



## TOOLS FOR FORMALIZED DESCRIPTION AND TRANSFORMATION OF ALGORITHMS AIMED AT SYNTHESIZING DEVICES OF CONTROL COMPUTING SYSTEMS

<sup>1</sup>Musayev Muhammadjon Usarovich, <sup>2</sup>Abduganiyeva Yulduzoy Shahabidinovna, <sup>2</sup>Miratoyev Zohidjon Mirvaliyevich

Cand. techn. scient, ass. prof. of the Almalyk branch of the Tashkent University state technical University named after Islam Karimov Almalyk city Uzbekistan<sup>1</sup>, Senior lecturer of the Tashkent state technical University named after Islam Karimov Almalyk city. Uzbekistan<sup>2</sup>, Assistant of the Tashkent state technical University. Islam Karimov Almalyk city Uzbekistan<sup>2</sup>  
[asadbek50@mail.ru](mailto:asadbek50@mail.ru)<sup>1</sup>

### ANNOTATION

Download the following main tasks are solved:- development of existing and development of new methods for analysis and synthesis of the logical structure of the FOCS, taking into account the level of structural, technological and element base of computer technology and ensuring an increase in the regularity of the structure of algorithms for solving problems;

- construction and research of formal models of algorithms aimed at solving the problems of choosing and justifying the internal structure of the FOCS, which differs in the maximum degree of its adaptability to the complex of tasks and the flow of requirements.

**Key words:** FOCS, computing system, BR, facilities of computing system.

### INTRODUCTION

Currently, quantitative methods of analysis are widely used in solving problems of optimal work planning and organization of information exchange, as well as communication in information management systems. a distinctive feature of the optimal approach to the problems of building complex control systems is the study and use of the relationship between the stages of information collection, accumulation, processing and technical parameters of the system. The possibility of an analytical solution to this problem has become real thanks to the proposed method of describing the aircraft model as a Queuing system (QMS) and the development of methods for the theory of Markov chains.

One of the main directions of development of computer systems is to create technical and mathematical complexes focused on classes of application, in which the development of computer equipment (CF) should follow the path of increasing consideration of the specifics of the tasks being solved. They must be reflected in decisions of organizational – functional construction all – internal languages and software, and approaches, methods of their development, in which the design (macro synthesis) and design integration of the armed forces should be conducted in unity with the software, languages, algorithms, task and data sets.

To ensure the ability to solve an ever-expanding range of tasks, the structure of the aircraft was improved, its speed and memory capacity increased. Computer-based complexes of computing tools and information processing systems for various purposes began to be created.

However, this rapid development of CF posed a number of difficult problems for developers. Among them, the most important is the problem of communicating a person with a machine and providing simultaneous access to data and computing tools for a large number of subscribers of the system. The solution to this problem determines the choice of parameters of the aircraft and modes of information exchange between the functional devices of the aircraft and modes of information processing. With the modular principle of building an aircraft, issues of standardization of technical solutions for the structure, command systems and mathematical support are significantly simplified, since each module can be built optimally in terms of the functions it performs, and many procedures in the modules can be performed hardware-based.

To weakly developed and not put on a scientific basis the tasks the General problem of design automation entirely suited to our tasks include: creation of a convenient for engineering practice methods of choice for this class of algorithms the model of chains of operators used in the synthesis of the functional units of the armed forces, development of criteria and rational models, suitable for analysis of adaptation entirely to our tasks; and finally, the formalization of the process of constructing the logical structure of the armed forces for the task of synthesis.

## MAIN PART

It follows that the problem of analysis and development of rational methods of building armed forces, are best adapted to the solution of a given set of tasks and requirements, as well as, the development of new methods of analysis and synthesis of functionally oriented computational systems FOCS and SDI, taking into account the peculiarities of algorithms of tasks, and improvement of SDI, planning and resource allocation of FOCS of the requirements of subscribers is relevant.

The purpose of this article is to develop theoretical and methodological apparatus of construction and methods of design automation of functional-centric CS, which incorporate the features of the algorithms of customer service and tasks, and the level of technological base CF and new methods of analysis and synthesis of the functional structure of the armed forces, taking into account the specifics of the algorithms of the tasks and peculiarities of governors of the armed forces, reflecting the essential characteristics of information exchange systems in the actual and future aircraft.

