

УДК 681.58

DOI: [10.26102/2310-6018/2020.28.1.014](https://doi.org/10.26102/2310-6018/2020.28.1.014)

## Представление алгоритмов системы управления сложными объектами в матрично–предикатном виде

**В.С. Поляков, О.А. Авдеюк, В.Ю. Наумов, И.Ю. Королева,  
И.Г. Лемешкина**

*Волгоградский государственный технический университет, Волгоград*

**Резюме.** В статье указано, что построение систем управления объектов, осуществляющих технологический процесс, начинается, как правило, с составления алгоритмов их функционирования. Этот процесс осуществляется зачастую эвристическими методами, сложные алгоритмы составляются отдельными блоками, а затем «сшиваются» в единое целое. В основном построение осуществляется в виде граф-схем, сложно в прочтении и неудобно в переработке. Качество этих алгоритмов полностью зависит от квалификации инженерного персонала, от знания технологии процесса, от владения теорией и практикой решения поставленной задачи. В данной работе показана возможность построения алгоритмов в матричной форме, то есть позволяющей получать формализованное описание в более удобном и компактном виде и дающее способ решения многих нестандартных ситуаций в процессе алгоритмизации. В статье рассмотрена возможность проведения ряда операций над граф-схемами, которые названы в дальнейшем «операциями доопределения», что позволило записать алгоритм в виде дуального графа и позволит представлять его виде модульных блоков, а также рассмотреть возможность представления алгоритмов в матрично-предикатном и таблично-предикатном виде. Сделан вывод о том, что задание алгоритмов систем управления сложными процессами в матрично-предикатном или таблично-предикатном виде даёт возможность использовать хорошо изученные методы теории графов, теории матриц и методы теории предикатов. Кроме того, появляется возможность использовать при работе с алгоритмами теоретико-множественные и алгебраические операции, разработанные для графов.

**Ключевые слова:** алгоритм, матрица, инцидентор, предикат, модульная структура.

**Для цитирования:** Поляков В.С., Авдеюк О.А., Наумов В.Ю., Королева И.Ю., Лемешкина И.Г. Представление алгоритмов системы управления сложными объектами в матрично–предикатном виде. *Моделирование, оптимизация и информационные технологии*. 2020;8(1). Доступно по: Available by: [https://moit.vivt.ru/wp-content/uploads/2020/02/PolyakovSoavtori\\_1\\_20\\_1.pdf](https://moit.vivt.ru/wp-content/uploads/2020/02/PolyakovSoavtori_1_20_1.pdf) DOI: 10.26102/2310-6018/2020.28.1.014

## Representation of algorithms of system of management of complex objects in matrico-predicate type

**V.S. Polyakov, O.A. Avdeyuk, V.Y. Naumov, I.Y. Koroleva, I.G. Lemeshkina**  
*Volgograd State Technical University, Volgograd*

**Abstract.** The article indicates that the construction of control systems for objects that carry out the process begins, as a rule, with the compilation of algorithms for their functioning. This process is often carried out by heuristic methods, complex algorithms are compiled in separate blocks, and then “stitched” into a single whole. Basically, the construction is carried out in the form of graph diagrams, is difficult to read and inconvenient to process. The quality of these algorithms depends entirely on the qualifications of the engineering staff, on the knowledge of the process technology, on knowledge of the

theory and practice of solving the problem. This paper shows the possibility of constructing algorithms in matrix form, that is, allowing one to obtain a formalized description in a more convenient and compact form and giving a way to solve many non-standard situations in the process of algorithmization. The article considers the possibility of carrying out a number of operations on graph diagrams, which are hereinafter referred to as “operations of additional definition”, which allowed us to write the algorithm in the form of a dual graph and allow us to represent it in the form of modular blocks, as well as to consider the possibility of representing the algorithms in matrix-predicate and tabular predicate form. It is concluded that defining the algorithms of control systems for complex processes in a mathematical-predicate or tabular-predicate form makes it possible to use well-studied methods of graph theory, matrix theory, methods of predicate theory. In addition, it becomes possible to use set-theoretic and algebraic operations developed for graphs when working with algorithms.

