

UDC 519.177:[512.643.12:512.643.93](045)

DOI [https://doi.org/10.24144/2616-7700.2026.48\(1\).84-95](https://doi.org/10.24144/2616-7700.2026.48(1).84-95)

N. A. Yakimova

Odessa I. I. Mechnikov National University,
Associate Professor of the Department of Algebra, Geometry and Differential Equations,
Candidate of Technical Sciences, Associate Professor
nataliya_yakimova@onu.edu.ua
ORCID: <https://orcid.org/0009-0005-8577-2775>

PECULIARITIES OF MATRIX EXECUTION OF SOME BINARY OPERATIONS ON ARBITRARY GRAPHS

Graph theory has a wide range of practical applications. Graphs play an important role in scientific research (e.g., electrical circuits) and also surround us in everyday life (e.g., road and path maps). For everyday use, the geometric implementation of graphs is certainly the most convenient. But for computer information processing, this is not rational. In these cases, algebraic, namely matrix representation of graphs is used. Therefore, research devoted to this topic is becoming increasingly important. This article proves the possibility of algebraically performing some binary operations on adjacency matrices representing graphs. These methods have their own peculiarities and limitations, which are emphasized in this article.

Keywords: directed and undirected graph, adjacency matrix, incidence matrix, operations on graphs, elementary logical operations, Boolean matrix, multivalued logic.

1. Introduction. Today, science faces ever-new challenges. Systematization and digitalization, which permeate all spheres of human activity, require the establishment of interdisciplinary connections between branches of science that have historically been separated from each other by the division into purely humanitarian and natural sciences. Graph theory has been an applied branch of discrete mathematics since its inception. Its applications in natural and computer sciences have become commonplace and are not surprising to ordinary users. But in recent years, artificial intelligence systems that use natural language modeling to create interfaces of any level have been rapidly developing. In this regard, graph theory has also found its application in linguistics [1]. The mathematical basis for computer processing of natural speech includes models that use the geometric apparatus of graph theory. It allows us to visually depict the connections between elements of various systems. When studying a system of objects connected by some arbitrary types of relationships, both directed and undirected graphs can be used. Each such system is an ordered collection of elements with which certain changes can occur. At the same time, each such specific system can be represented graphically as a graph or in digital format as an adjacency matrix or incidence matrix of such a graph. In any case, changes in the elements of the system or the results of the interaction of different such systems are reflected by operations on the vertices or edges of the corresponding graphs. But computer processing of information involves its digital representation in matrix form. Therefore, a topical task is to study the regularities of transformations of arbitrary graphs precisely in matrix representation without resorting to their geometric implementation.

2. Statement of the task. The geometric implementation of all operations on graphs has already been well studied and described [2]. The classical algebraic

matrix apparatus is also widely represented in mathematical research [3]. This article aims to establish a correspondence between known operations on matrices and some operations on elements of an arbitrary graph or operations that reflect the interaction of several graphs. In this case, any known algebraic apparatus for processing matrices of various nature can be involved. Intuitively, the algorithms for such processing of adjacency and incidence matrices of arbitrary graphs are clear [4], but they require rigorous mathematical proof. The presence of a mathematical justification for these algorithms makes it possible to use the geometric representation of graphs only at the first (initial description of the system) and last (visual representation of the final result of the system's functioning) stages. In addition, in this way, a transition can be made from a geometric to an algebraic method of not only representing, but also processing various information, which greatly facilitates its computer processing.

3. Review of literature. In [5], algorithms for characterizing any arbitrary graph by all its matrices are described in detail, as well as algorithms for transitioning between different matrix representations of the same graph without restoring its geometric realization. The classical literature on linear algebra describes possible arithmetic operations on arbitrary matrices [3, 6]. The works [7, 8, 9] describe in detail algorithms for performing logical operations on Boolean and predicate logical matrices in two-valued logic. The apparatus of multivalued logic is investigated in [10, 11]. Generalizing these mathematical methods, it can be stated that if matrices are Boolean, then both ordinary algebraic operations on matrices and operations of two-valued logic described in [8, 9] can be performed with them. If matrices are not Boolean, then to perform logical operations of disjunction and conjunction with them, it is necessary to use the apparatus of multivalued logic. In this case, the operations of disjunction and conjunction of matrix elements are performed according to the following rules [10]:

$$a \vee b = \max\{a, b\}, \quad (1)$$

$$a \wedge b = \min\{a, b\}, \quad (2)$$

But both logical and arithmetic operations on matrices require certain conditions regarding their dimensionality [3, 7].

