituents whose models are subsequently combined. The aim of the study is to establish relationships between the states of constituent two-terminal graphs and the state of the resulting graph under series and parallel composition. Three mutually exclusive classes are introduced to describe the state of a two-terminal graph: full two-terminal connectivity, separated two-terminal connectivity, and two-terminal disconnection. These classes are proved to cover all possible states of a two-terminal graph. Necessary and sufficient conditions are established for the resulting graph to belong to each class under series and parallel composition of its constituents. The resulting relationships make it possible to determine the state of a composite model from the states of its constituents and the method of their composition without reanalyzing the internal structure of each graph. Elementary single-edge graph-logical models are considered, and their suitability as basic constituents for constructing models with more complex behavior is demonstrated. The results are illustrated by constructing a composite graph-logical model of a fault-tolerant multiprocessor system that executes a task in stages and has alternative conditions for using processor resources. The model is formed by series-parallel composition of elementary single-edge models whose edge functions represent the conditions for completing individual stages. Exhaustive enumeration of the possible state vectors confirmed that connectivity of the resulting graph conforms to the specified system operability conditions. Future research should develop a formal algorithm for selecting constituent models and their composition directly from a system description, and should investigate the use of previously constructed multi-edge models as constituents.

Keywords: graph-logical models; fault-tolerant multiprocessor systems; failure behavior; two-terminal graphs; model composition; graph connectivity; series composition; parallel composition; edge functions

For citation: Romankevich V. A., Morozov K. V., Romankevich A. M., Halytsky D. V., Zacharioudakis Lefteris “Series-parallel composition of graph-logical models of fault-tolerant multiprocessor systems”. *Informatics. Culture. Technology*. 2026; Vol. 3 No. 1 (3): 1069–1082. DOI: <https://doi.org/10.15276/ict.03.2026.83>

Relevance. Modern technical systems are characterized by an ever-increasing level of automation. Assigning some control functions to automated equipment reduces the operator workload and the volume of routine operations, improves the accuracy and speed of task execution, and mitigates the effects of the human factor. A key component of such technical systems is the control system, which receives information from sensors, processes it according to specified algorithms, and generates control signals for actuators [1], [2]. Microprocessor and computing technologies are widely used to implement control systems.

Particularly stringent reliability requirements are imposed on critical application systems [3], [4]. Failure of such systems may cause substantial material losses, endanger human life and health, disrupt critical infrastructure, or lead to adverse environmental consequences. Because a control-system failure may cause the controlled object as a whole to operate incorrectly, ensuring control-system reliability and fault tolerance is a central task in the design of critical application systems.

Control systems subject to enhanced reliability requirements may be implemented using fault-tolerant multiprocessor systems (FTMSs) [5], [6], [7]. Such systems contain several processors and, in some cases, dozens or hundreds of them. Owing to redundant computing resources and reconfiguration mechanisms, they can remain operational after some processors have failed. The system state may depend not only on the number of failures but also on which particular processors have failed. Consequently, for the same number of failures, different combinations of failed processors may either preserve system operability or cause system failure.

An important stage in the design of a fault-tolerant multiprocessor system is the evaluation of its reliability measures [8], [9], [10]. The resulting estimates make it possible to verify compliance with the specified requirements, compare alternative structural solutions, and identify components whose failure has the greatest effect on system operability. For complex and heterogeneous FTMSs, this task may require considerable computational effort because a large number of possible combinations of processor states must be considered [11], [12].

The reliability measures of an FTMS may be evaluated using various methods [11], [12], [13], [14]. One approach is to conduct statistical experiments with a model of the system's failure behavior. Graph-logical models (GL-models), which combine the properties of undirected graphs and Boolean functions, may be used for this purpose [15].

A GL-model is based on an undirected graph in which each edge is associated with a Boolean edge function. The arguments of the edge functions are the components of the system state vector. Each component represents the state of a particular processor and takes the value 1 if the processor is operational and 0 if it has failed. For a given state vector, edges whose functions evaluate to zero are removed from the model graph. Connectivity of the resulting graph corresponds to an operational system state for the failure configuration under consideration, whereas loss of connectivity corresponds to transition to a failed state [15].

Given the probabilistic characteristics of processor failures, statistical experiments with a GL-model make it possible to estimate FTMS reliability measures. A complete analysis of all possible state vectors requires considering 2^n combinations for a system with n processors. As the system dimension increases, exhaustive enumeration rapidly becomes computationally infeasible. In such cases, estimation may be performed using a sample of randomly generated state vectors. The accuracy of the results depends on the sample size, the sampling procedure, and the statistical estimation methods applied.

Thus, GL-models are one means of representing FTMS failure behavior and may be used to evaluate reliability measures by means of statistical experiments. For the evaluation to be correct, the GL-model must accurately represent system operability for every possible processor state vector.

For a complex FTMS, directly constructing a GL-model that represents all specified operability conditions may be difficult. One way to simplify the problem is through decomposition: separate, simpler GL-models are first constructed using known methods and are then combined into a resulting model. The way in which the graphs of the constituent models are combined may substantially affect the failure behavior of the resulting model. It is therefore relevant to investigate different methods of combining GL-models and to establish the relationship between the states of the individual constituents and the state of the resulting model. Such relationships make it possible to determine the behavior of a composite GL-model from the states of its constituents and the method used to combine them, and to justify the structure selected for the resulting model in accordance with the specified system operability conditions [16].

Research aim. The aim of this study is to establish relationships between the states of the two-terminal graphs of constituent GL-models and the state of the resulting graph under their series and parallel composition, and to formulate rules for determining the state of a composite model without reanalyzing the internal structure of its constituents.

To achieve this aim, the following objectives must be addressed:

- to identify characteristics of the graph state of a constituent GL-model that are sufficient for subsequent analysis of its composition with other models;
- to investigate how different methods of composing the graphs of constituent models affect the state of the resulting model;
- to establish the conditions under which the resulting graph is connected or enters one of the possible disconnected states;
- to formulate rules for determining the state of the resulting model from the states of its constituents without reanalyzing their internal structure;
- to demonstrate the proposed approach by constructing a GL-model of a complex FTMS.

States of the two-terminal graph of a GL-model. Consider the graph of a GL-model for a fixed processor state vector $\mathbf{x}$. To simplify the notation, the dependence of the graph on $\mathbf{x}$ will hereafter be omitted, and the graph will be written as $G = (V, E)$, where V is the vertex set and E is the set of edges retained for the state vector under consideration.

Let two distinct vertices $a, b \in V$ be distinguished in the graph and referred to as its terminals, or terminal vertices. To describe graph composition below, terminal a will be called the left terminal and terminal b the right terminal. The remaining vertices will be called internal vertices, and their set will be denoted by $\hat{V} = V \setminus \{a, b\}$.

