ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 6 , June 2020

The Model of the Algorithm of Functioning of the Multi-Module Computer System

MUSAYEV MUKHAMMADJON USAROVICH,KHUJAEV TUYMUROD KHUDDIYEVICH, MAMASODIKOV XUMOYUN NASIVALI O'GLI

Associate professor of the department "Mathematics and natural science disciplines" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;
Senior lecturer of the department "Mathematics and natural science disciplines" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;
Student of the department "Engineering Technology" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;

ABSTRACT:A formal model of the algorithm for the functioning of a multi-module aircraft is proposed. Several important concepts and results of ordinary automata theory are developed for the case when finite algebras are considered instead of finite automata. The model differs from the known ones not only in the blockiness of data processing, but also in the fact that it is based on two sets of automatic mappings: a set of partial mappings (transformations) of input words into output ones, and a set of transformation control operators. Due to the coincidence of the principles of operation of the described modular aircraft and real machines, it can be expected that the concepts introduced and the results obtained will be easier to apply for programming calculation and analysis of algorithms.

KEYWORDS: multi-module VS, partial mappings, automaton, set of States, transition function, input function, output function, clock cycles.

I. INTRODUCTION

In the well-known works [1-5] different authors call a finite automaton four A = { Q,δ,q_0, Q_F} [1], five A = { Q, ∑, δ,q_0,Q_F} [2], and so on up to eight [3] ordered characters. The most widely used definition describes the automaton as six A = { Q, ∑, Δ, δ, λ, q_0,}, where Q, ∑, Δ, (sets of States, inputs and outputs, respectively) are finite non - empty sets; δ (transition function) is the mapping Q x ∑ to Q ; λ (output function) is the mapping; Q x ∑ to Δ; q_0 is the initial state, which is an element of Q, $q_0 \in Q$. the function δ can be extended to the function of transitions from the set Σ* (the set of finite chains or words, on the set of elements of the input alphabet Σ, including the empty chain Λ) to Q by means of the equations δ(Λ)= q_0,) and δ(σ, x) = δ(σ, δ(x)) for all σ ∈∑. The function λ extends to the map Q x Σ* in Δ *, where Δ * is the set of all words in the alphabet Δ.

Problem statement. The structure of automaton A resembles a finite abstract algebra [1,4,5] - it is a finite set ΣUΔ together with operations f_1,..., f_Aна it. This representation is generalized for the case of a block assignment of an automaton as a Cartesian product of two sets of automata maps; the set of transformations of input words to output and on the set of transformation control operators. The first set is used to describe the law of transformation Y=F(X) of the input vector into an output or a class of algorithms of problems to be solved on a computer system, and the second set is used to describe the control of the process of implementation of these algorithms on a given set of functional modules of a computer system with a given architecture.

II. METHODOLOGY

1.Meaningful description of the structure of the law of transformation Y=F(X)

The analysis of the model. From the above it follows that given the structure of the law of transformation is to determine the composition of partial maps (operators) and the structure of the relationship between them (or more generally the ratio of the quasiorder on the set of partial maps).

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 6 , June 2020

The structure of the connection determines the structure of the function of transformation of the input vector X into control actions u, i.e. the structure of the first set of automata maps, which should be considered as a regular expression [6] of the automaton a, if the input vector X is taken as input data (States), as links Z - the results of intermediate transformations (calculations), and the final data (States) that do not have links to the output, is presented as components of the output vector Y. This function is a set-theoretic and algebraic operations on functions describing the behavior of local abstract sub – tomatoes-partial maps.

n General, it is difficult to determine the structure of a function describing an abstract automaton with an arbitrary matrix of relations, although possible, because of the cumbersome expressions. However, for some special cases of operations on sub-automata (combinations of which allow us to obtain a description of any automaton) we can find quite simple expressions [6]. Using the properties of these operations on abstract automata, described in detail [7-10], it is possible to determine the structure of the law of transformation of the input vector into control actions

Y, Y= F (X) (1).

However, let us use the heuristic method already used in section 1.3. in the General description of the mapping algorithm.