The classical literature on graph theory [2, 11, 12, 13] considers operations on graphs exclusively in geometric representation. Single attempts to perform some of these operations matrix-wise consider only the cases of simple graphs with the same number of vertices, using for this purpose the apparatus of exclusively classical two-valued logic [14] without additional preliminary matrix transformations. Pseudographs and multigraphs remain outside the scope of these studies. But in practice, different systems represented by geometric implementations of graphs do not always have the same number of objects (nodes). Therefore, the graphs corresponding to them will have a different number of vertices. This implies a different dimension of their adjacency matrices. Neither logical nor arithmetic operations can be performed on such matrices. This obstacle can be avoided by reducing both matrices to the same dimension by introducing additional identically named all-zero rows and columns into them. According to the characterization of graphs by their adjacency matrices, such pairs will correspond to isolated vertices [5, 15]. The new

extended adjacency matrices of both graphs participating in the operation will have the same dimension. Implementing such matrix expansion can lead to irrational use of the memory required to store this amount of information. This becomes especially relevant in cases where such a reduction required multiple execution of the operation of inserting an isolated vertex. The matrices become very sparse and their processing is too cumbersome. However, this obstacle can be avoided if the obtained extended adjacency matrices of the studied graphs are presented in the form of binary predicates [16]. This representation allows to significantly reduce the amount of memory used. The computer procedure for performing operations on logical matrices does not require additional space for storing information about their zero elements, storing the coordinates of only significant cells of both the studied and the obtained new matrices. As already noted above, this article aims only to mathematically prove the principle possibility of matrix execution of binary operations on graphs. Therefore, in this paper, to simplify the illustration of the obtained results, graphs that have the same number of vertices in advance are considered as examples. It is also assumed that vertices with the same numbers in these graphs have the same applied content load.

As is known, graphs are isomorphic if and only if their adjacency matrices can be obtained from each other by simultaneous permutations of the same-named rows and columns (i.e., simultaneously with the permutation of the i -th and j -th rows of the matrix, the permutation of the i -th and j -th columns of the matrix also occurs) [2]. From this fact it follows that using an isomorphism transformation for each graph in the extended adjacency matrix, new isolated vertices will correspond to such row-column pair numbers that reflect vertices that are absent in one graph but present in the other graph.

4. Main result. Let's consider the basic operations on graphs.

Theorem 1. *The adjacency matrix of the result of the graph union operation corresponds to the disjunction of the adjacency matrices of the graphs being joined.*

Proof. By definition, the graph H is the union of the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, if $H = (V_1 \cup V_2, E_1 \cup E_2)$. If the graphs being joined have the same number of vertices, then the set of vertices of the graph H coincides with the sets of vertices of the graphs G_1 and G_2 , i.e. $V_1 \cup V_2 = V_1 = V_2$. In this case, their adjacency matrices $A(H)$, $A(G_1)$ and $A(G_2)$ will have the same dimension, so any operations can be performed with them without additional preliminary transformations. If the number of vertices in the graphs under study is different or has a different semantic load and, as a result, different numbering, then after introducing additional zero row-column pairs, we obtain matrices of the same dimension. By permutations, each of these matrices can be reduced to matrices of graphs isomorphic to the original ones, where all vertices have the same numbering, which corresponds to their meaningful loading. The elements of the adjacency matrix correspond to the number of edges connecting the corresponding vertices. Therefore, if in at least one of the graphs the vertices are adjacent, i.e. connected by a certain number of edges, then in the adjacency matrix of the union of these graphs the specified vertices will be connected by the same number of edges. This corresponds to the definition of the disjunction operation in its multivalued sense according to formula (1). Thus, to calculate the adjacency matrix of the result of the union of two arbitrary graphs, it is necessary

to perform the disjunction operation of the adjacency matrices of these graphs. The theorem is proved.

We will illustrate the result of this theorem with an example. Consider the two directed graphs shown in Fig. 1.