Thus, a two-terminal graph can be represented as $G = (\hat{V} \cup \{a, b\}, E)$. The notation $u \sim v$ will indicate that a path exists between vertices u and v in graph G , whereas $u \not\sim v$ will indicate that no such path exists.

For the subsequent analysis, we introduce three possible states of a two-terminal graph.

Full two-terminal connectivity. A graph has full two-terminal connectivity if its terminals are connected to each other and every internal vertex is connected to both terminals:

$$a \sim b \wedge \forall v \in \hat{V}: (v \sim a \wedge v \sim b).$$

In this case, all vertices of the graph belong to a single connected component. Therefore, for an undirected graph, full two-terminal connectivity is equivalent to ordinary graph connectivity.

Separated two-terminal connectivity. A graph has separated two-terminal connectivity if its terminals are not connected to each other but every internal vertex is connected to at least one of them:

$$a \not\sim b \wedge \forall v \in \hat{V}: (v \sim a \vee v \sim b).$$

Because $a \not\sim b$, no internal vertex can be connected to both terminals simultaneously. Otherwise, combining the paths from that vertex to a and to b would produce a path between the terminals. Therefore, every vertex $v \in \hat{V}$ belongs to exactly one of two connected components: the component containing a or the component containing b .

Thus, in this state the graph consists of exactly two connected components, each containing one terminal.

Two-terminal disconnection. A graph has two-terminal disconnection if at least one internal vertex is connected to neither terminal:

$$\exists v \in \hat{V}: v \not\sim a \wedge v \not\sim b.$$

In this case, the graph contains at least one connected component containing neither terminal a nor terminal b . The number of other components and the connectivity between the terminals may be arbitrary. In particular, a path may exist between terminals a and b while the graph remains disconnected because it contains an isolated internal component.

Hereafter, $\mathcal{F}$, $\mathcal{S}$, and $\mathcal{D}$ will denote, respectively, the classes of two-terminal graphs with full two-terminal connectivity, separated two-terminal connectivity, and two-terminal disconnection.

Proposition. Classes $\mathcal{F}$, $\mathcal{S}$, and $\mathcal{D}$ are pairwise disjoint and collectively cover all possible states of a two-terminal graph.

Indeed, if graph G does not belong to $\mathcal{D}$, each of its internal vertices is connected to at least one terminal. If, in addition, $a \sim b$, every internal vertex belongs to the same connected component as both terminals. Hence, all graph vertices belong to a single component and $G \in \mathcal{F}$.

If $a \not\sim b$, the components containing terminals a and b are distinct. Because every internal vertex is connected to at least one terminal, all internal vertices are distributed between these two components. Hence, $G \in \mathcal{S}$.

Thus, $\mathcal{F} \cap \mathcal{S} = \mathcal{F} \cap \mathcal{D} = \mathcal{S} \cap \mathcal{D} = \emptyset$, and $G \in \mathcal{F} \cup \mathcal{S} \cup \mathcal{D}$ for every two-terminal graph.

Therefore, classes $\mathcal{F}$, $\mathcal{S}$, and $\mathcal{D}$ form a complete classification of the states of a two-terminal graph.

Membership of a graph in class $\mathcal{F}$, $\mathcal{S}$, or $\mathcal{D}$ is determined for a particular processor state vector. Consequently, the graph of the same GL-model may belong to different classes for different state vectors. Hereafter, this classification will be used to determine the state of the resulting graph from the states of its constituents and the method used to compose them.

Formalization of the problem of composing two-terminal graphs. Let $N \geq 2$ GL-models $M_1, M_2, \dots, M_N$ be given. For a fixed state vector $\mathbf{x}$, model M_i corresponds to graph $G_i = (V_i, E_i)$, $i = 1, 2, \dots, N$, in which the values of the edge functions have already been taken into account and all edges whose functions evaluated to zero have been removed.

In each graph G_i , fix two distinct terminal vertices a_i and b_i , referred to as its left and right terminals, respectively. Denote the set of internal vertices by $\hat{V}_i = V_i \setminus \{a_i, b_i\}$. Thus, $G_i = (\hat{V}_i \cup \{a_i, b_i\}, E_i)$.

Before composition, assume that the vertex sets of the constituent graphs are pairwise disjoint: $V_i \cap V_j = \emptyset, i \neq j$. According to the classification introduced above, every graph G_i belongs to exactly one of the classes $\mathcal{F}$, $\mathcal{S}$, or $\mathcal{D}$.

We next consider different ways of composing graphs $G_1, G_2, \dots, G_N$ into a resulting graph G . The composition is performed by identifying the corresponding terminals of the constituent graphs. Their internal vertices, edges, and edge functions remain unchanged.

The objective of the subsequent analysis is to establish necessary and sufficient conditions for the resulting graph to belong to classes $\mathcal{F}$, $\mathcal{S}$, and $\mathcal{D}$, depending on the states of the constituent graphs and whether they are composed in series or in parallel.

Series composition of graphs. Consider the series composition of two-terminal graphs $G_1, G_2, \dots, G_N$ (Fig. 1). In this composition, the right terminal of every graph except the last is identified with the left terminal of the next graph: $b_i \equiv a_{i+1}, i = 1, 2, \dots, N-1$.

The result is a two-terminal graph $G = (\hat{V} \cup \{a, b\}, E)$, whose left and right terminals are $a = a_1$ and $b = b_N$, respectively.

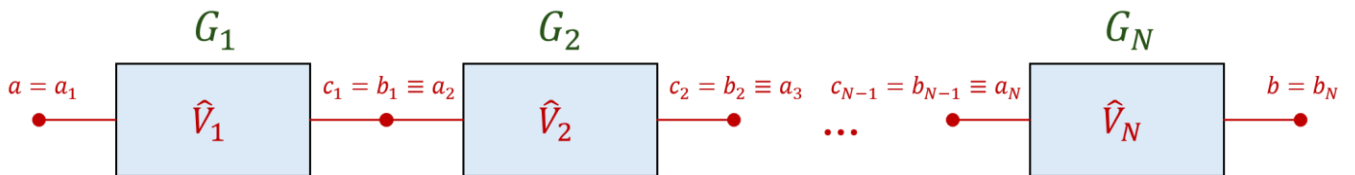


Fig. 1. Scheme of the series composition of two-terminal graphs $G_1, G_2, \dots, G_N$

Source: compiled by the authors

Let c_i denote the vertex formed by identifying terminals b_i and a_{i+1} : $c_i = b_i \equiv a_{i+1}$, $i = 1, 2, \dots, N-1$. The set of internal vertices of the resulting graph is then

$$\hat{V} = \bigcup_{i=1}^N \hat{V}_i \cup \bigcup_{j=1}^{N-1} \{c_j\}.$$

The edge set of the resulting graph is the union of the edge sets of all constituent graphs:

$$E = \bigcup_{i=1}^N E_i.$$

Theorem 1. A graph G obtained by the series composition of graphs $G_1, G_2, \dots, G_N$ has full two-terminal connectivity if and only if every constituent graph has full two-terminal connectivity:

$$G \in \mathcal{F} \Leftrightarrow \forall i \in \{1, 2, \dots, N\}: G_i \in \mathcal{F}.$$

Proof. We first prove sufficiency. Let $G_i \in \mathcal{F}$, $i = 1, 2, \dots, N$. In each graph G_i , its terminals a_i and b_i and all internal vertices belong to a single connected component. As a result of successively identifying vertices $b_i \equiv a_{i+1}$, $i = 1, 2, \dots, N-1$, the connected components of adjacent graphs are joined. Because this identification is performed for every pair of adjacent graphs, all vertices of the resulting graph, including its external terminals a and b , belong to a single connected component. Therefore, $G \in \mathcal{F}$.