First, replace transformation (1) with two equivalent transformations:

-transformation $F_{1,i}, i \in J_1 = \{1,\dots,n_1\}; \quad Z_{1,i} = F_{1,i}(x_i)$ (2)

over a subset $X_i \leq X$, where $X = X_1 \times \dots\dots \times X_{n1}$;

- and transformation. $F_1 Y = F_1(Z_1)$ (3)

where $z_1 \in Z = \bigotimes_{i=1}^{n_1} z_{1,i}$, F_1 - display a subset of the first stage of the partition function F.

n this case, the conversion (1) is performed in two steps, and for it to be equivalent to the conversion (2) and (3), the ratio must be performed

$$F = F_1 \left(\boxdot_{i=1}^{n_1} F_{1,i} \right) \quad (4)$$

where $\boxdot$ -some operation on partial maps (or operators) $F_{(1,\,i),}$ i $\in J_1$, first stage of partition F.

The function F_1 can also be implemented by a set of partial mappings

$F_{2j}, j \in J_2 = \{1,\dots,n_r\}$ (or partial operators) and is represented as :

$$F_1 = F_2\left[\boxdot_{j=1}^{n_2} F_{2,j}, \left(\Phi_{1,\xi} \subseteq F_{1,o}\right] \right. \quad (5)$$

In General, the mapping (s-1) of the partition stage will be:

$$F_{s-1} = F_s\left[\boxdot_{k=1}^{n_s} F_{s,k}, \left(\Phi_{s-1,\xi} \subseteq F_{s-1,o}\right] \right. \quad (6)$$

where F_s is the partial display (partial operators) of the s - th step of splitting of the function F.

Sequence of sets of partial maps $F_{s,.} = \left\{F_1^{(s)}, \dots \dots F_{n_s}^{(s)}\right\}$ for all = 1,...., m and the relationships between them form the structure of the law of development of control actions *yY* (control law), which we denote by

$$G = (\{X_{s-1,.}\}), \{Z_s^{(s)},.\}, F, U) \quad (7)$$

where $X_{s-1,o} = \{X_{s-1,k}\}$ - set of input words of partial maps (operators)

$\{F_{s,k}\}$ S - stages of breaking;

$Z_s^{(s)},_o = \{Z_{s,k}\}$ – many intermediate results of partial maps $\{F_{s,k}\}$ s-th stage of partition;

$F = \{F_1,\dots,F_m,.\}$ – a set of mappings at different levels of partitioning;

U – many relationships between partial maps.

The questions of the depth of division of the function F into a set of partial maps should be solved with the problem of aircraft synthesis, which in modern conditions should be formalized as a problem of covering the algorithm of aircraft functioning with a set of standard, functional modules. The set of functional modules must be optimal from

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 7, Issue 6 , June 2020

the point of view of functional matching and a variety of limiting factors: a temporary, energy costs, material and other means.

Thus, the described structure of the first set of automata maps as a set of blocks of the algorithm for converting the input vector X into the control action Y and the communication network between them U (partial maps or operators), which is free from the restrictions associated with the technical implementation of hardware and software of the computer system.

2. Formal setting of the transformation algorithm F: X→ Y.

For further discussion we introduce sequentially numbered partial mappings

$$F_{sk}\ (s = 1,\dots,m), k = 1,\dots\dots n_s), I.e.\ \ F_i, i = 1,\dots,N', \text{where } N' = \Sigma n_s.$$

Moreover, we assume that the sets N={1,...,N} denotes the sequence numbers of the algorithm blocks – partial maps F_i:$X_i \rightarrow Z_i$.

Therefore, each pair (s,k) representing or a block is put in correspondence with some number i, so in the future i will be understood as a partial map $F_i, i \in N$, if this is not in doubt. Each F_ic has its own set of input $X_{i-1}$and output Y_i words From the definition and value areas of the algorithm, respectively, as well as the results of Z_i, calculations distributed over the first two areas.

n this case, to study the algorithm of transformation F: X →Y, you can build a model in the form of a directed graph, the vertex a of which corresponds to individual operators (or partial transformations F_i), the arc Z operands.