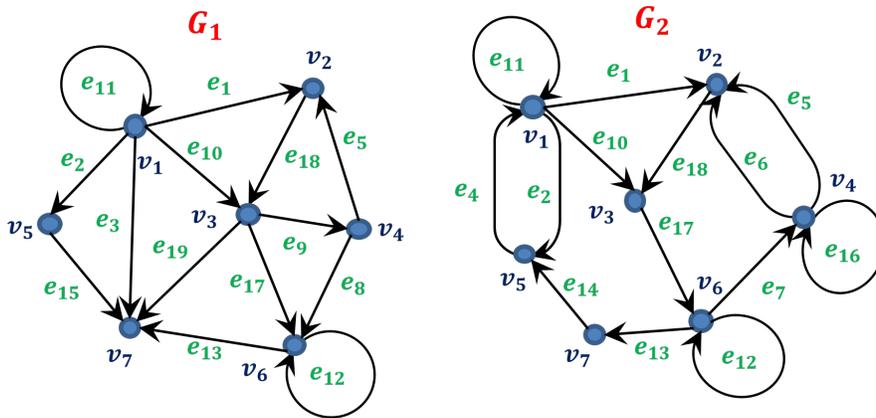


Figure 1. Directed graphs G_1 and G_2 .

For both of these graphs, we can construct adjacency matrices:

$$A(G_1) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad A(G_2) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

These matrices are constructed assuming that the row number corresponds to the starting vertex and the column number corresponds to the final vertex of each edge. But these graphs can be defined by these matrices from the very beginning. Performing operations on such graphs does not require reproduction of their geometric implementation [4].

The graph G_1 does not contain multiple edges, so its adjacency matrix is Boolean. Graph G_2 contains strictly parallel edges $e_5(v_4, v_2)$ and $e_6(v_4, v_2)$. Therefore, its adjacency matrix contains the element $a_{42} = 2$, i.e. it is not Boolean [8, 9]. But if we take into account that the disjunction operation for multivalued logic is performed according to rule (1), then the disjunction for these matrices takes the form [4, 17]:

$$A(G_1) \vee A(G_2) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = A_{\cup}.$$

If the union operation of the specified graphs is performed geometrically, then we will get the graph shown in Fig. 2.

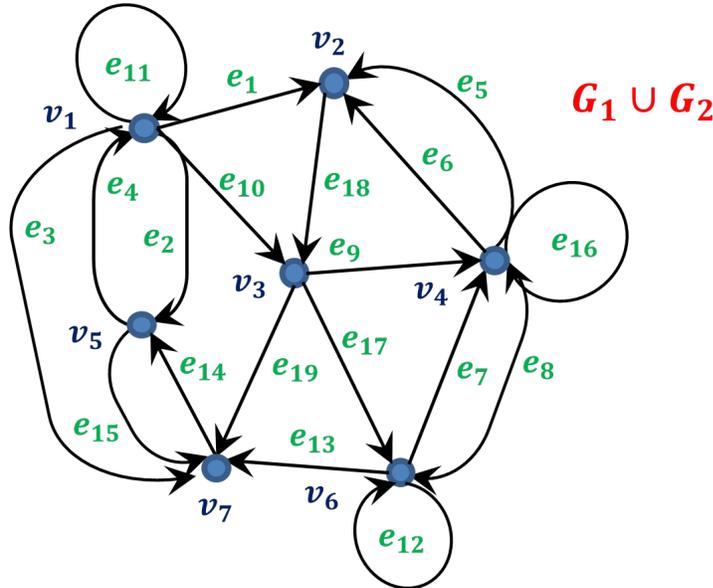


Figure 2. Graph $G_1 \cup G_2$.

It is easy to see that the adjacency matrix A_{\cup} corresponds to this graph.

Theorem 2. *The adjacency matrix of the result of the graph intersection operation corresponds to the conjunction of the adjacency matrices of the intersecting graphs.*

Proof. By definition, the graph F is the intersection of the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, if $F = (V_1 \cap V_2, E_1 \cap E_2)$. If the graphs being joined have the same number of vertices, then the set of vertices of the graph F coincides with the sets of vertices of the graphs G_1 and G_2 , i.e. $V_1 \cap V_2 = V_1 = V_2$. In this case, their adjacency matrices $A(F)$, $A(G_1)$ and $A(G_2)$ will have the same dimension, so any operations can be performed with them without additional preliminary transformations.