We prove necessity by contraposition. Suppose there exists an index $k \in \{1, 2, \dots, N\}$ such that $G_k \notin \mathcal{F}$. Because classes $\mathcal{F}$, $\mathcal{S}$, and $\mathcal{D}$ form a complete classification of two-terminal graph states, two cases are possible.

If $G_k \in \mathcal{S}$, its terminals a_k and b_k belong to different connected components. The part of the series composition to the left of G_k is attached to it only through a_k , whereas the part to its right is attached only through b_k . Consequently, composition with the other constituent graphs cannot create a path bypassing G_k between these components. Thus, the resulting graph is disconnected and $G \notin \mathcal{F}$.

If $G_k \in \mathcal{D}$, it contains a connected component containing neither a_k nor b_k . Because G_k is connected to the other graphs exclusively through its terminals, this component remains isolated after series composition. The resulting graph is therefore also disconnected and $G \notin \mathcal{F}$.

Thus, $G \in \mathcal{F}$ is possible only if $G_i \in \mathcal{F}$ for every $i = 1, 2, \dots, N$. ■

Theorem 2. A graph G obtained by series composition has separated two-terminal connectivity if and only if exactly one constituent graph belongs to class $\mathcal{S}$ and all the others belong to class $\mathcal{F}$:

$$G \in \mathcal{S} \Leftrightarrow (\exists! i \in \{1, 2, \dots, N\}: G_i \in \mathcal{S}) \wedge (\forall j \in \{1, 2, \dots, N\} \setminus \{i\}: G_j \in \mathcal{F}).$$

Proof. We first prove sufficiency. Suppose that, for some index $k \in \{1, 2, \dots, N\}$, $G_k \in \mathcal{S}$, and that all other constituent graphs belong to $\mathcal{F}$. Because $G_k \in \mathcal{S}$, it has exactly two connected components. Let C_a denote the component containing terminal a_k , and let C_b denote the component containing terminal b_k .

All graphs to the left of G_k , if any, belong to $\mathcal{F}$. After series composition, all their vertices join component C_a . Therefore, this component contains external terminal $a = a_1$ of the resulting graph.

Similarly, all graphs to the right of G_k belong to $\mathcal{F}$, and their vertices join component C_b . Hence, this component contains external terminal $b = b_N$.

Because graph G_k contains no path between a_k and b_k , components C_a and C_b remain distinct after series composition. There are no other components because all other constituent graphs belong to $\mathcal{F}$. Thus, the resulting graph has exactly two connected components, each containing one of its external terminals. Therefore, $G \in \mathcal{S}$.

We now prove necessity. If the condition of the theorem is not satisfied, the following cases are possible.

Case 1. All constituent graphs belong to $\mathcal{F}$. By Theorem 1, $G \in \mathcal{F}$ in this case, and therefore $G \notin \mathcal{S}$.

Case 2. At least one constituent graph $G_i \in \mathcal{D}$ exists. Such a graph contains a connected component containing neither a_i nor b_i . Because it is connected to the other graphs only through these terminals, that component also remains isolated in the resulting graph. It contains neither external terminal a nor b ; therefore, $G \in \mathcal{D}$.

Case 3. In the absence of constituents of class $\mathcal{D}$, at least two constituent graphs belong to $\mathcal{S}$. Select two consecutive such graphs G_i and G_j , where $i < j$. Then all graphs $G_{i+1}, G_{i+2}, \dots, G_{j-1}$, if any, belong to $\mathcal{F}$.

After composition, the right-hand component of graph G_i , the intermediate graphs $G_{i+1}, G_{i+2}, \dots, G_{j-1}$, and the left-hand component of graph G_j form a component of the resulting graph that contains neither external terminal $a = a_1$ nor external terminal $b = b_N$. If $j = i + 1$, this component contains at least the common vertex $b_i \equiv a_{i+1}$. Therefore, the resulting graph belongs to class $\mathcal{D}$ rather than $\mathcal{S}$.

Thus, the resulting graph belongs to $\mathcal{S}$ only when exactly one constituent graph belongs to class $\mathcal{S}$ and all other constituents belong to $\mathcal{F}$. ■

It follows from Theorems 1 and 2 and the completeness of the classification $\mathcal{F}, \mathcal{S}, \mathcal{D}$ that the resulting graph has two-terminal disconnection if and only if at least one constituent graph belongs to $\mathcal{D}$ or at least two constituent graphs belong to $\mathcal{S}$:

$$G \in \mathcal{D} \Leftrightarrow (\exists i: G_i \in \mathcal{D}) \vee (|\{j: G_j \in \mathcal{S}\}| \geq 2).$$

Parallel composition of graphs. Consider the parallel composition of the two-terminal graphs $G_1, G_2, \dots, G_N$ (Fig. 2). In a parallel composition, all left terminals of the constituent graphs are identified with one another and form the left terminal a of the resulting graph. Similarly, all right terminals form its right terminal b :

$$\begin{aligned} a_1 &\equiv a_2 \equiv \dots \equiv a_N = a, \\ b_1 &\equiv b_2 \equiv \dots \equiv b_N = b. \end{aligned}$$