**Keywords:** algorithm, matrix, incidentor, predicate, modular structure.

**For citation:** Polyakov V.S., Avdeyuk O.A., Naumov V.Y., Koroleva I.Y., Lemeshkina I.G. Presentation of the algorithms of the control system for complex objects in a matrix – predicate form. *Modeling, optimization and information technology*. 2020;8(1). Available by: [https://moit.vivt.ru/wp-content/uploads/2020/02/PolyakovSoavtori\\_1\\_20\\_1.pdf](https://moit.vivt.ru/wp-content/uploads/2020/02/PolyakovSoavtori_1_20_1.pdf) DOI: 10.26102/2310-6018/2020.28.1.014 (In Russ).

## Введение

Построение автоматизированных систем управления техническими процессами состоит из двух взаимодополняющих частей: содержательной и формальной. Обобщённый подход к автоматизированным системам и их составляющим, определяющие их философские принципы составляют основу содержательной части. В ней описываются факторы, явления и технологии автоматизированных систем, осуществляющей процесс [1,2], а так же рассматривается суть автоматизации процесса [3,4]. Трактовка автоматизации процесса образует концепцию формального описания, законы и закономерности, принципы и способы решения соответствующих задач, исходный базис, основные определения и категории[5,6].

Формальная часть определяет, какие методики необходимо разработать для построения формального описания, какие логико-математических правила необходимо использовать, какого рода приемы и методы разработать.

Наиболее часто описание процессов осуществляется в виде алгоритмов

## Материалы и методы

**Описание алгоритмов.** Алгоритм – чётко определённое, заранее известное предписание исполнителю совершить задаваемую определенную последовательность действий для получения результата решения задачи за определённое число шагов [7,8]. Для задания алгоритмов чаще всего используются следующие способы: словесный, графический, псевдокод, программный.

При построении систем управления (СУ) часто используется графический способ задания алгоритмов в виде граф-схем алгоритмов (ГСА) [7], произвольная ГСА представлена на Рисунке (Рисунок 1).

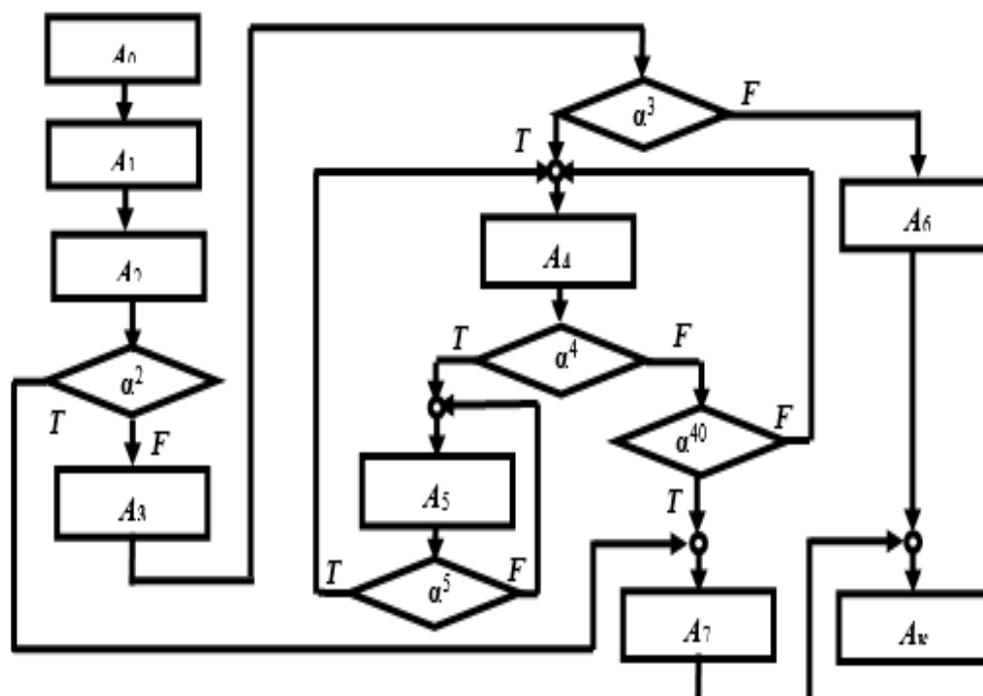


Рисунок 1 – Граф-схема алгоритма  
 Figure 1 – Graph-diagram of the algorithm

Здесь:

- $A = \{ A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8 \}$  – множество вершин ГСА, характеризующих блоки действия, например, технологические операции,
- $\alpha = \{ \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^{40} \}$  – множество вершин ГСА, характеризующих предикативные блоки, определяющих логику выполнения операций.