If the sets of vertices in the graphs under study do not coincide, then, as in the case of unification graphs, after introducing additional zero row-column pairs, we obtain matrices of the same dimension, which are transformed into matrices of graphs isomorphic to the original graphs. The elements of the adjacency matrix correspond to the number of edges connecting the corresponding vertices. Therefore, if in at least one of the graphs any two vertices are connected by a smaller number of edges than in the second graph, then in the adjacency matrix of the intersection of these graphs the specified vertices will be connected by the same smaller number

of edges. This corresponds to the definition of the conjunction operation in its multivalued sense according to formula (2). Thus, to calculate the adjacency matrix of the result of the intersection of two arbitrary graphs, it is necessary to perform the conjunction operation of the adjacency matrices of these graphs. The theorem is proved.

Let us illustrate this theorem using the example of the graphs shown in Fig. 1. As already noted, for multivalued logic the conjunction operation is performed according to rule (2), so the conjunction for the matrices $A(G_1)$ and $A(G_2)$ takes the form:

$$\begin{aligned}
 A(G_1) \wedge A(G_2) &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \wedge \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = A_{\cap}.
 \end{aligned}$$

If the intersection of the specified graphs is performed geometrically, we will get the graph shown in Fig. 3.

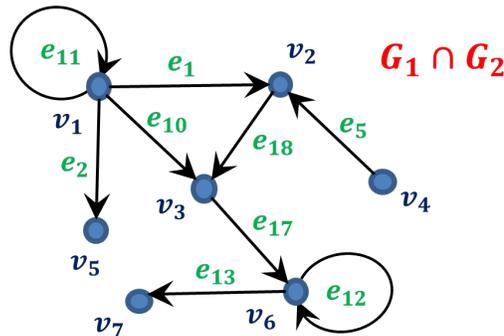


Figure 3. Graph $G_1 \cap G_2$.

It is easy to see that the adjacency matrix A_{\cap} corresponds to this graph [4].

Theorem 3. *The adjacency matrix of the result of performing the ring sum operation of graphs corresponds to the arithmetic subtraction of the adjacency matrices of the union and intersection of the graphs that participate in the specified operation.*

Proof. By definition, a graph R is a ring sum of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ if it does not contain isolated vertices and consists of edges belonging to either graph G_1 or graph G_2 , but not to both simultaneously, i.e. [17]

$$E = \{(E_{G_1} \cup E_{G_2}) \setminus (E_{G_1} \cap E_{G_2})\}. \tag{3}$$

The set of edges $E_{G_1} \cup E_{G_2}$ is the result of the graphs union operation, and the set $E_{G_1} \cap E_{G_2}$ is the result of their intersection. This means that the matrix

$$A(R) = A_{\cup} - A_{\cap} = (A(G_1) \vee A(G_2)) - (A(G_1) \wedge A(G_2)), \quad (4)$$

will contain complete information about all edge-connected vertices of the graph R , i.e. will be its adjacency matrix in the extended sense. The theorem is proved.

By the definition of a ring sum of graphs, the graph obtained as a result of this operation cannot contain isolated vertices. Therefore, if they appear, they must be deleted from the resulting graph. Deleting a vertex from the graph entails deleting all edges incident to it, that is, deleting all connections of this object or node with other objects or nodes [4, 18]. This means that when deleting the vertex v_i from the adjacency matrix, it is necessary to delete the i -th row and i -th column. In this regard, the algorithm for performing the operation of deleting the vertex v_i from the graph in the matrix representation is similar to the algorithm for constructing the minor M_{ii} for the adjacency matrix of this graph [6]. The algorithm for deleting rows and columns from the matrix is already computerized. In this case, the software implementation will provide two shifts: for rows and columns [4, 17].

A sign of an isolated vertex in the adjacency matrix is the presence of a row and column of the same name that are all zero [5]. According to the described algorithm, this zero row-column pair should be removed from the extended adjacency matrix of the ring sum of graphs if such a pair appeared as a result of the procedure described in Theorem 3. The final matrix obtained as a result of these actions will be the adjacency matrix of the ring sum of the graphs under study.