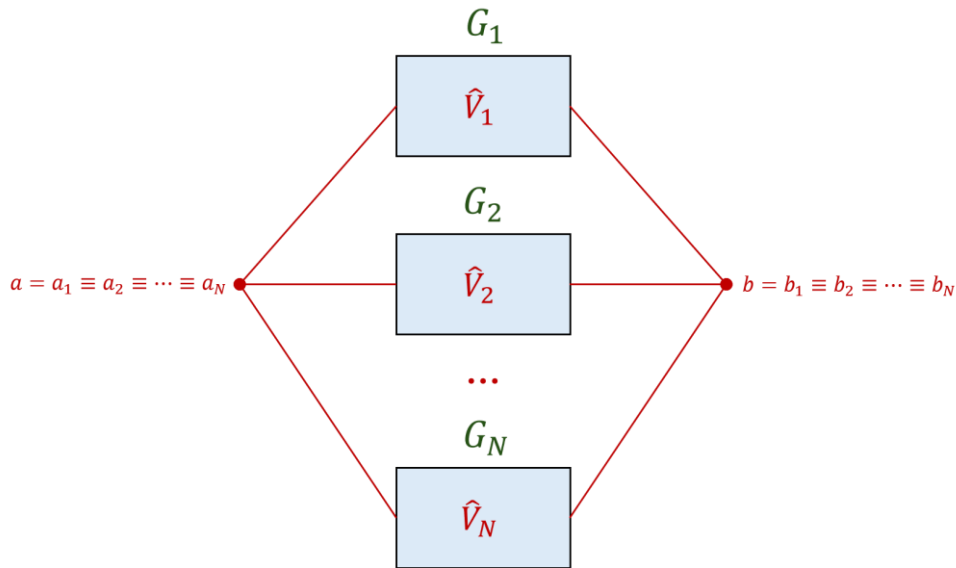


Fig. 2. Scheme of the parallel composition of the two-terminal graphs $G_1, G_2, \dots, G_N$

Source: compiled by the authors

The composition produces the two-terminal graph $G = (\hat{V} \cup \{a, b\}, E)$, whose set of internal vertices is the union of the sets of internal vertices of the constituent graphs:

$$\hat{V} = \bigcup_{i=1}^N \hat{V}_i,$$

and its edge set is defined as

$$E = \bigcup_{i=1}^N E_i.$$

Theorem 3. A graph G obtained by the parallel composition of graphs $G_1, G_2, \dots, G_N$ has separated two-terminal connectivity if and only if every constituent graph has separated two-terminal connectivity:

$$G \in \mathcal{S} \Leftrightarrow \forall i \in \{1, 2, \dots, N\}: G_i \in \mathcal{S}.$$

Proof. We first prove sufficiency. Let $G_i \in \mathcal{S}$, $i = 1, 2, \dots, N$. Each graph G_i has exactly two connected components: one contains its left terminal a_i , and the other contains its right terminal b_i . Every internal vertex of the graph belongs to one of these components.

After parallel composition, all components containing the terminals a_i merge into a single component C_a containing terminal a . Similarly, the components containing the terminals b_i merge into a component C_b containing terminal b .

No constituent graph contains a path between its own terminals a_i and b_i . Before composition, the sets of internal vertices of the constituent graphs are pairwise disjoint, and after composition only the terminals a and b are shared. Therefore, any path from a to b would have to contain a path between the two terminals of at least one constituent graph, which is impossible. Hence, $a \not\sim b$.

Moreover, every internal vertex of the resulting graph belongs to either C_a or C_b . Thus, graph G has exactly two connected components, each containing one of its terminals. Therefore, $G \in \mathcal{S}$.

We now prove necessity. Suppose that at least one constituent graph does not belong to $\mathcal{S}$.

If there exists a graph $G_i \in \mathcal{D}$, it contains a connected component that includes neither a_i nor b_i . Since G_i is joined to the other constituent graphs only through its terminals, this component remains isolated after parallel composition. It contains neither a nor b ; therefore, the resulting graph belongs to $\mathcal{D}$.

If no constituent graph belongs to $\mathcal{D}$, then all constituent graphs belong to $\mathcal{F}$ or $\mathcal{S}$. Because not all constituent graphs belong to $\mathcal{S}$, there exists at least one graph $G_i \in \mathcal{F}$. This graph contains a path between a_i and b_i , which becomes a path between a and b after the terminals are identified. Every internal vertex of any other constituent graph belonging to $\mathcal{F}$ or $\mathcal{S}$ is connected to at least one terminal. Since $a \sim b$, all such vertices belong to a single connected component. Therefore, $G \in \mathcal{F}$ in this case.

Thus, the resulting graph can belong to $\mathcal{S}$ only if all constituent graphs belong to $\mathcal{S}$. ■

Theorem 4. A graph G obtained by parallel composition has full two-terminal connectivity if and only if at least one constituent graph has full two-terminal connectivity and no constituent graph has two-terminal disconnection:

$$G \in \mathcal{F} \Leftrightarrow (\exists i \in \{1, 2, \dots, N\}: G_i \in \mathcal{F}) \wedge (\forall j \in \{1, 2, \dots, N\}: G_j \in \mathcal{F} \cup \mathcal{S}).$$

Equivalently, the second condition can be written as

$$\nexists j \in \{1, 2, \dots, N\}: G_j \in \mathcal{D}.$$

Proof. We first prove sufficiency. Suppose that at least one constituent graph G_i belongs to $\mathcal{F}$ and that all constituent graphs belong to $\mathcal{F}$ or $\mathcal{S}$.

Because $G_i \in \mathcal{F}$, it contains a path between its terminals a_i and b_i . After parallel composition, this path becomes a path between the external terminals a and b . Hence, $a \sim b$.

In every constituent graph belonging to $\mathcal{F}$ or $\mathcal{S}$, each internal vertex is connected to at least one of its terminals. After the terminals are identified, every internal vertex of the resulting graph is connected to at least one of a and b . Since $a \sim b$, all internal vertices and both terminals belong to a single connected component. Therefore, $G \in \mathcal{F}$.

We now prove necessity. Let $G \in \mathcal{F}$.

First, no constituent graph can belong to $\mathcal{D}$. Indeed, if, for some index i , condition $G_i \in \mathcal{D}$ holds, then G_i contains a connected component that includes neither a_i nor b_i . Because the other graphs are joined to G_i only through its terminals, this component would remain isolated in the resulting graph. This contradicts the assumption $G \in \mathcal{F}$.

Therefore, $\forall j \in \{1, 2, \dots, N\}: G_j \in \mathcal{F} \cup \mathcal{S}$. In addition, at least one constituent graph must belong to $\mathcal{F}$. If all constituent graphs belonged to $\mathcal{S}$, then, by Theorem 3, the resulting graph would also belong to $\mathcal{S}$, contradicting the assumption $G \in \mathcal{F}$.

Thus, at least one constituent graph belongs to $\mathcal{F}$, and none belongs to $\mathcal{D}$. ■

Theorems 3 and 4, together with the completeness of the classification $\mathcal{F}$, $\mathcal{S}$, $\mathcal{D}$, imply that a graph obtained by parallel composition has two-terminal disconnection if and only if at least one constituent graph belongs to $\mathcal{D}$:

$$G \in \mathcal{D} \Leftrightarrow \exists i \in \{1, 2, \dots, N\}: G_i \in \mathcal{D}.$$

Elementary single-edge graphs. For the subsequent analysis and synthesis of GL-models, it is useful to consider the simplest nontrivial two-terminal graph that can serve as the basis of such a model.