**Доопределение алгоритмов.** При задании СУ в виде ГСА следует учитывать возможность возникновения некоторых неточностей в функционировании СУ [9,10]:

- отсутствие в отдельных случаях фиксации перехода между операциями действия, например, блоки действия  $A_0 - A_1 - A_2$  в рассматриваемом алгоритме (Рисунок 1),

- переход от одного блока действия ( $A_4$ ) к другому ( $A_7$ ) осуществляется при выполнении условий, определяемых предикативными блоками  $\alpha^4$  и  $\alpha^{40}$ .

Эти неточности удаляются несколькими способами:

- а) заменить воздействие нескольких блоков действия ( $A_0 - A_1 - A_2$ ) одним блоком действия ( $A_{012}$ );

- б) выполнение каждого из рассматриваемых блоков действия ( $A_0, A_1, A_2$ ) фиксируется соответствующим предикативным блоком ( $\mu^0, \mu^1, \mu^2$ ).

Вторая неточность устраняется применением алгебры логики. Предикативные блоки  $\alpha^4$  и  $\alpha^{40}$  формируют сигналы:  $\mu^4_0 = \alpha^4 \wedge \alpha^{40}$ ,  $\mu^4_1 = \alpha^4 \wedge \alpha^{40}$ ,  $\mu^4_2 = \alpha^4 \wedge \alpha^{40}$ .

Устранение этих неточностей назовём операцией доопределения [11], в результате проведения этой операции получаем начертание ГСА в виде модульных блоков (Рисунок 2).

