UDC 517.7
DOI: 10.37069/1683-4720-2019-33-14

©2019. **S.V. Sapunov**

COLLECTIVES OF AUTOMATA ON INFINITE GRID GRAPH WITH DETERMINISTIC VERTEX LABELING

Automata walking on graphs are a mathematical formalization of autonomous mobile agents with limited memory operating in discrete environments. Under this model broad area of studies of the behaviour of automata in finite and infinite labyrinths (a labyrinth is an embedded directed graph of special form) arose and intensively developing. Research in this regard received a wide range of applications, for example, in the problems of image analysis and navigation of mobile robots. Automata operating in labyrinths can distinguish directions, that is, they have a compass. This paper examines vertex labellings of infinite square grid graph thanks to these labellings a finite automaton without a compass can walk along graph in any arbitrary direction. The automaton looking over neighbourhood of the current vertex and may move to some neighbouring vertex selected by its label. We propose a minimal deterministic traversable vertex labelling that satisfies the required property. A labelling is said to be deterministic if all vertices in closed neighbourhood of every vertex have different labels. It is shown that minimal deterministic traversable vertex labelling of square grid graph uses labels of five different types. Minimal deterministic traversable labelling of subgraphs of infinite square grid graph whose degrees are less than four are developed. The key problem for automata and labyrinths is the problem of constructing a finite automaton that traverse a given class of labyrinths. We say that automaton traverse infinite graph if it visits any randomly selected vertex of this graph in a finite time. It is proved that a collective of one automaton and three pebbles can traverse infinite square grid graph with deterministic labelling and any collective with fewer pebbles cannot. We consider pebbles as automata of the simplest form, whose positions are completely determined by the remaining automata of the collective. The results regarding to exploration of an infinite deterministic square grid graph coincide with the results of A.V. Andzhan (Andzans) regarding to traversal of an infinite mosaic labyrinth without holes. It follows from above that infinite grid graph after constructing a minimal traversable deterministic labelling on it and fixing two pairs of opposite directions on it becomes an analogue of an infinite mosaic labyrinth without holes.

MSC: 68R10, 05C85, 68Q45, 68T40.

Keywords: *square grid graph, graph-walking automaton, vertex labelling, collective of automata.*

Introduction

Automata walking on graphs are a mathematical formalization of autonomous mobile agents with limited memory operating in discrete environments. Under this model broad area of studies of the behaviour of automata in finite and infinite labyrinths (a labyrinth is an embedded directed graph of special form) arose and intensively developing [1, 2]. Research in this regard received a wide range of applications, for example, in the problems of image analysis and navigation of mobile robots [3]. The results for automata and labyrinths are based on the important assumption that automata operating in labyrinths can distinguish directions, that is, they have a compass [4, 5]. This paper discusses automata without compass, that is, they do not distinguish between the directions and relative positions of the vertices. Such restriction of capabilities makes the behaviour of automata on the graph much more complicated. For example, the problem of preserving the movement direction on the graph is trivial for an automaton with a compass, but for an automaton without a compass it requires using additional equipments and development of methods for their usage [6]. This poses the problem of enrichment of the graph model (by adding some preferably minimal properties) to ensure that the automaton could move along graph in any arbitrarily chosen direction. The most natural enhancement of a graph is to label its structural elements: vertices, edges, incidentors, etc. The automaton gets an opportunity to read labels in a local neighbourhood of the current vertex and use them for movement. This article deals with the vertex-labelled graphs.

1. Problem formulation.

This paper sets out to consider two related problems.

1) Let some labels (colours) be assigned to the vertices of the square grid graph, and let an automaton be able to receive at the input the label of the current vertex and the labels of all vertices from its neighbourhood. The automaton can move between adjacent vertices by selecting a target vertex by its label. Does there exists a vertex labelling such that using it an automaton can move along a graph in any arbitrarily chosen direction?

2) If the first is possible, then does there exists an automaton that traverse such labelled graph, having the only ability to read the vertex labels from a closed neighbourhood of the current vertex?