A single-vertex graph does not allow two distinct terminal vertices to be selected and cannot change its connectivity state. A graph with two or more vertices but no edges remains disconnected for every state vector. Therefore, the simplest graph whose connectivity can vary with the value of an edge function consists of two vertices joined by one edge.

Let the initial graph of the model be $G = (V, E)$, $V = \{a, b\}$, $E = \{e\}$, where a and b are terminal vertices and e is the edge between them. Since the graph contains no other vertices, its set of internal vertices is empty: $\hat{V} = V \setminus \{a, b\} = \emptyset$.

Edge e is associated with the edge function $f_e(\mathbf{x})$, whose value determines whether this edge is retained for a given state vector $\mathbf{x}$. If $f_e(\mathbf{x}) = 1$, edge e is retained, and vertices a and b belong to the same connected component. Therefore, the resulting graph has full two-terminal connectivity: $G \in \mathcal{F}$.

If $f_e(\mathbf{x}) = 0$, edge e is removed, and the graph consists of two isolated terminal vertices. In this case, $a \sim b$, and there are no internal vertices. Therefore, the graph has separated two-terminal connectivity: $G \in \mathcal{S}$.

Two-terminal disconnection is impossible for this graph because it has no internal vertices and therefore cannot contain a connected component that includes neither terminal. Hence, $G \notin \mathcal{D}$ for either value of the edge function.

Such a single-edge GL-model represents a specified Boolean condition through the presence or absence of a connection between its terminals and can be used as an elementary component in the synthesis of composite GL-models.

Example. Consider a fault-tolerant multiprocessor system that performs a task in three consecutive stages. The system comprises 13 processors $P_1, P_2, \dots, P_{13}$, divided into five functional groups: $\mathcal{P}_1 = \{P_1, P_2\}$, $\mathcal{P}_2 = \{P_3, P_4\}$, $\mathcal{P}_3 = \{P_5, P_6, P_7\}$, $\mathcal{P}_4 = \{P_8, P_9, P_{10}\}$, $\mathcal{P}_5 = \{P_{11}, P_{12}, P_{13}\}$.

Processors within each group are interchangeable. Processors P_8 and P_{11} additionally support a special interaction mode between the fourth and fifth groups. When both processors are operational, the third stage can be performed with fewer operational processors.

The first stage can be performed if either of the following conditions holds:

- both processors in group $\mathcal{P}_1$ and at least one processor in group $\mathcal{P}_2$ are operational;
- at least one processor in group $\mathcal{P}_1$ and both processors in group $\mathcal{P}_2$ are operational.

The second stage requires at least one operational processor in group $\mathcal{P}_3$.

The third stage can be performed if any one of the following three conditions holds:

- at least two processors in group $\mathcal{P}_4$ and all three processors in group $\mathcal{P}_5$ are operational;
- all three processors in group $\mathcal{P}_4$ and at least two processors in group $\mathcal{P}_5$ are operational;
- processors P_8 and P_{11} are operational, and at least two processors in each of groups $\mathcal{P}_4$ and $\mathcal{P}_5$ are operational.

In the latter case, P_8 and P_{11} count toward the two required processors in their respective groups.

Let $\mathbf{x} = (x_1, x_2, \dots, x_{13})$, where $x_i = 1$ if processor P_i is operational and $x_i = 0$ if it has failed.

We construct twelve elementary single-edge GL-models $M_1, M_2, \dots, M_{12}$. The graph of model M_i is denoted by G_i , and its single edge function by $f_i(\mathbf{x})$.

The edge functions of the models are defined as follows:

$$\begin{array}{lll} f_1 = x_1 x_2; & f_5 = x_5 \vee x_6 \vee x_7; & f_9 = x_{11} x_{12} \vee x_{13} (x_{11} \vee x_{12}); \\ f_2 = x_3 \vee x_4; & f_6 = x_8 x_9 \vee x_{10} (x_8 \vee x_9); & f_{10} = x_8 x_{11}; \\ f_3 = x_1 \vee x_2; & f_7 = x_{11} x_{12} x_{13}; & f_{11} = x_9 \vee x_{10}; \\ f_4 = x_3 x_4; & f_8 = x_8 x_9 x_{10}; & f_{12} = x_{12} \vee x_{13}. \end{array}$$

Since graph G_i has a single edge,

$$G_i \in \mathcal{F} \Leftrightarrow f_i(\mathbf{x}) = 1, \quad G_i \in \mathcal{S} \Leftrightarrow f_i(\mathbf{x}) = 0, \quad G_i \notin \mathcal{D}$$

The model of the first stage is formed by two parallel branches:

$$H_1 = (G_1 \circ G_2) \parallel (G_3 \circ G_4),$$

where $\circ$ and $\parallel$ denote series and parallel composition of graphs, respectively. The first branch corresponds to both processors in the first group and at least one processor in the second group being operational, whereas the second branch corresponds to at least one processor in the first group and both processors in the second group being operational. Therefore,

$$H_1 \in \mathcal{F} \Leftrightarrow f_1(\mathbf{x}) f_2(\mathbf{x}) \vee f_3(\mathbf{x}) f_4(\mathbf{x}) = 1.$$

In this case, the full two-terminal connectivity of one branch precludes the other branch from belonging to class $\mathcal{D}$. Indeed, $f_1(\mathbf{x}) = 1 \Rightarrow f_3(\mathbf{x}) = 1$ and $f_4(\mathbf{x}) = 1 \Rightarrow f_2(\mathbf{x}) = 1$. Hence, if one branch belongs to $\mathcal{F}$, the other can belong only to $\mathcal{F}$ or $\mathcal{S}$.

The model of the second stage is a single-edge GL-model: $H_2 = G_5$. Accordingly,

$$H_2 \in \mathcal{F} \Leftrightarrow f_5(\mathbf{x}) = 1 \Leftrightarrow x_5 \vee x_6 \vee x_7 = 1,$$

which corresponds to at least one processor in the third group being operational.

The model of the third stage is formed by three parallel branches:

$$H_3 = (G_6 \circ G_7) \parallel (G_8 \circ G_9) \parallel (G_{10} \circ G_{11} \circ G_{12}).$$

The first branch corresponds to at least two processors in the fourth group and all three processors in the fifth group being operational. The second branch corresponds to all processors in the fourth group and at least two processors in the fifth group being operational. The third branch represents the special mode. Its condition for full two-terminal connectivity is

$$f_{10}(\mathbf{x}) f_{11}(\mathbf{x}) f_{12}(\mathbf{x}) = x_8 x_{11} (x_9 \vee x_{10}) (x_{12} \vee x_{13}) = 1.$$

Thus, the third branch has full connectivity if processors P_8 and P_{11} , at least one of processors P_9 and P_{10} , and at least one of processors P_{12} and P_{13} are operational.