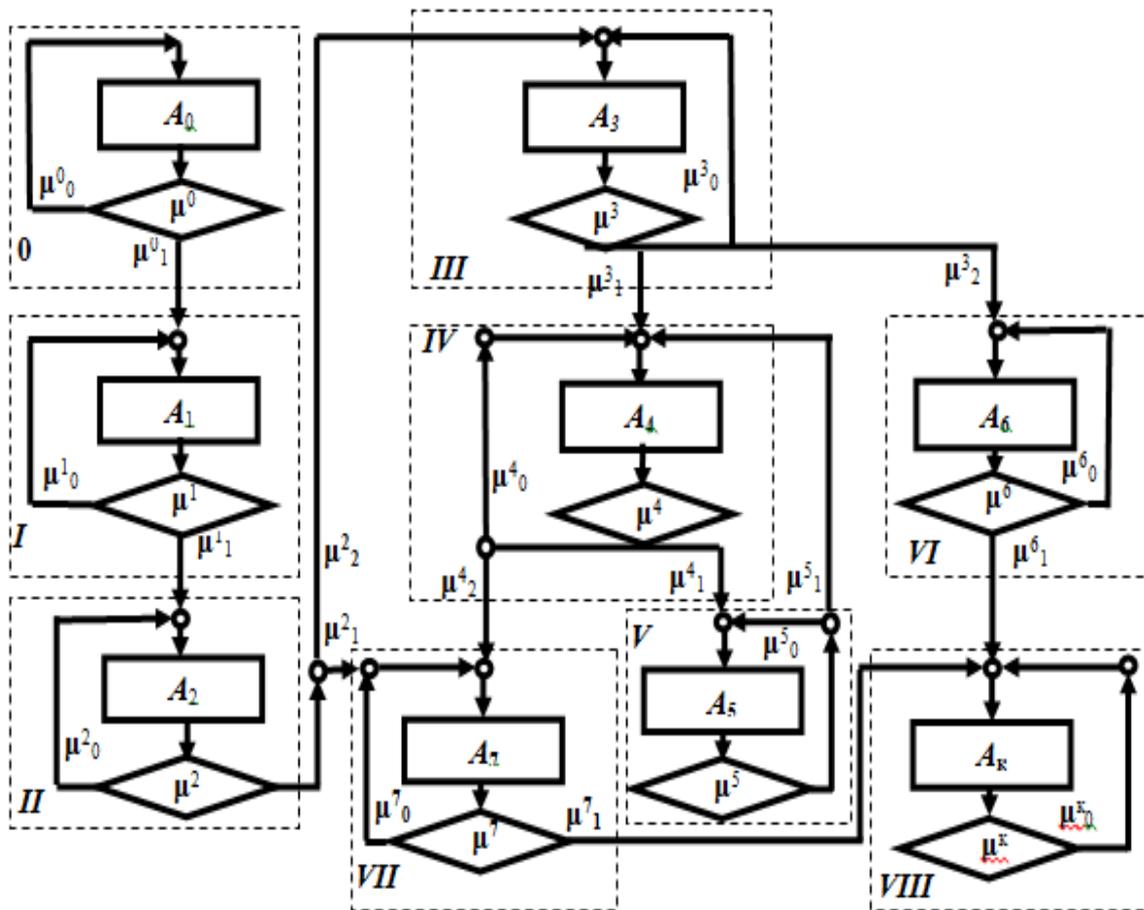


Рисунок 2 – Граф-схема алгоритма в результате операции доопределения  
 Figure 2 - Graph-diagram of the algorithm as a result of the operation of additional definition

В результате «доопределённая» ГСА можно записать в виде дуального графа (Рисунок 3).

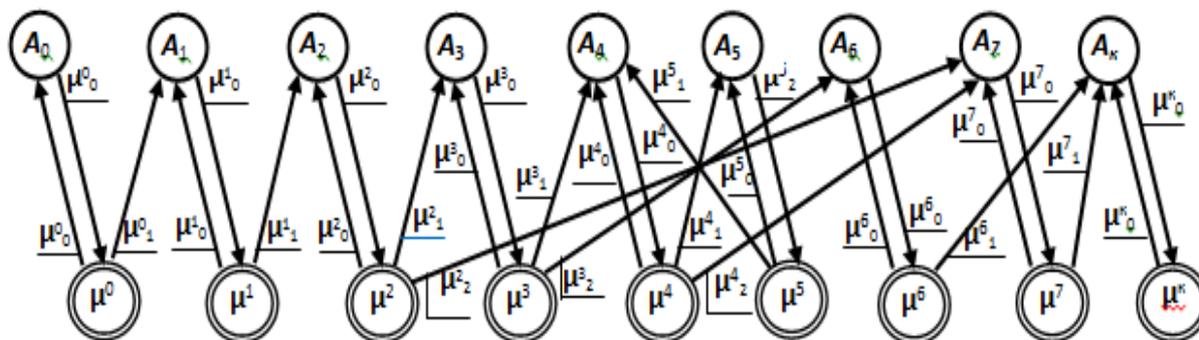


Рисунок 3 – Дуальный граф  
 Figure 3 - Dual graph

### Результаты

**Представление алгоритма в матрично-предикатном виде.** В [10-12] приведена разработанная авторами методология задания графа квадратной матрицей, каждый ненулевой элемент которой определяется истинным значением характеризующего граф трёхместного предиката. Граф на рисунке (Рисунок 3) представим в виде матрицы (Рисунок 4).

$$\mathbf{M}^A = \begin{pmatrix}
 A_0\mu^0_0A_0 & A_0\mu^0_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \mu^1\mu^0_0A_0 & \mu^1\mu^0_0\mu^1_1 & \mu^1\mu^0_1A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & A_1\mu^1_0A_1 & A_1\mu^1_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & \mu^1\mu^1_0A_1 & \mu^1\mu^1_0\mu^1_1 & \mu^1\mu^1_1A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & A_2\mu^2_0A_2 & A_2\mu^2_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & \mu^2\mu^2_0A_2 & \mu^2\mu^2_0\mu^1_1 & \mu^2\mu^2_1A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^2\mu^2_1A_7 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & \mu^2\mu^2_0A_2 & 0 & A_3\mu^3_0A_3 & A_3\mu^3_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & \mu^3\mu^3_0A_3 & \mu^3\mu^3_0\mu^1_1 & \mu^3\mu^3_1A_4 & 0 & 0 & 0 & \mu^3\mu^3_1A_6 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4\mu^4_0A_4 & A_4\mu^4_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^4\mu^4_0A_4 & \mu^4\mu^4_0\mu^1_1 & \mu^4\mu^4_1A_5 & 0 & 0 & 0 & \mu^4\mu^4_1A_7 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5\mu^5_0A_5 & A_5\mu^5_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^5\mu^5_0A_5 & 0 & \mu^5\mu^5_0A_5 & \mu^5\mu^5_0\mu^1_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_6\mu^6_0A_6 & A_6\mu^6_0\mu^1_1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^6\mu^6_0A_6 & \mu^6\mu^6_0\mu^1_1 & 0 & 0 & \mu^6\mu^6_1A_8 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_7\mu^7_0A_7 & A_7\mu^7_0\mu^1_1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7\mu^7_0A_7 & \mu^7\mu^7_0\mu^1_1 & \mu^7\mu^7_1A_9 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_8\mu^8_0A_8 & A_8\mu^8_0\mu^1_1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^8\mu^8_0A_8 & \mu^8\mu^8_0\mu^1_1
 \end{pmatrix}$$