In addition to the question of labelling existence, there is the question of minimizing the amount of label types that is also worth studying.

2. Basic definitions.

Let $\mathbb{Z}$ denote the set of integers, and let $\mathbb{N}$ denote the set of natural numbers. We will use the symbol $\mathbb{Z}_n$ to denote the set $\{0, 1, ..., n-1\}$ for any $n \in \mathbb{N}$.

We will use standard terminology for graphs (we refer the reader to [7]).

The path graph P_n is a tree with two nodes of vertex degree 1, and the other $n-2$ nodes of vertex degree 2. A 1-way infinite path graph (or a ray) $P^{+\infty}$ is a graph which isomorphic to the graph with vertex set $\{v_i : i = 1, 2, \dots\}$ and edge set $\{(v_i, v_{i+1}) : i = 1, 2, \dots\}$. A 2-way infinite path graph (or a double ray) P^{∞} is a graph which isomorphic to the graph with vertex set $\{v_i : i = \dots, -2, -1, 0, 1, 2, \dots\}$ and edge set $\{(v_i, v_{i+1}) : i = \dots, -2, -1, 0, 1, 2, \dots\}$. An infinite two-dimensional grid graph G^{∞} is the graph cartesian square of 2-way infinite path graph. A 2-way infinite ladder graph $G^{(\infty,2)}$ is the graph cartesian product of P^{∞} and P_2. A rectangle graph $G^{(n,m)}$ is the graph cartesian product of P_n and P_m.

We will use the embedded square grid graph, which vertices corresponds to the distinct points of the integer lattice $\mathbb{Z}^2$ and two vertices are connected by an edge if and only if the corresponding points are at distance 1. Suppose that the name of the vertex of the embedded graph is the coordinates of the corresponding point on the plane. A half-grid graph is the subgraph of the embedded graph G^{∞} induced by the set of vertices lying on one side from an infinite straight line including vertices that are on the line. The line is called the half-grid boundary. An angle graph is the subgraph of the embedded graph G^∞ induced by the set of vertices that are between two different half-lines sharing a common starting vertex including vertices that are on half lines. Half-lines are called the sides of the angle, and their common vertex is called the vertex of the angle. The value of the angle is the value of the geometric angle between its sides. A stripe graph is the subgraph of the embedded graph G^∞ induced by the set of vertices that are between two parallel lines including vertices that are on lines. The lines are called the borders of the stripe, their direction is called the direction of the stripe, and the distance between them is called the width of the stripe. A half-stripe graph is the subgraph of a stripe graph induced by the part of the stripe vertices that are on one side from a straight line crossing the stripe borders. A segment of this line, enclosed between the borders of the stripe, is called the end of the half-stripe.

A labelled graph is a simple connected vertex-labelled graph $G = (V, E, M, \mu)$, where V is a set of vertices, E is a set of edges, M is a set of labels, $\mu : V \to M$ is a surjective labelling function. The open neighbourhood O_v of a vertex $v \in V$ is the set of all vertices adjacent to v. A neighbourhood in which v itself is included, called the closed neighbourhood and denoted by $O_{(v)}$. A multiset of labels of all vertex from $O_{(v)}$ is called the labelling of vertex v neighbourhood and denote by $\mu\,(O_v)$. A walk in graph G is a series of vertices $p = v_1 \dots v_k$ such that $(v_i, v_{i+1}) \in E$, $i = 1, \dots, k-1$. The positive integer k (the number of vertices) is the length of p. The label $\mu(p)$ of the walk p is a word $w = \mu\,(v_1) \dots \mu\,(v_k)$ in label alphabet M. We say that the word w is defined by the vertex v_1.