Let us illustrate the result of Theorem 3 using the example of the graphs shown in Fig. 1. For these graphs, the matrices A_{\cup} and A_{\cap} have already been obtained. Therefore,

$$\begin{aligned} A_{\cup} - A_{\cap} &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = A_{\oplus}. \end{aligned}$$

This matrix does not contain all zero row-column pairs. This means that the ring sum operation did not result in any isolated vertices that would be subsequently deleted from the resulting graph. Therefore, the obtained adjacency matrix does not require further processing by constructing its corresponding minor. Thus, the

resulting matrix is the final adjacency matrix of the ring sum of the considered graphs.

If we perform the ring sum operation of graphs G_1 and G_2 geometrically, we obtain the graph shown in Fig. 4.

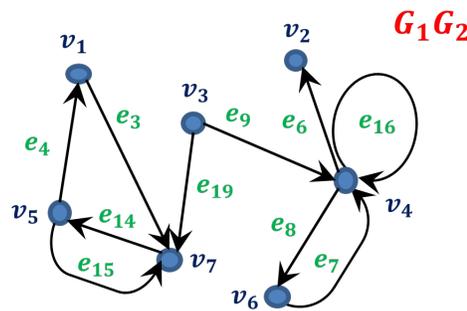


Figure 4. Graph $G_1 \oplus G_2$.

For this graph, it is also easy to see that the adjacency matrix A corresponds to it.

Theorem 4. *If two graphs do not contain multiple edges, or their number between corresponding vertices in these graphs differs by no more than one, then the adjacency matrix of the ring sum of these graphs can be obtained as a result of the sum modulo 2 operation of their adjacency matrices.*

Proof. As in the cases of graph union and intersection, first, if necessary, it is necessary to construct extended adjacency matrices for both graphs in order to achieve their same dimension. After that, using the isomorphism transformation, they should be reduced to matrices corresponding to the graphs under study. After that, the following situations are possible for the obtained adjacency matrices.

- A) If the graphs do not contain multiple edges at all, then their adjacency matrices are Boolean. In this case, if two vertices are adjacent in both graphs (in the case of a directed graph, the vertices in both graphs are connected by the same directed edge), then in both matrices there will be ones at the corresponding place. But the given edge is not included in the result of the ring sum of these graphs. Therefore, in the adjacency matrix of the result, the corresponding element must be equal to zero. If two vertices are adjacent in only one graph, then such an edge will be an element of the ring sum of these graphs, that is, the corresponding element of the adjacency matrix of the result will be equal to one. If the vertices are not adjacent in any graph, then the result of the ring sum will also not be adjacent, i.e. the corresponding element of the result's adjacency matrix will be equal to zero. All this corresponds to the sum modulo 2 as an elementary Boolean operation.
- B) In general, the sum modulo 2 is defined as the remainder of dividing the sum of the corresponding numbers by 2. If two vertices in both graphs are connected by the same number of q -multiple edges, then the sum of these edges will be equal to $2q$, which is an even number. The remainder of dividing such a number by 2 will always be zero, i.e. the corresponding matrix element will be equal to zero. This means that no multiple edge common to both graphs will be an

element of the ring sum of graphs, which corresponds to the definition of this operation.

- C) If any two vertices in both graphs are connected by multiple edges, and the number of these edges differs by more than one, then the number of multiple edges between the vertices when performing the ring sum of graphs must be equal to this number. This means that the corresponding element of the adjacency matrix of the result of this operation must be greater than one. But such a number cannot be the result of the sum modulo 2 operation. In this case, to calculate the adjacency matrix of the ring sum of graphs, we must use the formula (4) proposed by Theorem 3.
- D) If the number of multiple edges connecting two vertices in both graphs differs by exactly one, then these numbers can be denoted as k and $k + 1$. It follows from this that

$$(k + k + 1) \bmod 2 = (2k + 1) \bmod 2 = 1.$$

So, in this situation, the result of the sum modulo 2 is indeed equal to the number of edges that are different among the multiple edges between the two vertices. Incidentally, situation A) can be considered a special case of this situation.

The theorem is proven.

Let us illustrate the result of Theorem 4 with an example. Let us perform the sum modulo 2 operation with the given matrices $A(G_1)$ and $A(G_2)$, keeping in mind its general definition as the remainder of dividing the corresponding sum by 2 [4]. The indicated matrices correspond to situation D) described in Theorem 4.

$$\begin{aligned} A(G_1) \oplus A(G_2) &= \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = A_{\oplus}. \end{aligned}$$

This matrix coincides with the matrix A for the ring sum of graphs G_1 and G_2 , calculated by formula (4), that is, the adjacency matrix of the graph obtained as a result of the ring sum of two graphs can be calculated in this way. The results of Theorem 3 and Theorem 4 clearly show that for matrix execution of operations on graphs, the simultaneous use of both arithmetic and logical operations is permissible.