For graph H_3 , we obtain

$$H_3 \in \mathcal{F} \Leftrightarrow f_6(\mathbf{x}) f_7(\mathbf{x}) \vee f_8(\mathbf{x}) f_9(\mathbf{x}) \vee f_{10}(\mathbf{x}) f_{11}(\mathbf{x}) f_{12}(\mathbf{x}) = 1.$$

The validity of this parallel composition with respect to class $\mathcal{D}$ follows from the relations

$$\begin{array}{l} f_8(\mathbf{x}) = 1 \Rightarrow f_6(\mathbf{x}) = f_{11}(\mathbf{x}) = 1, \\ f_7(\mathbf{x}) = 1 \Rightarrow f_9(\mathbf{x}) = f_{12}(\mathbf{x}) = 1, \\ f_{10}(\mathbf{x}) f_{11}(\mathbf{x}) = 1 \Rightarrow f_6(\mathbf{x}) = 1, \\ f_{10}(\mathbf{x}) f_{12}(\mathbf{x}) = 1 \Rightarrow f_9(\mathbf{x}) = 1. \end{array}$$

Therefore, the full two-terminal connectivity of any of the three branches precludes the other branches from belonging to class $\mathcal{D}$.

Because all three stages of the task must be completed, the resulting GL-model is formed by composing their models in series:

$$G = H_1 \circ H_2 \circ H_3.$$

Substituting the constituent models gives

$$G = ((G_1 \circ G_2) \parallel (G_3 \circ G_4)) \circ H_2 \circ ((G_6 \circ G_7) \parallel (G_8 \circ G_9) \parallel (G_{10} \circ G_{11} \circ G_{12})).$$

According to the properties of series composition,

$$G \in \mathcal{F} \Leftrightarrow (H_1 \in \mathcal{F}) \wedge (H_2 \in \mathcal{F}) \wedge (H_3 \in \mathcal{F}).$$

Thus, the full two-terminal connectivity of the resulting graph corresponds exactly to the conditions under which all three stages of the task can be completed. The GL-model of the system based on graph G is shown in Fig. 3.

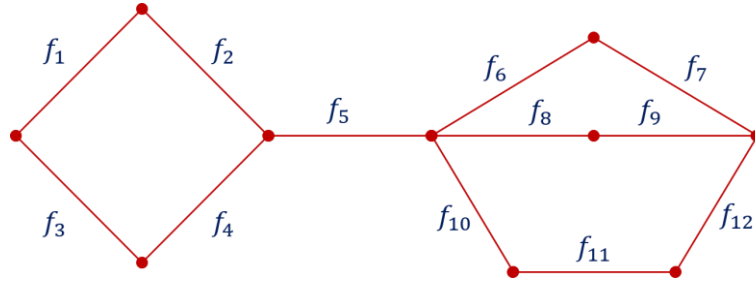


Fig. 3. GL-model of the system considered in the example

Source: compiled by the authors

For example, for the state vector $\mathbf{x}^{(1)} = 1110010110110$, the first stage is performed by both processors in the first group and one processor in the second group, whereas the second stage is performed by processor P_6 . Two processors are operational in each of the fourth and fifth groups, including P_8 and P_{11} . We have $f_1(\mathbf{x}^{(1)}) = f_2(\mathbf{x}^{(1)}) = f_5(\mathbf{x}^{(1)}) = f_{10}(\mathbf{x}^{(1)}) = f_{11}(\mathbf{x}^{(1)}) = f_{12}(\mathbf{x}^{(1)}) = 1$. Therefore, $H_1, H_2, H_3 \in \mathcal{F}$ and, consequently, $G \in \mathcal{F}$. In this case, the third branch of graph H_3 enables the third stage.

Consider also the state vector $\mathbf{x}^{(2)} = 0111100011111$. The first stage is performed by processor P_2 together with processors P_3 and P_4 , the second stage is performed by processor P_5 , and the third stage is performed by processors P_9 and P_{10} together with all processors in the fifth group. In this case, $f_3(\mathbf{x}^{(2)}) = f_4(\mathbf{x}^{(2)}) = f_5(\mathbf{x}^{(2)}) = f_6(\mathbf{x}^{(2)}) = f_7(\mathbf{x}^{(2)}) = 1$. Hence, $H_1, H_2, H_3 \in \mathcal{F}$ and therefore $G \in \mathcal{F}$. In this case, the first branch of graph H_3 enables the third stage.

For the vector $\mathbf{x}^{(3)} = 1110010111011$, the first stage is performed by processors P_1 , P_2 , and P_3 , the second stage is performed by processor P_6 , and the third stage is performed by all processors in the fourth group together with processors P_{12} and P_{13} . Since $f_1(\mathbf{x}^{(3)}) = f_2(\mathbf{x}^{(3)}) = f_5(\mathbf{x}^{(3)}) = f_8(\mathbf{x}^{(3)}) = f_9(\mathbf{x}^{(3)}) = 1$, we have $H_1, H_2, H_3 \in \mathcal{F}$ and $G \in \mathcal{F}$. In this case, the second branch of graph H_3 enables the third stage.

Consider the vector $\mathbf{x}^{(4)} = 0111100001111$. The first two stages can be completed, but only processor P_{10} remains operational in the fourth group. For the three branches of graph H_3 , respectively, we have $(f_6, f_7) = (0, 1)$, $(f_8, f_9) = (0, 1)$, and $(f_{10}, f_{11}, f_{12}) = (0, 1, 1)$. Each of these series branches belongs to class $\mathcal{S}$. Therefore, $H_3 \in \mathcal{S}$. Since $H_1, H_2 \in \mathcal{F}$, the resulting graph also belongs to class $\mathcal{S}$: $G \in \mathcal{S}$. The graph thus remains disconnected, which corresponds to an inoperable state of the system.

Finally, consider the vector $\mathbf{x}^{(5)} = 0111100001011$, which differs from $\mathbf{x}^{(4)}$ by the failure of processor P_{11} . For the first branch of graph H_3 , we have $f_6(\mathbf{x}^{(5)}) = f_7(\mathbf{x}^{(5)}) = 0$. Hence, this series branch belongs to class $\mathcal{D}$. According to the rule for parallel composition, the presence of at least one constituent of class $\mathcal{D}$ implies $H_3 \in \mathcal{D}$ and therefore $G \in \mathcal{D}$.

Exhaustive enumeration of all $2^{13} = 8192$ state vectors confirmed that the connectivity of the constructed GL-model corresponds to the specified conditions for task completion.