A graph-walking automaton on labelled graph G is a sextuple $A = (S, X, Y, s_0, \varphi, \psi)$, where S is a finite set of internal states, $X = \{(a_0, \{a_1, \dots, a_k\}) \,|\, a_i \in M, 0 \leqslant i \leqslant k\}$ is a finite input alphabet (a_0 is a current vertex label, $\{a_1, \dots, a_k\}$ is a set (or multiset) of labels of all vertices on the current neighbourhood, k is a degree of the current vertex), $Y = M$ is a finite output alphabet ($y = a$ means that the automaton moves from the current vertex to an adjacent vertex with the label a), $s_0 \in S$ is the initial state, $\varphi : S \times X \to S$ is a transition function, $\psi : S \times X \to Y$ is an output function. Automaton operates as follows: observes the labelling of current vertex neighbourhood, chooses some label, and moves to the vertex with this label. The automaton does not have a compass, that is, it does not distinguish directions and relative position of vertices. Therefore, it does not distinguish vertices with the same labels. It is shown in [6] that automaton without additional resources cannot maintain movement direction on the graph all whose vertices are unlabelled or, equivalently, are labelled with the same label. Let an automaton A at a moment of time t be at a vertex $v(t)$ of the embedded graph G^∞. The automaton movement is called uniform and directional if there exists natural period T such that $v(t+T) - v(t) = v(t+2T) - v(t+T)$ holds for any moment of time t.

We will consider a collective of interacting automata $\mathcal{A} = (A_1, \dots, A_m)$. In addition to information about labelling each automaton A_i also receives information about presence of other automata from collective in the closed neighbourhood of current vertex. We call $\mathcal{A}$ the collective of automata without a compass if any component of $\mathcal{A}$ is an automaton without a compass. Further we will consider only such collectives.

Let $J \subset \{1, \dots, m\}$. A subsystem $(A_j)_{j\in J}$ of the collective $\mathcal{A}$ of interacting automata is called the pebbles in collective $\mathcal{A}$ if for all $j \in J$ the following conditions hold: (1) A_j has a single inner state; (2) A_j can only move if there is an automaton A_i ($i \notin J$) on the same vertex, and A_j can only move to the same vertex as A_i. For non-pebble automata pebbles play the role of external memory.

We will provide more precise definitions. Here $\mathcal{P}(S)$ denotes the set of all subsets of an arbitrary set S. Let I be a set of indices, and let $\{S_i | i \in I\}$ be a family of sets. We will denote by $\mathcal{T}(\{S_i | i \in I\})$ the set of all partial transversal of this family and by definition partial transversal contains no more than one element from each S_i. Note that the empty set is also a partial transversal.

The collective $\mathcal{A} = (A_0, A_1, \dots, A_m)$ consisting of one automaton A_0 and m pebbles A_1, ..., A_m is called the collective of interacting automata of type $(1, m)$. Let $I = \{0, 1, \dots, m\}$. Each automaton A_i from collective $\mathcal{A}$ is a sextuple $A_i = \left(S_i, X_i, Y_i, s_0^i, \varphi_i, \psi_i\right)$, where S_i is a finite set of internal states, $X_i = \{(\alpha_0, \{\alpha_1, \dots, \alpha_k\}) \mid \alpha_0, \alpha_1, \dots, \alpha_k \in M \times \mathcal{T}\left(\{S_j | j \in I \setminus \{i\}\}\right)\}$ is a finite input alphabet (here α_0 denotes the label of the current vertex ($pr_1(\alpha_0)$) and automata placed on it ($pr_2(\alpha_0)$)), and multiset $\{\alpha_1, \dots \alpha_k\}$ denotes the labels of vertices from neighbourhood of current vertex and automata placed on them); $Y_i = M \times \mathcal{P}\left(I \setminus \{i\}\right) \cup \{h\}$ is a finite output alphabet (here $y = h$ means "stay at the current vertex", $y \in Y_i \setminus \{h\}$ means "move to the vertex with label $pr_1(y)$, on which there are automata from the list $pr_2(y)$ and only they", $pr_2(y) = \varnothing$ means that there are no other automata in the target vertex); $s_0^i \in S_i$ is an initial state; $\varphi_i : S_i \times X_i \to S_i$ is a transition function; $\psi_i : S_i \times X_i \to Y_i$ is an output function. For any pebble A_j, $1 \leqslant j \leqslant m$ the following conditions are true: (1) $S_j = \left\{s_0^j\right\}$; (2) for any $x = (\alpha_0, \{\alpha_1, \dots, \alpha_k\}) \in X_j$ either $\psi_j\left(s_0^j, x\right) = h$, or if $\psi_j\left(s_0^j, x\right) = y \neq h$, then there exists $s \in S_0$ such that $s \in pr_2(\alpha_0)$ and $\psi_0(s, x') = y$, where $x' = \left(\left(pr_1(\alpha_0), (pr_2(\alpha_0) \setminus \{s\}) \cup \left\{s_0^j\right\}\right), \{\alpha_1, \dots, \alpha_k\}\right)$.