Рисунок 4 – Задание алгоритма в матрично-предикатном виде (модульный вариант)  
 Figure 4 - Setting the algorithm in a matrix-predicate form (modular version)

Переставим в матрице  $\mathbf{M}^A$  функциональные и предикативные вершины алгоритма так, как приведено на Рисунке (Рисунок 5). Так как при построении матрицы  $\mathbf{M}^{A*}$  используются одни и те же истинные значения предикатов, то они будут описывать один и тот же граф.

Такое представление алгоритма будем называть функционально-предикативным в матрично-предикатной форме. Матрицу, заданную таким способом, легко представить в виде четырех частей:

$$\mathbf{M}^{A*} = \begin{vmatrix}
 \mathbf{M}^A_{Д} & \mathbf{M}^A_{Д-Л} \\
 \mathbf{M}^A_{Д-Л} & \mathbf{M}^A_{Л}
 \end{vmatrix}$$

- где  $\mathbf{M}^A_{Д}$  – Матрица блоков действия,
- $\mathbf{M}^A_{Л}$  – Матрица блоков логики,
- $\mathbf{M}^A_{Д-Л}$  – Матрица переходов от блоков действия – к блокам логики,
- $\mathbf{M}^A_{Д-Л}$  – Матрица переходов от блоков логики – к блокам действия.

$$M^A = \begin{pmatrix} A_0 \mu^0 A_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_0 \mu^0 \mu^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 \mu^1 A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1 \mu^1 \mu^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_2 \mu^2 A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 \mu^2 \mu^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_3 \mu^3 A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_3 \mu^3 \mu^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_4 \mu^4 A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4 \mu^4 \mu^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_5 \mu^5 A_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5 \mu^5 \mu^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_6 \mu^6 A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_6 \mu^6 \mu^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_7 \mu^7 A_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_7 \mu^7 \mu^7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_8 \mu^8 A_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_8 \mu^8 \mu^8 \\ \hline \mu^0 \mu^0 A_0 & \mu^1 \mu^1 A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^0 \mu^0 \mu^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu^1 \mu^1 A_1 & \mu^2 \mu^2 A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^1 \mu^1 \mu^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu^2 \mu^2 A_2 & \mu^3 \mu^3 A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^2 \mu^2 \mu^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu^3 \mu^3 A_3 & \mu^4 \mu^4 A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^3 \mu^3 \mu^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^4 \mu^4 A_4 & \mu^5 \mu^5 A_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^4 \mu^4 \mu^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu^5 \mu^5 A_5 & \mu^6 \mu^6 A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^5 \mu^5 \mu^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu^6 \mu^6 A_6 & \mu^7 \mu^7 A_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^6 \mu^6 \mu^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7 \mu^7 A_7 & \mu^8 \mu^8 A_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7 \mu^7 \mu^7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^8 \mu^8 A_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^8 \mu^8 \mu^8 & 0 \end{pmatrix}$$

Рисунок 5 – Задание алгоритма в матрично-предикатном виде (функционально-предикативный вариант)  
 Figure 5 - Setting the algorithm in a matrix-predicate form (functional-predictive option)

**Представление алгоритма в виде таблицы.** Квадратная матрица, задающая алгоритм в матрично-предикатном виде имеет недостаток: большое количество нулевых элементов, например, квадратная матрица (Рисунок 5) содержит 324 элемента, а значащих (ненулевых) всего 47. Для уплотнения информации при задании алгоритма в матрично-предикатном виде будем использовать таблицу составляемую следующим образом.