Solve problems. To achieve this goal, the following main tasks are solved:

- development of existing and development of new methods of analysis and synthesis of the logical structure of the FOCS, taking into account the level of structural, technological and element base of computer technology and ensuring an increase in the regularity of the structure of algorithms for solving problems;

- construction and research of formal models of algorithms aimed at solving the problems of choosing and justifying the internal structure of the FOCS, which differs in the maximum degree of its adaptability to the complex of tasks and requirements flow;

- formalization of the process of synthesis of the logical structure of the FOCS and bringing theoretical solutions and conclusions to practically implementable machine methods of macro synthesis of the FOCS structure, calculated ratios and recommendations;

The problem is devoted to the research and development of tools for formalized description and transformation of a given set, the initial class of algorithms focused on the automation of the design of FM CS.

For this purpose, the structural elements of algorithms are classified according to the most significant features for technical applications, operators and relations, divided into two subsets of automatic maps. The analysis of the structure of relations on a set of operators is performed, and the properties of these relations are defined. The invariance of the properties of order relations and functional dependence on the set of operators of algorithm schemes with respect to the improved  $P_1'$ - transformation is proved. A formal definition of the concepts of content and basic routes (BR) on the schemes of algorithms, relations on a set of operators and routes is given. The analysis of the internal language of the aircraft, focused on a set of algorithms for solving problems. The possibility of using basic route systems (SBR) as an internal language of the aircraft is theoretically justified, the principles and method of developing such a command system for specialized aircraft are studied, and the methods of encoding chains of operations and a set of source data and resulting values of individual operators are considered; the most suitable variants of command formats are analyzed and a single mechanism for interpreting commands is developed. several important concepts and results of the usual automata theory are Developed for the case when finite algebras are considered instead of finite automata.

A well-established approach to describing and developing algorithms, as well as separate solution of the task of equivalent transformation of schemes of algorithms, from problems of synthesis of CS, as noted in, do not provide a satisfactory solution to the problem of synthesis entirely agreed in structure with the algorithms of solving tasks.

Usually, the algorithm model is set as a graph  $G(V, F)$ , where  $V$  is the set of vertices  $v_i \in V$ , and  $f$  is a mapping of  $V$  to  $V$ , or a rule that matches the vertices of the same set to each vertex  $v_i$ . On  $V$ , there are sets  $A$  of operators,  $a \in A$ , and a set  $R$  of quantities  $r$ ,  $r \in R$ , of three types  $X, Y$ , and  $Z$ . the Vertices that make up the zero level are fictitious and are sources of input variables (words) and constants  $x$ ,  $x \in X$ , of the algorithm. The output values  $Z$  of elementary operators (or transformations) and conditions  $\alpha$  are either the final results of the implemented algorithm  $G$  of mapping  $A: X \Rightarrow Z$ , and then  $z \in Y$ , i.e.  $Z=Y$ , or the result of intermediate calculations, i.e.  $z \in Z$  and then  $z$  is the input value of the operator  $a_i$  following  $a_j$  or the condition. Therefore, the entire scope of the implemented algorithm  $G$  display is a lot  $A_1 X_1 = X Y Z$  and area values – the set  $Y_1 = Y Y Z$ , i.e.,  $A_1: X_1 \Rightarrow Y_1$ .

In mapping  $f$ , the following two types of relations are distinguished first of all [3]:  $U$  and  $F$ ,  $U$  is a relation defined on the set  $A$  of operators  $a$ ,  $a \in A$ , and called the relation of order or control relations. It sets the sequence of implementing operators according to the rules:

$$U_{a_i} = \begin{cases} \{a_j\} & \text{if } a_i \text{ is related by control to } \{a_j\} \in A \text{ and } a_i \text{ precedes } \{a_j\}. \\ \emptyset & \text{if } a_i \text{ is not related by control to any operator.} \end{cases}$$

Vertices  $\{a_i\} \subset A$  that are not related by control to any other operators form a subset of  $A^{(k)}$  - finite operators; vertices  $\{a_j\} \in A$ , for which  $\bar{U}_{a_j} = \emptyset$ , where  $\bar{U}$  is the inverse of  $U$ , form a subset of  $A^{(n)}$  - initial operators of the algorithm.