The behaviour of collective $\mathcal{A} = (A_0, A_1, \dots, A_{m+1})$ of type $(1, m)$ on graph G is the sequence $\pi(\mathcal{A}, G)$: $(\vec{x}_0, \vec{s}_0, \vec{y}_0)$, ..., $(\vec{x}_t, \vec{s}_t, \vec{y}_t)$, $(\vec{x}_{t+1}, \vec{s}_{t+1}, \vec{y}_{t+1})$, ..., where $\vec{x}_t = \left(x_t^1, \dots, x_t^{m+1}\right)$, $x_t^j = \left(\alpha_{0_t}^j, \left\{\alpha_{1_t}^j, \dots \alpha_{t_k}^j\right\}\right) \in X_j$, $\vec{s}_t = \left(s_t^1, \dots, s_t^{m+1}\right)$, $s_t^j \in S_j$ $\vec{y}_t = \left(y_t^1, \dots, y_t^{m+1}\right)$, $y_t^j \in Y_j$ $(0 \leqslant j \leqslant m)$ such that $s_{t+1}^j = \varphi_j\left(s_t^j, x_t^j\right)$, $y_{t+1}^j = \psi_j\left(s_t^j, x_t^j\right)$.

Suppose that all automata from the collective $\mathcal{A}$ are placed on the same vertex of the graph G at the initial moment of time.

3. Vertex labelling sufficient for directional movement.

Consider an embedding of graph G^∞ on the plane $\mathbb{Z}^2$. Let's choose two pairs of opposite directions on graph G^∞ corresponding to the abscissas and ordinates axes of the plane $\mathbb{Z}^2$. We call these directions basic. It easy to check that the movement of the automaton in any fixed direction different from the basic can be represented as a combination of moves in the basic ones by increasing the number of states of the automaton. Hence we can restrict ourselves to considering automata moving only in the basic directions. A labelling is said to be periodic in direction $(q,r) \in \mathbb{Z}^2$ if $\mu(i+q,j+r) = \mu(i,j)$ for all $i,j \in \mathbb{Z}$. We call a labelling traversable if by using it an automaton can move along graph in any directions. A vertex labelling that minimize the number of labels needed for a given graph G is called a minimum vertex labelling of G.

We call a labelling function μ deterministic (or D-labelling) if all vertices in closed neighbourhood of any vertex have different labels. A labelling graph with deterministic labelling function is said to be deterministic (or D-graph). In [8] it is proved that following properties follow from the D-graph definition: (1) for any graph vertex any word in alphabet M defines no more than one path from this vertex; (2) the distance between two vertices with the same labels greater than or equal to 4. These properties enable targeted movement of a graph-walking automaton along D-graph. For example, if we know labels of the paths connecting the vertices of the graph with each other, then we can construct an automaton which can move along these paths.