Каждая строка представляет собой истинные значения предиката. Первое и последнее место каждого значения предиката (каждой строки) определяют связи предиката между собой.

№ строки	1-е место	2-е место	3-е место	4-е место	5-е место	6-е место	7-е место	8-е место	9-е место	10-е место	11-е место	12-е место	К строке
1													1
2													3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
<i>n</i>													12

Три места предиката, определяющие процесс функционирования
 
 Места предиката, определяющие параметры функционирующего компонента

Рисунок 6 – Таблично-предикатное представление алгоритма  
 Figure 6 - Table predicate representation of the algorithm

### Заключение

Задание алгоритмов СУ сложными процессами в матрично-предикатном или таблично-предикатном виде даёт возможность использовать хорошо изученные методы теории графов [13], теории матриц [14, 15] и методы теории предикатов. Кроме того появляется возможность использовать при работе с алгоритмами теоретико-множественные и алгебраические операции, разработанные для графов.

### ЛИТЕРАТУРА

1. Дубов В.М., Капустянская Т.И. и др. *Проблематика сложных систем (концептуальные основы модельных представлений)*. СПб.: Элмор. 2006:184.
2. Kron G. *Diakoptics; piecewise solution of large-scale systems*. N.Y.: General Electric Co.1957.
3. Лисицин А.Л., Зотов И.В. Особенности автоматизации управления сложными системами с использованием систем логического управления. *Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение*. 2016;2(19):35-38.
4. Gabriel Kron *Tensor analysis of networks*. London: MacDonald. 1965.
5. Орешкин С.А., Спесивцев А.В., Дайманд И.Н. и др. Синтез интеллектуальных автоматизированных систем управления сложными ТП. *Автоматизация в промышленности*. 2013;7:3-9.
6. Кононюк А. Е. *Дискретно-непрерывная математика. (Начала)*. К.: Освіта України. 2014: 560.
7. John E Hopcroft; Rajeev Motwani; Jeffrey D Ullman. *Introduction to automata theory, languages, and computation*. London: Addison-Wesley. 2001:537.
8. Гуц А. К. *Математическая логика и теория алгоритмов*. М.: Либроком. 2009:234.
9. Зинкина Н. С. Методы и модели логического управления дискретными процессами в распределенных вычислительных системах на основе концепции согласования. *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2011;1(17):35-47.
10. Поляков В.С., Поляков С.В. Запись алгоритма матрицей инцидентора. *Инновации на основе информационных и коммуникационных технологий. Инфо 2014: матер. XI междунар. научн.-практ. Конф.(г. Сочи, 1–10 окт. 2014)*. 2014:149-152.
11. Поляков В.С., Поляков С.В. Представление формального описания функционирования механизмов судоходного шлюза в матрично-предикатной форме. *Молодой ученый*. 2017;17:69-75.
12. Поляков В.С, Поляков С.В. Представление алгоритма в матрично-предикатном виде. *European Research*. 2016;2(13):29-35.
13. Berge C. *The theory of graphs and its applications*. N.Y.:John Wiley.1962.
14. Richard Bellman. *Introduction to Matrix Analysis*. New York: McGraw-Hili Book Company. 1970.
15. Смирнов А. В. Сетевая модель для задачи целочисленного сбалансирования четырехмерной матрицы. *Моделирование и анализ информационных систем*.2016; 23(4):466-478.

### REFERENCES

1. Dubov V.M., Kapustyanskaya T.I. i dr. *The problems of complex systems (conceptual foundations of model representations)*. SPb .: Elmore. 2006.