$F$  is a relation of argumentative-functional relations, it establishes a correspondence between the set of operators  $A$  and the set  $R = X$  by the following rule:

$$F_{a_i} = \begin{cases} \{r_i\} - \text{if } \{r_i\} \in R \text{ has arguments for } a_j \in A; \\ \{r_k\} - \text{if the result of executing } \{a_j\} \in A \text{ is } \{r_k\} \\ 0 - \text{otherwise.} \end{cases}$$

Sets  $A$  and  $R$  of operators  $a, a \in A$ , and quantities  $r, r \in R$ , with a set of relations  $\{U, F\}$  defined on them, which fully describe algorithms containing all the necessary information for organizing the computational process, we will call the initial model of the problem-solving algorithm or  $A$ -model

$$G = \{A, R; \{U, F\}\} \quad (1)$$

In accordance with two types of relations, the  $A$  – model combines two types of graphs: the functional graph and the transformation control graph. The functional relation  $F$  on the set of operators in the  $a$ -model of the algorithm does not reflect the structure of a specific program, but the structure of the selected method (algorithm) for solving a set of problems. The corresponding graph of the algorithm is represented as a set of functional operators and information links between them. Control relations  $U$  sets the order of execution of functional operators, and the corresponding graph is represented by a set of transition operators, logical conditions, and control relations.

Important in the theory of algorithms, programming practice, and block synthesis of CS, adapted to the structure of implemented algorithms, is the solution of the problem of constructing a  $U$ -relation that is consistent in structure with relation (i.e., matching the functional graph and the transformation control graph). This task is not yet sufficiently formalized and has not found its final solution. The  $a$  - model only reflects the fact of such matching, but it does not contain an answer to the question how this matching is performed and whether the resulting model does not contain an error algorithm when constructing the – relationship.

This model also "masks" the "pure algorithm" – a method for solving problems – and limits the possibility of flexible use of new technical and software-mathematical tools of the armed forces.

Therefore, the task is to develop such means of formalization of algorithms and transition from the original model to one that would not have these disadvantages and would allow solving the issues of coordinating the relationship between the structure of functional relations and formal definition of chains of operations that are suitable for solving a number of applied engineering problems of computer synthesis.

As shown by the analysis of properties, graph schemes of different types of relations, there are a number of prerequisites for obtaining practically useful formalization of algorithms:

- 1) well-known formal rules that can be improved and used for recognizing and enumerating all meaningful routes on graph diagrams;
- 2) there are such graph schemes for which it is possible to develop formal rules for recognizing and enumerating BR on graph schemes of algorithms;

3) using the well – known and improved rules for converting graph schemes, it is possible to construct such a  $G_F$  ( $A_F$  – model) from the original graph scheme  $G$  of the correct algorithm  $A$  ( $a$  – model), for which assumptions 1) and 2) are valid.

**W-model of the algorithm.** Let's start considering the proposed approach to formalization of algorithms by clarifying the meaningful meaning of matching the structure-relation with the structure of the  $F$  - relation.

Let's assume that there are two operators  $a_i$  and  $a_j$  (or blocks of the algorithm  $G_i$  and  $G_j$ ) that are directly related to each other. The  $a_i$  operator or the  $G_j$  block contains  $\{x_j\}$  as arguments results  $\{y'_i\}$  other operator  $a_i$  or block  $G_i$  ( $y'_i = x_j = z_{ij}$ ) so it must be executed later than  $a_i$  or  $G_i$  ( $i < j$ ). Enumerations of such pairs and sets the relation of functional dependence  $F$  on the set of operators  $a$ ,  $a \in A$ , and blocks,  $G_i, G_i \in G$ , without reference to the automata implementing them or the BC device. Therefore, one of the issues of matching the  $U$  – relation with the  $F$  - relation is to identify such chains of  $a_i, a_j, \dots, a_k$  operators that are directly related to information:  $a_i F a_j, a_j F a_q, \dots, a_\xi F a_k$ .