Theorem 1. *For a minimal traversable deterministic labelling of G^∞ it is necessary and sufficient to have five types of labels.*

Proof. Consider an embedding of graph G^∞ on the plane $\mathbb{Z}^2$. For any vertex v of graph G^∞ the equality $\left|O_{(v)}\right| = 5$ holds. Therefore five types of labels needed for D-labelling of closed neighbourhood of this vertex. Let $M = \{a,b,c,d,e\}$. We will choose an arbitrary vertex (i,j) of embedded graph G^∞ and define the labelling of neighbourhood of this vertex as follows: $\mu((i,j)) = a$, $\mu((i+1,j)) = b$, $\mu((i,j+1)) = c$, $\mu((i-1,j)) = d$, $\mu((i,j-1)) = e$ (consult Fig. 1(a) for illustration). By definition of the D-labelling it follows that vertex $(i+1,j+1)$ must be labelled with either label d or label e. Let $\mu((i+1,j+1)) = e$. Then, by definition of D-labelling, it follows that $((i-1,j+1)) = b$, $\mu((i-1,j-1)) = c$, and $\mu((i+1,j-1)) = d$. We continue labelling in this fashion obtaining $\mu((i+2,j)) = c$, $\mu((i,j+2)) = d$, $\mu((i-2,j)) = c$, $\mu((i,j-2)) = b$ (consult Fig. 1(b) for illustration).

Note that after performing these steps for each label from M there exists a vertex labelled by this label such that all vertices from its neighbourhood are already labelled. For the label a it is the vertex (i,j), for the label b, the vertex $(i+1,j)$, for the label c , the vertex $(i,j+1)$, for the label d, the vertex $(i-1,j)$, and for the label e, the vertex $(i,j-1)$. Using labelling of the neighbourhoods of these vertices as the patterns for further labelling, we obtain a deterministic subgraph of the graph G^∞, shown in Fig. 1(c).

For labelling of neighbourhood of vertex $(i+2,j+2)$, $\mu((i+2,j+2)) = b$, we will use corresponding pattern i.e. the labelling of the neighbourhood of the vertex $(i+1,j)$, $\mu((i+1,j)) = \mu((i+2,j+2))$. Vertex $(i+3,j+2)$ will labelled by label c as a result. Note that $\mu((i+3,j+2)) = \mu((i-2,j+2))$. We will continue labelling by using corresponding patterns and obtain that $\mu((i+3,j+1)) = \mu((i-2,j+1)) = a$, $\mu((i+3,j)) = \mu((i-2,j)) = e$, $\mu((i+3,j-1)) = \mu((i-2,j-1)) = b$, and $\mu((i+3,j-2)) = \mu((i-2,j-2)) = d$. It is easy to check that $\mu((i,j)) = \mu((i+5k,j))$ holds for all $i,j,k \in \mathbb{Z}$ provided that corresponding labelling patterns are used. Similarly

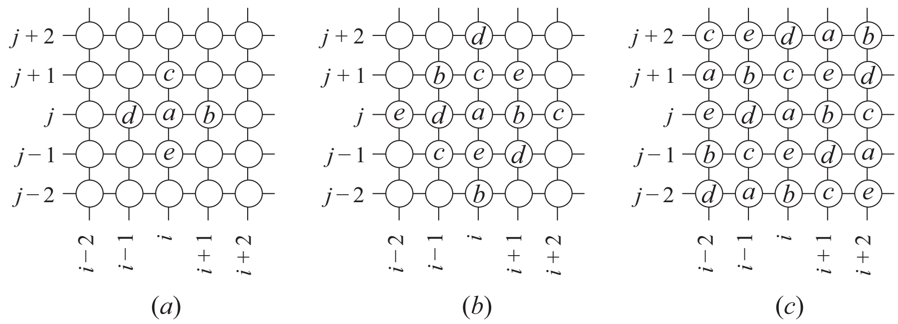

Fig. 1. A minimal D-labelling of graph G^∞.

we obtain that $\mu((i,j)) = \mu((i,j+5l))$ holds for all $i,j,l \in \mathbb{Z}$ provided that corresponding labelling patterns are used. It follows from above that we can construct D-labelling of graph G^∞ provided that corresponding labelling patterns are used. Thus, it is necessary and sufficient to have labels of five different types to construct a minimal D-labelling of the graph G^∞.