Definitions and designations

Similarly [11] we say that the graph of the algorith$G = (A, Z; \mathbb{D})$ if two sets of elements are specified, this is the set $A \neq \acute{O}$ operator vertices $a_i, a_i \in A,$ and many $Z\ (A \cap Z = \acute{O})$ edges of quantities, as well as a triple predicate D satisfying the following conditions: $\mathbb{D}$ – defined on all ordered triples of elements a_i, z, a_j, where

$$a_i, a_j \in A, \qquad z \in Z;$$

$$\forall z \in Z \exists\, a_i, a_j \in A \{\mathbb{D}(a_i, z, a_j) \wedge \forall a_i', a_j' \in A[\mathbb{D}(a_i, z, a_j) \Rightarrow \Rightarrow (a_i = a_i' \wedge a_j' = a_j') \wedge (a_i' = a_j' \wedge a_j = a_i')]\}$$

Triple predicate called incident that enumerates all the edge values connecting the adjacent vertices of the operators. It defines a binary relation $\mathbb{D}$ defined on the set a and formed by the Union of the two relations, first, the relation with "to be a contour element" having transitivity properties

(if$a_i C a_j$ и $a_j C a_k$)

and installed only within some subsets $A_i \subseteq$A (narrowed with respect to their area),and, second, the ratio of the**Γ** strict order, with the properties of ant reflexive (if$a_i \Gamma a_j$ *and* $a_j \Gamma a_k$that follows $a_i \Gamma a_k$) and set on the set $A \setminus \cup A_i$_i as well as between subsets $A_i \subseteq A$.

The ratio D indicates that if $a_j \in \mathbb{D}a_j$, then a_j is connected with I edge by the value Z, a_i.e. $\mathbb{D}(a_i, z, a_j)$

A. therefore, D is such an order relation that is defined on A if$\forall a_i \in$A [$a_i \neq \acute{O}$ **],** and me Z, if$\forall a_j \in \mathbb{D}a_i \exists z \in$**Z [**$\mathbb{D}a_i \neq \acute{O}$ =>$\mathbb{D}_1$ ($\mathbb{D}_1 a_i$))]. Therefore, $\forall a_i \in A$ **[**$\mathbb{D}a_i \neq \acute{O}$ =>$\mathbb{D}_1$ ($\mathbb{D}_2 a_i$)] ,where $\mathbb{D}_2 a_i$ = {$Z_{ij,}, j = 1,2,\dots$ $\mathbb{D}_1 a_i | i = f_{iX}\}$- sets of values passed to each $a_j \mathbb{D}a_i$, otherwisethe relation $\mathbb{D}$ establishes a direct relationship $a_i \rightarrow z_{ij} \rightarrow \{a_j\}$between any operators $a_i \in A$.

- In General, the algorithm scheme has a complex structure. The vertices that make up the zero tier are fictitious and are sources of input variables (words) x ,x∈ x of the algorithm.

The output signals Z of the elementary operators a_i(or transformations, which are called partial representations of some information objects $x \in X$the other in$y \in Y$) and conditions is or final result of the implemented algorithm G mapping $X \rightarrow Y$. Then z∈Y, i.e. z=y, or the result of intermediate calculations, i.e., z∈Z, then z is input as the

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 6 , June 2020

nexta_i Haparandaa_j or conditions. Therefore, the predicate $\mathbb{D}$ of the algorithm G is the ratio of the order on the set A of partial maps (elementary operators and conditions representing elements of some algebra), the domain of which is not only X, but also Z. Therefore, the entire domain of the definition of the implemented algorithm G of the map A_1, will be the set $X_1 = X \cup Z$, and the domain of values will be the set $Y_1 = Y \cup Z$, i.e. $A_1: X \to Y$..

In this case, you can maintain the concepts of input and output of the algorithm

Definition 1.1. The output (combined) of the algorithm G is called the tuple y', belonging to the information set Y_1 obtained by combining the images $\{A_1(x')\}$ of all elements x', input to the set X_1. Let's call it a complete output in comparison with the output tuple y, which belongs only to the set Y, i.e.

y∈Y. Obviously, $\forall y \in Y$ let us assume that $y \in Y_1$, as $Y \subseteq Y_1$, but not $\forall y' \in Y$ true y' ∈Y, but only $\forall y' \in Y \exists y \in Y \{y' = y \Rightarrow y' \in Y\}$.