Conclusions. The paper considers an approach to constructing composite GL-models of the failure behavior of fault-tolerant multiprocessor systems. For two-terminal graphs, the classes of full two-terminal connectivity, separated two-terminal connectivity, and two-terminal disconnection

are introduced. Conditions under which the resulting graph belongs to these classes are established for series and parallel composition of constituent graphs. The resulting relations make it possible to determine the state of a composite model from the states of its constituents and the way in which they are composed, without reanalyzing the internal structure of each graph.

The use of elementary single-edge GL-models as constituents for synthesizing models with more complex behavior is examined. A system with 13 processors is used to demonstrate the construction of the resulting GL-model by series-parallel composition of twelve single-edge models. Analysis of representative state vectors and exhaustive enumeration of all $2^{13} = 8192$ possible states confirmed that the resulting graph is connected if and only if the system can complete all three stages of the task.

A promising direction for further research is to formalize the transition from a description of the system and its operability conditions to a set of edge functions and a scheme for composing them. The paper establishes relations between the states of the constituent and resulting models; however, it does not propose a formal algorithm for selecting constituent models or their method of composition. The results provide a basis for developing such an algorithm. It is also advisable to investigate the use of GL-models constructed by existing methods, rather than only elementary single-edge models, as constituents.

Conflict of interests: Authors declare that they do not have conflict of interests, in particular financial, personal, copyright or any of a different nature, which could have influenced the research, as well as the results published in this articles

Financing: Research was carried out without financial support

Accessibility data: All data available in digital or graphic form in the main text manuscript

Using artificial intelligence tools: During preparation manuscript used tool GPT-5.6 Sol of artificial intelligence (AI) for grammatical and stylistic text checks. All fragments processed by him were checked and edited by the author. AI was not used to generate text, formulas, and tables, formation scientific provisions, receipt results or formulation conclusions. Scientific the provisions, results and conclusions are my own, intellectual by the contribution of the authors

REFERENCES

1. Kotov, D. O. “A generalized model of an adaptive information-control system of a car with multi-sensor channels of information interaction”. *Applied Aspects of Information Technology*. 2022; 5 (1): 25–34. DOI: <https://doi.org/10.15276/aait.05.2022.2>.
2. Nazarova, O. S., Osadchyy, V. V. & Rudim, B. Y. “Computer simulation of the microprocessor liquid level automatic control system”. *Applied Aspects of Information Technology*. 2023; 6 (2): 163–174. DOI: <https://doi.org/10.15276/aait.06.2023.12>.
3. Kovalev, I. S., Drozd, O. V., Rucinski, A., Drozd, M. O., Antoniuk, V. V. & Sulima Y. Y. “Development of computer system components in critical applications: Problems, their origins and solutions”. *Herald of Advanced Information Technology*. 2020; 3 (4): 252–262. DOI: <https://doi.org/10.15276/hait.04.2020.4>.
4. Antoniuk, V. V., Drozd, M. O. & Drozd, O. B. “Power-oriented checkability and monitoring of the current consumption in FPGA projects of the critical applications”. *Applied Aspects of Information Technology*. 2019; 2 (2): 105–114. DOI: <https://doi.org/10.15276/aait.02.2019.2>.
5. Joshi, H. & Sinha, N. K. “Adaptive fault tolerant control design for stratospheric airship with actuator faults”. *IFAC-PapersOnLine*. 2022; 55 (1): 819–825. DOI: <https://doi.org/10.1016/j.ifacol.2022.04.134>. <https://www.scopus.com/pages/publications/85132157930>
6. Abbaspour, A., Mokhtari, S., Sargolzaei, A. & Yen, K. K. “A survey on active fault-tolerant control systems”. *Electronics*. 2020; 9 (9): 1–23, <https://www.scopus.com/pages/publications/85090902832>. DOI: <https://doi.org/10.3390/electronics9091513>.
7. Nedeljkovic, J. N., Dosic, S. M. & Nikolic, G. S. “A survey of hardware fault tolerance techniques”. *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies, Icest 2023 – Proceedings*. 2023. p. 223–226, <https://www.scopus.com/pages/publications/85167870342>.

DOI: <https://doi.org/10.1109/ICEST58410.2023.10187275>.

8. Dong, H., Xie, H. & Xu, H. “Reliability evaluation of motor controllers based on FMEA and DFTA”. *Hangkong Dongli Xuebao/Journal of Aerospace Power*. 2025; 40 (9): 20230656, <https://www.scopus.com/pages/publications/105011205376>.

DOI: <https://doi.org/10.13224/j.cnki.jasp.20230656>.

9. Jianu, M., Daus, L., Dragoi, V. F. & Beiu, V. “Reliability polynomials of consecutive-k-out-of-n:F systems have unbounded roots”. *Networks*. 2023; 82 (3): 222–228, <https://www.scopus.com/pages/publications/85163222383>. DOI: <https://doi.org/10.1002/net.22168>.

10. Dong, H., Xie, H. & Xu, H. “Reliability evaluation of motor controllers based on FMEA and DFTA”. *Hangkong Dongli Xuebao/Journal of Aerospace Power*. 2025; 40 (9): 20230656, <https://www.scopus.com/pages/publications/105011205376>.

DOI: <https://doi.org/10.13224/j.cnki.jasp.20230656>,

11. Yin, J. & Cui, L. “Reliability for consecutive-k-out-of-n:F systems with shared components between adjacent subsystems”. *Reliability Engineering & System Safety*. 2021; 210: 107532, <https://www.scopus.com/pages/publications/85100736521>.

DOI: <https://doi.org/10.1016/j.ress.2021.107532>.

12. Nashwan, I. “Reliability and failure probability functions of the m-Consecutive-k-out-of-n: F linear and circular systems”. *Baghdad Science Journal*. 2021; 18 (2): 430, <https://www.scopus.com/pages/publications/85100161580>.

DOI: <https://doi.org/10.21123/bsj.2021.18.2.0430>.

13. Yin, J., Cui, L. & Balakrishnan N. “Reliability of consecutive-(k,l)-out-of-n: F systems with shared components under non-homogeneous Markov dependence”. *Reliability Engineering & System Safety*. 2022; 224: 108549, <https://www.scopus.com/pages/publications/85129557917>.

DOI: <https://doi.org/10.1016/j.ress.2022.108549>.

14. Nakamura, T., Yamamoto, H. & Akiba, T. “Reliability of a toroidal connected-(r,s)-out-of-(m,n):F lattice system”. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (2): 329–338, <https://www.scopus.com/pages/publications/85077453315>.

DOI: <https://doi.org/10.1177/1748006X19893533>.

15. Romankevich, A., Feseniuk, A., Maidaniuk, I. & Romankevich, V. “Fault-tolerant multiprocessor systems reliability estimation using statistical experiments with GL-models”. *Advances in Intelligent Systems and Computing*. 2019; 754: 186–193, <https://www.scopus.com/pages/publications/85047465084>.

DOI: https://doi.org/10.1007/978-3-319-91008-6_19.