Due to the peculiarities of the functioning, mathematical support and technical devices of CS, the operator  $a_i$  (or block  $G_j$ ) can not always be executed directly after the operator  $a_i$  (or block  $G_i$ ). Between them, some other operators  $a_{i1}, \dots, a_{i\ell}$  (blocks  $G_{i1}, \dots, G_{i\ell}$ ) can be executed, which may or may not be associated with them ( $a_i$  with  $a_j$ ) or ( $G_i$  with  $G_j$ ), but which should ( or can ) be executed in this interval. This situation occurs, for example, with a limited number of simultaneously operating devices, when alternating data processing procedures in control procedures that implement an algorithm for changing the addresses of service disciplines, and so on. Then the set  $U$  of pairs  $\{a_i, a_j\}$ , obtained as a result of setting this order of execution of operators (or blocks), sets the relation  $U$  of the order of sequence. In the chain of  $a_i, a_j, \dots, a_k$  set operators And between neighbors which is the ratio of  $U : a_i U a_j, \dots, a_\xi U a_k$  the operator  $a_k$  should be much later operator  $a_j$ , although not necessarily to the information and depended on  $a_i$ , however, must not be the consistency of  $U$  – relations  $F$  – ratio. This means that if the operator  $a_j$  information dependent on the  $a_i$ , then  $a_j$  cannot be executed earlier than  $a_j$ .

Based on the sense of consistency of the  $U$  – relation to the  $F$  – relation, rules for establishing the order of implementation of operators should be developed. The following options are offered.

**Rule 1.1.** If  $a_i, a_j, \dots, a_k$  are information related:

$a_i F a_j, a_j F a_q, \dots, a_\xi F a_k$  then the order  $U$  must be determined according to the relation  $F$ :

$$a_i F a_j \vee \dots \vee a_i F a_k \Rightarrow a_i \cup^{(l)} a_j \vee \dots \vee a_i \cup^{(n)} a_k \quad (2)$$

where  $\xi$  and  $k > i$ .

**Rule 1.2.** If  $a_i, a_j, \dots, a_k$  are not related, then the order  $U$  is indifferent from the point of view of  $F$  and must be determined taking into account:

- restrictions on the time of implementation of operators by CS;

- functional specialization of CS;
- status of the CS.

Therefore, to solve the problem of building consistency of a  $U$  - relation by the structure of a  $C$  - relation, one must:

- find a directly information-related chain of operators;
- define linear sections of directly related operators;
- construct a matrix of the relation  $F'$  on a set of linear sections.
- according to the formulated (1.1 and 1.2) rules and taking into account the  $F'$  - relation, construct the relation  $U$  on the set of these operators and linear sections of the algorithm.

For further discussion, a number of definitions are used as the basis for the formalization of the development task, procedures for searching for information-related chains of operators and linear sections on them.

In accordance with three subsets of  $X$ ,  $Y$ ,  $Z$ , and  $R$  values and two types of relations on sets  $A$  and  $R$ , we will distinguish two types of arcs on graph schemes of the algorithm: functional and control.

**Definition 1.1.** arcs are called functional If they connect vertices  $a_i$  and  $a_j$  only if the result obtained after the operation of the  $a_i$  operator is one of the arguments of the  $a_j$  operator.

**Definition 1.2.** arcs are called control Arcs if they specify the order of operators and connect the vertices  $a_i$  and  $a_j$  only if there are conditions when the  $a_i$  operator is followed by the  $a_j$  operator.

**Definition 1.3.** the Route on the graph scheme  $G$  of the algorithm between the vertices  $v_i$  and  $v_j$  is a sequence of vertices  $v_i, v_{i1}, \dots, v_{im}, v_j$  such that the relation is fulfilled:  $v_i F v_{i1}, v_{i1} F v_{i2}, \dots, v_{im} F v_j$ , where  $F$  is in General some binary relation on the set  $v, v \in V$ , including the relation  $U$  or  $F$ .

In the future, we will be interested in the chains on the set obtained by establishing the relation  $f$  on it. To designate a certain route will  $w_i$  and write it in the form:

$$w_i = (a_{i1}, \dots, a_{ij}, \dots, a_{n1}).$$

**Definition 1.4.** the route  $w_i = (a_i, a_{i1}, a_{i2}, \dots, a_{in}, a_j)$  between the operators  $a_i$  and  $a_j$  of the graph scheme  $G$  is called content if:

- operator  $a_i$  - initial,  $a_i \in A^{(H)}$ , and operator  $a_j$  - final,  $a_j \in (A^{(K)} \vee A/A^{(H)})$ - the result of implementing the chain  $a_i, \dots, a_j$  is the output vector  $Y$  of the algorithm, or it is one of the vertices from its components  $y_i, y_i \in Y$ .