Definition 1.1.a. The output tuples of y and z of the algorithm G_l and G_γ are identical if a one-to-one correspondence $\forall y' \in Y \exists z_k \in Z\{y_i \leftrightarrow z_k \Rightarrow y \equiv z\}$ can be established between the numbering of their members}

Definition 1.2. The input(Union) of the algorithm G_1 is a tuple x, belonging to the information set x_1,, obtained by combining the prototypes $\{A^{-1}(y')\}$ of all elements y', included in the set Y_1..

Similarly to definition 1.1a , we formulate

Definition 1.2.a. The input tuples x and χ algorithms G_l and G_j are identical, if the numbering of their members can establish a one-to-one correspondence $\forall x_i \in X \exists x_k \in \chi \{x_i \leftrightarrow \chi_k \Rightarrow x \equiv \chi\}$.Thus, in General, the model of the algorithm is such a triple G =(R, A; D), in which R is the combined set of information objects ,$r \in R = X \cup Z \cup Y$; $A_1: X_1 \to Y_1$ – is the set of partial maps (the set of operators and conditions) that are elements of some σ-subalgebra of subsets of the set R, and at the end, $\mathbb{D}$A → A, $\mathbb{D} \subseteq$ AxA is the binary relation of the quasi-order on the set A, σ-algebra is the set A, which is the set of algebraic expressions composed of the set R, in this case, the set a of algebraic expressions is closed with respect to the operations of unions, enumeration and complement.

Note 1. The model of the algorithm is given by two kinds of relations on two sets - this is the relation A on the set R and the relations $\mathbb{D}$ on the set A. in accordance with them, we will distinguish two types of relations (arcs) on the scheme (graph) of the algorithm : functional and control.

Definition 1.3. Relationships are called functional if they bind the a_i anda_j operators only when the result obtained after the a_i operator is one of the arguments of the a_j operator.

Definition 1.3.a. Connection called control, if they set the order of the operators and associated operator $a_i$and a_j, only in the case when the conditions under which after the operator should the operator a_i and a_j.

3. The structure of the law of functioning of computer systems

Individual fragments of the algorithm are performed by certain functional modules or sets of modules of computing systems.Therefore, each block or operator a_ $a_i \in$ A (partial mapping F_s or even F_{sk} $(s = 1,\ldots,m;\ k = 1,\ldots,n)$ from section 2 can be put in accordance with some automaton A. However, such a trivial comparison of elements of the scheme (model) of the algorithm of elements of the block diagram of computer systems is not the best in terms of equipment, although in this case on the basis of the algorithm F: X→Y it is easy to build an algorithm for the functioning of computer systems, which is obtained from the first by introducing the simplest mechanism of synchronization of interacting processes, which is reduced only to the expectation of the latest of the source data (this explains the prostate).. Taking into account the limitations on the equipment and the implementation time of partial maps, the algorithm of multi-modular computing systems can be set as follows.

Let given set N = {1,...,N} denoting the blocks of the algorithm for converting the input vector X into control actions, i.e. maps $F_i: x_i \to y_i,\ i \in N,$, and the order from the sequence are given. Let also given a set M={1,...,M}, indicating the functional modules of the computing systems and $1 \le M \le N$ we will Build the algorithm of functioning of the multi-module computing systems. To do this, we will use the system of features describing the algorithm of algorithmic objects and the relations between them introduced in [12].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 6 , June 2020

4. The algorithm of functioning of a multi-modular computer system.

When specifying algorithmic objects, it was noted that the algorithm model can contain a number of similar fragments and their implementation may require several identical functional modules in the scheme of the computer system. Taking into account the above and formalization from [13] at each time there will be a function f: N→M, given as follows :

$$f[\{i_\sigma\}\{x_i,y_j\}] = \begin{cases} \{k_\lambda\},\{x_k,y_k\} - \text{if partial mapping;} \\ \{i_\sigma\} - \text{and initial data}(x_i, y_i)\text{is in the queue}\{k_\lambda\}\{x_i,y_i\}; \\ \{k_\xi\} - \text{if } \{i_\sigma\}\text{is a service } \{k_\xi\}; \end{cases}$$

Otherwise, it is uncertain.