5. Conclusions and prospects for further research. In the graphs considered as example, each edge is given a serial number. In practical application, these numbers may mean a certain content load. But with matrix display, this content can

be lost. The matrix reflects the presence or absence of an edge, that is, the presence or absence of a connection between objects. Therefore, if in two graphs between two vertices the same edge has a different content load (for example, a road and a dirt road), then the adjacency matrix will only show the presence or absence of this connection without explaining its nature. But usually, in practical applications, information about the presence of a connection is sufficient, therefore, for binary operations on directed graphs [17], the use of elementary multivalued logic operations on adjacency matrices according to proven algorithms is an effective mathematical tool. The same algorithms have differences depending on whether directed or free graphs are involved in the considered operations. Depending on the types of graphs, there are also restrictions on the display of meaningful information by the matrices of these graphs. But in practical applications, these restrictions are usually insignificant. Therefore, for each operation on graphs and each type of graph, it is possible to propose a combination of algebraic operations (arithmetic and logical) that allow obtaining the matrix of a new graph, or a clear, easily programmable algorithm for transforming the matrices of the initial graphs [4, 19]. None of the considered operations on graphs is impossible in the matrix implementation. The proposed algorithms can significantly simplify the computer processing of graphs.

With certain restrictions, the obtained results can be extended to the representation of graphs by incidence matrices. From the adjacency matrix, one can always go to the incidence matrix of a graph without restoring its geometric realization. For undirected graphs, incidence matrices are always Boolean [8]. Therefore, operations on them do not require the involvement of the multivalued logic apparatus. However, even in the case of the same number of vertices and edges, as for the adjacency matrix, one must pay attention to the numbering (naming) of these objects. Both the vertex sets and the edge sets for two graphs can be equivalent, but not identical [5]. In this case, the incidence matrices of the graphs under study also require a preliminary expansion procedure. However, the resulting zero columns in the incidence matrix are a sign of a loop. Exceptions may be undirected hypergraphs, where the loop feature is a single unit in the column. Constructing an adjacency matrix for hypergraphs is a very cumbersome procedure [20]. Therefore, the next direction of research should be to perform operations on hypergraphs using incidence matrices, the construction of which in this case does not cause difficulties. Therefore, it is these matrix models that can become an acceptable mathematical apparatus for studying complex systems represented by hypergraphs.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Copyright ©



(2026). Yakimova N. A. This work is licensed under a Creative Commons Attribution 4.0 International License.