**Definition 1.5.** a Vertex operator (or logical condition)  $a_j^{(c)}$  is an Assembly if its arguments  $\{x_j\}$  are the results of  $\{y_j\}$  implementing two or more  $\{a_i\}$  operators, i.e.  $F\{a_i\} = a_j$

**Definition 1.6.** a Vertex operator (or logical condition)  $a_i^{(B)}$  is a branch if its result  $y_j$  is an argument  $x_j$  of two or more  $\{ a_j \}$  operators, i.e.  $F\{a_i\} = \{ a_j \}$ .

**Definition 1.7.** Routes on the algorithm diagram between two vertex operators belonging to branches, assemblies, and a subset of initial  $A^{(H)}$  or final  $A^{(K)}$  operators that are elementary paths or loops are called basic routes of the algorithm, which generally satisfy the conditions:

$$a_i \in A^{(H)} \vee A^{(B)} \vee A^{(C)}; \quad a_j \in A^{(B)} \vee A^{(C)} \vee A^{(K)} \quad i, j \in I$$

$$a_{j\xi} \in A/A^{(H)} \vee A^{(B)} \vee A^{(C)} \vee A^{(K)}, \quad \xi = 1, 2, \dots, n_i$$

where  $I$  is the set of operator numbers on the algorithm diagram.

Loops and paths are called elementary if all their operator vertexes are different (except for the beginning and ending of the loop). A loop is a finite path  $w = (a_i, \dots, a_j)$  whose starting vertex  $a_i$  coincides with the ending  $a_j$ .

If we consider BM as a new operator, and if we know the procedure for constructing the relation  $F'$  on the set  $W$  of BR, then by building the relation of order  $U'$  on the same set  $W$ , consistent with the  $F$  - relation, we come to the following model of the  $G^*$ algorithm.

$$G^* = \{ W; R', F', U' \} \quad (3)$$

Which we call the  $W$  - model, where  $W$ -sets of basic routes  $w_i = (a_{i_1}, \dots, a_{i_2}, \dots, a_{i_k})$ ; argument-functional belonging on the set  $W$  of the BR;  $U'$  is the relation of order (control relation) on  $W$ .

## CONCLUSION

Within the framework of solving the problem of macro synthesis of CS, the application of the proposed formalization of the description of algorithms. Algorithms is advisable for the following reasons:

- in the synthesis of functional devices of the aircraft, the use of BR as generalized operations of the command system will simplify the system of addressing operands, both for single-module and multi-module memory, and the control automata implementing it, both for centralized and distributed organization of the aircraft resources;

- when designing the algorithm, it is not necessary to develop the structure of the relationship of order  $U'$ , since control relationships are established directly in the process of implementing the algorithm, according to the structure  $F'$  and in accordance with the States of resources (technical and software).;

- using chains of directly information-related operations as generalized operations, commands and encoding them in a single command increases the efficiency of the resulting computer command system in terms of reducing the length of programs and memory capacity, the frequency of accessing memory for commands and values, and the time to implement algorithms.

## LITERATURE

1. Вальковский В. А. Распараллеливание алгоритмов и программ. Структурный подход. -М.: Радио и связь. 1989 , -328page.
2. Геллер Д.А., Журавлев Ю.П. Основы логического проектирования ЦВМ. –М.: Сов радио, 1967. -286page.
3. Верма П. Сети связи ЭВМ. Оценка эффективности функционирования: Структурный анализ. Пер. с англ. –М: Радио и связь, 1992г. 113page
4. Глушков В.М. Синтез цифровых автоматов. –М.: Физматгиз, 1962. – 476
5. Гудман С., Хидетниемеи С. Введение в разработку и анализ алгоритмов. -М.: Мир. 1981.275page
6. Мусаев М.У. Средства формализованного описания и преобразования автоматов и алгоритмов, ориентированных на проектирования ВС / Таш.политех.ин-т.-Ташкент.1988. -29page. 27.12.88. - 29page. № 911 Uzb 88.
7. Qushimov, B., Ganiev, I. M., Rustamova, I., Haitov, B., & Islam, K. R. (2007). Land degradation by agricultural activities in Central Asia. *Climate Change and Terrestrial Carbon Sequestration in Central Asia*; Lal, R., Suleimenov, M., Stewart, BA, Hansen, DO, Doraiswamy, P., Eds, 137-146.
8. Sanaev, G., Kim, K. R., Hasanov, S., & Ganiev, I. (2015). Review of postharvest aspects of fruits and vegetables subsector in Uzbekistan. *Connecting local and global food for sustainable solutions in public food procurement*, 14, 557.