This function is called "algorithm block map". It reflects the real situation of placing the operators and the initial data of the queue and the devices (blocks) of the computer system that are under maintenance. Its structure can be divided into three parts. The first two parts are the "queue map", algorithm blocks and constant arrays (the latter describe the placement of input and output data). They form the basis of the mechanism of synchronization of interacting processes. The third part of the function f is the "implementation map" of partial maps by functional devices of the computer system.

f the partial map F_i((block of algorithm G_i) contains n_i operators (let it be block size) and each operator is placed in one memory cell, then the really executed operator i (virtual address) is an integer α (0≤ α ≤ nni). Similarly, the really working element (components of the module), or rather, the element implementing the partial mapping operator, is β(0 ≤ β ≤ mni) .

The mechanism for establishing the correspondence between the blocks of the algorithm and modules of the computing system (device address translation, establishing the nature of the units) having the address α is indeed performed by the operator, finds the pair пару (i_σ, w) such that $\alpha = (i_\sigma - 1)n_i + w)$, where $0 \le w < n_i$ and then generates the address of the module components of the computing system $\beta = [f(i_\sigma) - 1]n_i + w$, if the value of $f(i_\sigma)$ is defined; otherwise, produces a "sign of the employment module," computing systems, if the value of $f(i_\sigma)$ is not defined. On the basis of the module occupancy, the execution of the algorithm fragment is stopped for the time of "waiting for the module" until the module is free, selects the corresponding partial display with arrays of numbers and changes the necessary maps in accordance with the change in the state of the computer system.

All sequences (chains) of blocks G of algorithm G of length N over n, where k ≥ 0, are induced from the binary relation of the quasiorder phases on set Φ . They can be found on the set N^k. In the process of the algorithm generates a sequence of calls not only m to partial maps of the algorithm $w_T = r_1,\ldots,r_t,\ldots,r_T$, which is an element of a subset of N^T for some T of the set N^k, as well as to arrays of numbers $z' = z'_1,\ldots,z'_T$. If $r_t = x$, then x means that the t-th inversion of the algorithm or the inversion at the t-th moment is the inversion to the partial map of x.

We draw an analogy with the classical models of the functioning of the computer system. Let's call S, S⊆N - the state of the computer system. Let Ω be the set {S |S⊆N and| S | ≤ m of States S , where $|S|$denotes the number of elements in the set S , including similar elements.

Definition 1.4. Let M = [1,..., m] and N = [1,...,n]are given.

Block algorithm of the computing system for M and N is a system $A = (\mathcal{R}xQ, N, Z, \delta, \lambda, q_0, Q_T)$, in which

Q-set of control States of the algorithm

1) 1. Q-set of control States of the algorithm $(Q = N^m, q = (r_1,\ldots,r_m))$; ;

2) N-finite non-empty set of input characters;

3) Z-finite non-empty set of outputs;

4) δ:$\mathcal{R}$ x QxN → R-load conversion of the functional modules of the computing system. It has the property that X belongs to S 'whenever δ (S, q, X) = (S', q');

5) λ: $\mathcal{R}$ x $QxN \rightarrow Z$ -mapping of the set $\mathcal{R}xQxN$ in the set of all words from the alphabet R ;

6) $q_0, q_0 \in Q$ -initial control state;

7) $Q_T, Q_T \in$Q set of final States.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 6 , June 2020

Note 2. The block algorithm is nothing more than an automaton with States $\mathcal{R}$xQ, inputs N and transition functions δ and λ.