References

1. Yakimova, N. A. (2008). Variants of formalization of natural language sentences and their graphical representation. *Opera annua in linguistica generalis*. Odessa: Astroprint, (8–9), 302–308 [In Ukrainian].
2. Kapitonova, Yu. V., Kryvyy, S. L., Letychevskiy, O. A., Lutzkiy, G. M., & Pechurin, M. K. (2002). *Foundation of Discrete Mathematics*. Kyiv: Naukova dumka [In Ukrainian].
3. Gantmacher, F. R. (2010). *The Theory of matrices*. Kyiv: Nadrukovano v Ukraine [In Ukrainian].
4. Yakimova, N. A., & Klishin, N. E. (2022). Matrix representation of operations on graphs. *Researches in mathematics and mechanics*, 27(1–2(38–39)), 121–141. [https://doi.org/10.18524/2519-206X.2022.1-2\(39-40\).294314](https://doi.org/10.18524/2519-206X.2022.1-2(39-40).294314) [In Ukrainian].
5. Yakimova, N. A. (2021). *Discrete Math. Part 1. Set theory. Graf theory*. Odessa: ONU im. I. I. Mechnikova [In Ukrainian].
6. Zykov, A. A. (2007). *Lectures on algebra*. Odessa: Astroprint [In Ukrainian].
7. Yakimova, N. A. (2024). Operation of reversing of logical matrices. *SWorldJournal*, 27(1), 165–172. <https://doi.org/10.30888/2663-5712.2024-27-00-017>
8. Yakimova, N. A. (2019). Predicative logical matrices. *Visnyk Odes'kogo Natsional'nogo universitetu. Researches in Mathematics and Mechanics*, 24(2(34)), 67–74. [https://doi.org/10.18524/2519-206x.2019.2\(34\).190052](https://doi.org/10.18524/2519-206x.2019.2(34).190052) [In Ukrainian].
9. Yakimova, N. A. (2023). Operations on block predicate matrices. *Researches in mathematics and mechanics*, 28(1–2(41–42)), 185–199. [https://doi.org/10.18524/2519-206X.2023.1-2\(41-42\).305269](https://doi.org/10.18524/2519-206X.2023.1-2(41-42).305269) [In Ukrainian].
10. Matvienko, M. P. (2012). *Computer logic*. Kyiv: Lira-K [In Ukrainian].
11. Kotzovskiy, V. M. (2016). *Discrete Mathematics and theory of algorithms*. Uzhgorod: UNU [In Ukrainian].
12. Nikolaeva, K. V., & Koybichuk, V. V. (2007). *Discrete analysis. Graphs and their applications in economics*. Sumy: UABS NBU [In Ukrainian].
13. Mikhailenko, V. M., Fedorenko, N. D., & Demchenko, V. V. (2003). *Discrete Mathematics*. Kyiv: European University [In Ukrainian].
14. Serpinska, O. I., Terentjev, O. O., Balina, O. I., Bezklubenko, I. S., & Ryabchun, Yu. V. (2023). *Discrete Mathematics*. Kyiv: KNUBA [In Ukrainian].
15. Yakimova, N. A., & Klishin, N. E. (2024). Binary operations on graphs in matrix representation. In *Proceedings of the XXX International Scientific and Practical Conference «Youth, education and science through today's challenges»*, Porto: Portugal, 120–125. <https://doi.org/10.46299/ISG.2024.1.30> [In Ukrainian].
16. Yakimova, N. A. & Simonova, I. G. (2007). Binary predicates as a main of representation Boolean logic objects. *Researches in mathematics and mechanics*, 12(7), 174–184 [In Ukrainian].

17. Yakimova, N. A., & Sharai, N. V. (2024). Performing some operations on graphs using adjacency matrices. In *Proceedings of the 10th International Scientific and Practical Conference «Theory and Practice of Science: Key Aspects»*, Rome: Italy, 184–196. <https://doi.org/10.51582/interconf.19-20.09.2024.019>
18. Yakimova, N. A. (2025). Possibility of matrix representation of basic unary operations on graphs. *SWorldJournal*, 30(2), 147–153. <https://doi.org/10.30888/2663-5712.2025-30-02-013>
19. Kruglov, V. E., & Yakimova, N. A. (2025). Predicate model of some operations on logic matrices. In *Proceedings of the 14th International Scientific and Practical Conference «International Forum: Problems and scientific solutions»*, Melbourne: Australia, 249–254. <https://doi.org/10.51582/interconf.2024.231>
20. Yakimova, N. A. (2026). Economic interpretation of some operations on undirected hypergraphs. *Grail of Science*, 61, 265–275. <https://doi.org/10.36074/grail-of-science.23.01.2026.027>

Якімова Н. А. Особливості матричного виконання деяких бінарних операцій над довільними графами.

Теорія графів має широке розповсюдження з практичної точки зору. Графи відіграють важливу роль в наукових дослідженнях (наприклад, електричні схеми), а також оточують нас у повсякденному житті (наприклад, карти доріг та шляхів). Для побутового застосування, безумовно, найзручнішою є геометрична реалізація графів. Але для комп'ютерної обробки інформації це не є раціональним. В цих випадках використовується алгебраїчне, а саме матричне подання графів. Тому все більшого значення набувають дослідження, присвячені саме цій темі. В даній статті доведено можливість алгебраїчного виконання деяких бінарних операцій над матрицями суміжності, якими подано графи. Ці методи мають свої особливості та обмеження, на яких зроблено акцент в даній статті.

Ключові слова: орієнтований та неорієнтований граф, матриця суміжності, матриця інцидентності, операції над графами, елементарні логічні операції, булева матриця, багатозначна логіка.

Received: 29.11.2025

Accepted: 17.12.2025

Published: 29.01.2026