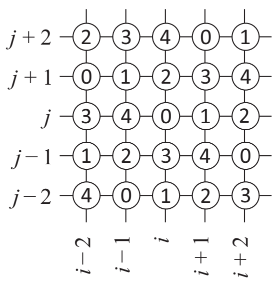

Fig. 2. A minimal traversable D-labelling of graph G^∞.

We next prove that the minimal D-labelling described above is traversable. Without loss of generality we can assume that $a = 0$, $b = 1$, $c = 2$, $d = 4$, $c = 3$. We will define labelling of neighbourhood of any vertex (i,j) of embedded graph G^∞ by the following condition: if $\mu((i,j)) = x$, $x \in M$, then $\mu((i+1,j)) = x \oplus_5 1$, $\mu((i-1,j)) = x \oplus_5 (-1)$, $\mu((i,j+1))\, x \oplus_5 2$, $\mu((i,j-1)) = x \oplus_5 (-2)$, where $\oplus_5$ denote modulo 5 addition (consult Fig. 2 for illustration). Let graph G^∞ labelled in accordance with the above rule. Since the automaton "sees" only vertex labels and not their names, it follows that we must set the directions on the D-graph G^∞ in the language "understandable" to the automaton. Let the automaton is on the vertex (i,j) with label $a \in M$. We will say that the vertex with label $a \oplus_5 1$ (i.e. the vertex $(i+1,j)$) is in the "east", the vertex with label $a \oplus_5 (-1)$ (i.e. the vertex $(i-1,j)$) is in the "west", the vertex with label $a \oplus_5 2$ (i.e. the vertex $(i,j+1)$) is in the "north", and the vertex with label $a \oplus_5 (-2)$ (i.e. the vertex $(i,j-1)$) is in the "south". Then "east" denotes the direction $(1,0)$, "west", the direction $(-1,0)$, "north", the direction $(0,1)$, and "south", the direction $(0,-1)$. Moving eastward the automaton every times moves to a vertex whose label is equal to the sum modulo 5 label of the current vertex and 1. The movements to the "west", "north" and "south" are determine similarly.

If the automaton only moves in one of four basic directions ("east", "west", "north" or "south") then the equality $v(t+T) - v(t) = v(t+2T) - v(t+T)$ holds for $T = 1$ for any time moment t. This means that directional movement of the automaton is uniform.

This finished the proof. □

Corollary 1.1. *For a minimal traversable deterministic labelling of any subgraph of G^{∞} with maximum degree 4 it is necessary and sufficient to have five types of labels.*

Consider minimal traversable D-labellings of subgraphs of the graph G^{∞}, whose degrees are less than 4.

Theorem 2. *For a minimal traversable deterministic labelling of $G^{(\infty,2)}$ it is necessary and sufficient to have five types of labels.*

Proof. Consider an embedding of graph $G^{\infty,2}$ on the grid $\mathbb{Z} \times \mathbb{Z}_2$. For any vertex v of graph $G^{\infty,2}$ the equality $\left|O_{(v)}\right| = 4$ holds. Therefore four types of labels needed for D-labelling of closed neighbourhood of this vertex.

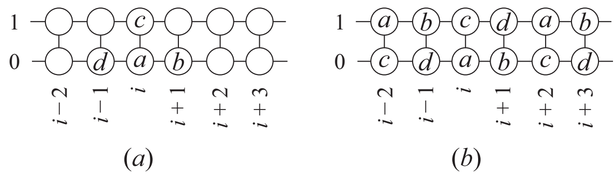

Fig. 3. A minimal D-labelling of graph $G^{(\infty,2)}$.