16. Romankevich, V. A., Morozov, K. V., Romankevich, A. M., Morozova, A. V. & Zacharioudakis, L. “On the method of building of non-basic GL-models which are formed on combination of edge functions of basic models”. *Applied Aspects of Information Technology*. 2024; 7 (2): 175–188. DOI: <https://doi.org/10.15276/aait.07.2024.13>.

Received 26.08.2026

Received after revision 28.09.2026

Accepted 07.10.2026

DOI: <https://doi.org/10.15276/ict.03.2026.01>

УДК 004.05

Послідовно-паралельна композиція графо-логічних моделей відмовостійких багатопроцесорних систем

Романкевич Віталій Олексійович¹⁾

ORCID: <https://orcid.org/0000-0003-4696-5935>; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058

Морозов Костянтин В'ячеславович¹⁾ORCID: <https://orcid.org/0000-0003-0978-6292>; mcng@ukr.net. Scopus Author ID: 57222509251**Романкевич Олексій Михайлович¹⁾**ORCID: <https://orcid.org/0000-0001-5634-8469>; romankev@scs.kpi.ua. Scopus Author ID: 6602114176**Галицький Данііл Володимирович¹⁾**ORCID: <https://orcid.org/0009-0004-4421-3443>; zipper135401@gmail.com. Scopus Author ID: 58553487600**Захаріудакіс Лефтеріс²⁾**ORCID: <https://orcid.org/0000-0002-9658-3073>; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200¹⁾ Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна²⁾ Університет Неаполіс Пафос, Данаїс Авеню, 2 Пафос, 8042, Кіпр

АНОТАЦІЯ

У роботі розглянуто побудову складених графо-логічних моделей поведінки відмовостійких багатопроцесорних систем у потоці відмов. Такі моделі можуть використовуватися для оцінювання показників надійності систем шляхом проведення статистичних експериментів. Побудова моделей складних систем безпосередньо за умовами їх роботоздатності може бути трудомісткою, тому доцільним є поділ початкової задачі на простіші складові з подальшим поєднанням побудованих для них моделей. Метою дослідження є встановлення залежностей між станами складових двополосних графів і станом результуючого графа за їх послідовного та паралельного з'єднання. Для опису поведінки двополосного графа введено три взаємовиключні класи: повної двополосної зв'язності, розділеної двополосної зв'язності та двополосної незв'язності. Доведено, що ці класи охоплюють усі можливі стани двополосного графа. Встановлено необхідні й достатні умови належності результуючого графа до кожного з уведених класів за послідовного та паралельного з'єднання його складових. Отримані залежності дають змогу визначати стан складеної моделі за станами її складових і способом їх з'єднання без повторного аналізу внутрішньої структури кожного графа. Розглянуто елементарні однореберні графо-логічні моделі та показано можливість їх використання як базових складових для побудови моделей зі складнішою поведінкою. Застосування отриманих результатів продемонстровано на прикладі побудови складеної графо-логічної моделі відмовостійкої багатопроцесорної системи з поетапним виконанням задачі та альтернативними умовами використання процесорних ресурсів. Модель сформовано шляхом послідовно-паралельного з'єднання елементарних однореберних моделей, реберні функції яких відображають умови виконання окремих етапів. Повний перебір можливих векторів стану підтвердив відповідність зв'язності результуючого графа заданим умовам роботоздатності системи. Перспективними напрямками подальших досліджень є розроблення формального алгоритму вибору складових моделей і способу їх поєднання безпосередньо за описом системи, а також дослідження можливості використання як складових готових багатореберних моделей, побудованих відомими методами.

Ключові слова: графо-логічні моделі; відмовостійкі багатопроцесорні системи; поведінка в потоці відмов; двополосні графи; композиція моделей; зв'язність графа; послідовне з'єднання; паралельне з'єднання; реберні функції

ABOUT THE AUTHORS



Vitaliy A. Romankevich - Doctor of Engineering Sciences, Professor, Head of System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ORCID: <https://orcid.org/0000-0003-4696-5935>; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058

Research field: Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Discrete mathematics

Романкевич Віталій Олексійович - доктор технічних наук, професор, завідувач кафедри Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Галузь досліджень: оцінювання надійності відмовостійких багатопроцесорних систем; GL-моделі; самодіагностовані системи; діагностика багатопроцесорних систем; дискретна математика



Kostiantyn V. Morozov - PhD, Associate Professor at the System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ORCID: <https://orcid.org/0000-0003-0978-6292>; mcng@ukr.net. Scopus Author ID: 57222509251

Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation; Self-diagnosable multiprocessor systems

Морозов Костянтин В'ячеславович - кандидат технічних наук, доцент кафедри Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Галузь досліджень: GL-моделі, оцінювання надійності відмовостійких багатопроцесорних систем; самодіагностовані багатопроцесорні системи



Alexei M. Romankevich - Doctor of Engineering Sciences, Professor, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ORCID: <https://orcid.org/0000-0001-5634-8469>; romankev@scs.kpi.ua. Scopus Author ID: 6602114176

Research field: Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Multi-valued logic

Романкевич Олексій Михайлович - доктор технічних наук, професор, кафедра Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Галузь досліджень: оцінювання надійності відмовостійких багатопроцесорних систем; GL-моделі; самодіагностовані системи; діагностика багатопроцесорних систем; багатозначна логіка



Daniil V. Halytsky - postgraduate student, System Programming and Specialized Computer System Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine

ORCID: <https://orcid.org/0009-0004-4421-3443>; zipper135401@gmail.com; Scopus Author ID: 58553487600

Research field: GL-models building and processing optimization; Parallel computing and multi-core system optimization; Algorithm design for distributed and high-performance computing; Graph algorithms and their applications in complex systems; Computational efficiency in large-scale data processing

Галицький Данііл Володимирович - аспірант, кафедра Системного програмування і спеціалізованих комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Галузь досліджень: побудова GL-моделей та оптимізація їх роботи; паралельні обчислення та оптимізація для багатоядерних систем; розробка алгоритмів для розподілених і високопродуктивних обчислень; алгоритми на графах та їх застосування в складних системах; обчислювальна ефективність при обробці великих обсягів даних



Lefteris Zacharioudakis - PhD (Eng), Assistant professor, Neapolis University Pafos, 2, Danaï Ave., Pafos, 8042, Cyprus

ORCID: <https://orcid.org/0000-0002-9658-3073>; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200

Research field: Principles of CyberSecurity; Operating systems; Information Security; Cryptography; Penetration Testing

Захаріудакіс Лефтеріс - кандидат технічних наук, доцент, Університет Неаполіс Пафос, Данаїс Авеню, 2, Пафос, 8042, Кіпр

Галузь досліджень: принципи кібербезпеки; операційні системи; інформаційна безпека; криптографія; тестування на проникнення