2. Kron G. Diakoptics; piecewise solution of large-scale systems. N.Y.: General Electric Co.1957.
3. Lisicin A.L., Zotov I.V. Features of automation of management of complex systems using logical control systems. *News of Southwestern State University. Series: Management, Computing, Informatics. Medical instrumentation.* 2016;2(19):35-38.
4. Gabriel Kron Tensor analysis of networks. London: MacDonald. 1965.
5. Oreshkin S.A., Spesivcev A.V., Dajmand I.N. i dr. Synthesis of intelligent automated control systems for complex TP. *Automation in industry.* 2013;7:3-9.
6. Kononyuk A. E. *Discrete-continuous mathematics. (Beginnings).* К .: Education of Ukraine. 2014.
7. John E Hopcroft; Rajeev Motwani; Jeffrey D Ullman. *Introduction to automata theory, languages, and computation.* London: Addison-Wesley. 2001.
8. Guc A. K. Mathematical logic and theory of algorithms. М .: Librocom. 2009.
9. Zinkina N. S. Methods and models of logical control of discrete processes in distributed computing systems based on the concept of coordination. *News of higher educational institutions. Volga region. Technical science.* 2011;1(17):35-47.
10. Polyakov V.S., Polyakov S. V. Recording an algorithm with an incident matrix. *Innovation based on information and communication technologies. Info 2014: Mater. XI international scientific-practical. Conf. (Sochi, October 1–10, 2014).* 2014:149-152.
11. Polyakov V.S., Polyakov S. V. Presentation of a formal description of the functioning of the mechanisms of a shipping lock in a matrix-predicate form. *Young scientist.* 2017;17:69-75.
12. Polyakov V.S., Polyakov S. V. Representation of the algorithm in a matrix-predicate form. *European Research.* 2016;2(13):29-35.
13. Berge C. *The theory of graphs and its applications.* N.Y.:John Wiley.1962:320.
14. Richard Bellman. *Introduction to Matrix Analysis.* New York: McGraw-Hili Book Company. 1970.
15. Smirnov A. V. Network model for the integer balancing problem of a four-dimensional matrix. *Modeling and analysis of information systems.* 2016;23(4)466-478.

## ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATIONS ABOUT AUTHORS

**Поляков Владимир Сергеевич**, кандидат технических наук, доцент, доцент кафедры «Вычислительная техника», ФГБОУ ВО «Волгоградский государственный технический университет», Волгоград, Российская Федерация  
email: [vladstrix@mail.ru](mailto:vladstrix@mail.ru)

**Vladimir S. Polyakov**, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Computer Engineering, Volgograd State Technical University, Volgograd, Russian Federation

**Авдеюк Оксана Алексеевна**, кандидат технических наук, доцент, заведующая кафедрой «Вычислительная техника», декан факультета электроники и вычислительной техники, ФГБОУ ВО «Волгоградский государственный технический университет», Волгоград, Российская Федерация.  
email: [oxal2@mail.ru](mailto:oxal2@mail.ru)  
ORCID: [0000-0001-6201-8773](https://orcid.org/0000-0001-6201-8773).

**Oksana A. Avdeyuk**, Candidate of Technical Sciences, Associate Professor, Head of the Department of Computer Engineering, Dean of the Faculty of Electronics and Computer Engineering, Volgograd State Technical University, Volgograd, Russian Federation.

**Наумов Вадим Юрьевич**, кандидат технических наук, доцент, доцент кафедры «Вычислительная техника», ФГБОУ ВО «Волгоградский государственный технический университет», Волгоград, Российская Федерация.

*email:* [naumovvt@inbox.ru](mailto:naumovvt@inbox.ru)

**Vadim Y. Naumov**, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Computer Engineering, Volgograd State Technical University, Volgograd, Russian Federation.

**Королева Ирина Юрьевна**, кандидат технических наук, доцент, доцент кафедры «Вычислительная техника», ФГБОУ ВО «Волгоградский государственный технический университет», Волгоград, Российская Федерация

*email:* [artmd64@rambler.ru](mailto:artmd64@rambler.ru)

**Irina Y. Koroleva**, Candidate of Technical Sciences, Associate Professor, Associate Professor, Department of Computer Engineering, Volgograd State Technical University, Volgograd, Russian Federation.

**Лемешкина Ирина Геннадьевна**, старший преподаватель кафедры «Вычислительная техника», ФГБОУ ВО «Волгоградский государственный технический университет», Волгоград, Российская Федерация.

*email:* [lem1969@yandex.ru](mailto:lem1969@yandex.ru)

**Irina G. Lemeshkina**, Senior Lecturer, Department of Computer Engineering, Volgograd State Technical University, Volgograd, Russian Federation.