Definition 1.5. The chain of the set N* $\mathcal{R}$xQ) N* is called the configuration of the block automaton $A = (\mathcal{R}xQ, N, Z, \delta, \lambda, q_0, Q_T)$. A configuration where $1 \leq i \leq m$ for each $r_i \in N, q_0 \in Q_T, S \in \mathcal{R}$ is interpreted as an indication that the block automaton, while in the state (S,q) observes (implements) i - partial mapping from the i chain $r_i, \ldots, r_m$.

Let for automaton $A = (\mathcal{R}xQ, N, Z, \delta, \lambda, q_0, Q_T)$ on configurations $N^* (\mathcal{R} xQ) N^*$ the relation ├ * is defined as follows: $r, x \in N, y = N$, then

1) rypax ├rqybx if (q, b, -1) ∈δ (p, a);
2) rypax ├ rqbx , if (q,b ,0)∈δ (p,a);
3) rypax ├ rbqx if (q, b,1) δδ(p ,a).

SUMMARY.

For each $r \in N^* (\mathcal{R} xQ) N^*$ we assume $r \vdash^* r$. For $r, p \in N^* (\mathcal{R} xQ) N^*$ we assume r ├* p if $r = r_0 = \ldots\ldots r_k = p$ such that $r r_i \vdash r_i + 1$ for each $i < k, r \vdash p$ means that the block automaton goes from configuration **r** to configuration **p** in one clock cycle. So, for example, under the condition $(q, b, -1) \in \delta(p, a)$ the entry **Ucpav ⊢ Uqcbv** means that the machine observes block a in the state **p**, goes to the state **q** , takes **a** on **b** and moves the algorithm forward for as many operations as they were occupied by the surveyed block **a**.

Similarly, you can interpret the machine for the case of movement back or standing still. r ├* p means that the machine transitions from the configuration in the configuration **p**for **a** few bars of the work.

REFERENCES.

[1] Yanov, Yu I. logical schemes of algorithms. Sat. "Problems of Cybernetics," vol.1, M. Fizmatgiz, 1958. p. 29-53..
[2] Yanov, Yu I. logical schema transformation algorithms. Sat. "Problems of Cybernetics," vol.20, Fizmatgiz, 1967, p. 255-259.
[3] Krinitskaya N. A. G. A. Mironov, G. D. Frolov Programming and algorithmic languages. M. from-in "Science", 1975 496 p.
[4] Kapitonova Yu. V., Letichevsky A. A., Mathematical theory of computer systems design. M. Science. GL. ed. of physics and Mat.lit., 1988. -296 p.
[5] Mamigonov A. G., In Kulba .Synthesis of optimal modular data processing systems. M. "Nauka", 1986. 280 p.
[6] Melikhov A. N. Oriented graphs and finite automata. M. "Science".1971. 415 PP.
[7] Valkovsky V. A. Parallelization of algorithms and programs. Structural approach.M. "Radio and communication".1989 176 p.
[8] Miller R. theory of switching circuits .Vol.1.Combinational circuit.M., "Nauka", 1970
[9] The Ajzerman M. A. and others Logic. Automata.Algorithms. M. Phys.mate, 1963.
[10] Merenkov N. N. Parallel programming for multimodule aircraft M., "Radio and communication" 1989 320 p.
[11] Samofalov K. G. ,Lutsky G. M. fundamentals of the theory of multi-level conveyer of the armed forces., M., "Radio and communication" 1989 223 p.
[12] Zykov A. A. theory of finite graphs. Novosibirsk," From " Science, 1969 544 p.
[13] Musaev M. U. Means of the formalized description and transformation of the automata and algorithms focused on design of VS/ Tashk. Polytechnic.in-t .- Tashkent, 1988. -29 p. DEP .inUsniate 27 .12. One thousand nine hundred eighty eight, N. 911.

AUTHOR'S BIOGRAPHY

MusayevMukhammadjonUsarovich, Associate professor of the department "Mathematics and natural science disciplines" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 6 , June 2020

КнujaevTuymurodKhuddiyevich, Senior lecturer of the department "Mathematics and natural science disciplines" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;	
MamasodikovXumoyunNasivalio'gli, Student of the department "Mathematics and natural science disciplines" of the Almalyk branch of the Tashkent State Technical University named after Islam Karimov, Almalyk, Uzbekistan;