Let $M = \{a, b, c, d\}$. We will choose an arbitrary vertex $(i, 0)$ of embedded graph $G^{\infty,2}$ and define labelling of neighbourhood of this vertex as follows: $\mu((i,0)) = a$, $\mu((i+1,0)) = b$, $\mu((i,1)) = c$, $\mu((i-1,0)) = d$ (consult Fig. 3(a) for illustration). By definition of D-labelling, the vertex $(i+1, 1)$ should be labelled by label d, the vertex $(i-1, 1)$, by label b. After that, the vertices $(i-2, 0)$ and $(i+2, 0)$ should be labelled by label c, the vertices $(i-2, 1)$ and $(i+2, 1)$, by label a (consult Fig. 3(b) for illustration). We continue labelling in this fashion and obtain D-labelling of graph $G^{\infty,2}$ with four type of labels.

Let us check if this labelling is traversable. For the subgraph shown in Fig. 3(b), the equality $\mu((i,0)) = \mu((i+2,1)) = a$ holds. Regardless of which vertex $(i, 0)$ or $(i+2, 1)$ the automaton is on, its movement to a vertex with label b is the movement in the direction $(1, 0)$, and its movement to a vertex with label d is the movement in the direction $(-1, 0)$. But in the case of the vertex $(i, 0)$, the automaton movement to the vertex with label c is a movement in the direction $(0, 1)$, and in the case of the vertex $(i+2, 1)$, the movement in the direction $(0, -1)$. Therefore, four different labels is not enough for a minimal traversable D-labelling of the graph $G^{\infty,2}$.

A traversable D-labelling of the graph G with five types of labels exists due to the fact that $G^{(\infty,2)}$ is a subgraph of the graph G^{∞}. This finished the proof. □

Theorem 3. *For a minimal traversable deterministic labelling of P^∞ it is necessary and sufficient to have three types of labels.*

Proof. Consider an embedding of graph P^∞ on the one-dimensional grid $\mathbb{Z} \times \mathbb{Z}_1$. For any vertex v of graph P^∞ the equality $\left|O_{(v)}\right| = 3$ holds. Therefore three types of labels needed for D-labelling of closed neighbourhood of this vertex. We will choose an arbitrary vertex v of the graph P^∞. Let $O_{(v)} = \{v, r, t\}$, $\mu(r) = a$, $\mu(v) = b$, $\mu(t) = c$, $\{a, b, c\} \subseteq M$. Next let $h \in O_{(r)}$ and $q \in O_{(t)}$, then vertex h is at a distance of 4 from vertex t and can be labelled by label c. Similarly vertex q is at a distance of 4 from vertex r and can be labelled by label a. We continue labelling in this fashion and obtain D-labelling of graph P^∞ with three type of labels. Therefore, such labelling is minimal.

Let $(i, 0) \in \mathbb{Z} \times \mathbb{Z}_1$ be an arbitrary vertex of embedded graph P^∞ with minimal D-labelling. By definition of D-graph it follows that $\mu\left((i,0)\right) \neq \mu\left((i+1,0)\right)$, $\mu\left((i,0)\right) \neq \mu\left((i+2,0)\right)$ and $\mu\left((i+1,0)\right) \neq \mu\left((i+2,0)\right)$. As shown above $\mu\left((i,0)\right) = \mu\left((i+3,0)\right)$. Since the vertex $(i, 0)$ is chosen arbitrarily, it follows from the last equality that the minimal D-labelling of the graph P^∞ is periodic in the direction 3.

Next we prove that the minimal D-labelling of graph P^∞ is traversable. Without loss od generality we can assume $M = \{0, 1, 2\}$. In this case if $\mu(v) = a$, $a \in M$, then vertices adjacent to v have labels $b = a \oplus_3 1$ and $c = a \oplus_3 (-1)$, where $\oplus_3$ denotes addition modulo 3. Using this labelling, an automaton can move in two opposite directions, which we will nominally call "east" and "west". Let the automaton is on a vertex with label a. Moving eastward the automaton every time moves to a vertex with label b, and moving westward, to a vertex with label c.

Let the automaton only moves eastward (or only westward), then the equality $v(t+T) - v(t) = v(t+2T) - v(t+T)$ holds for $T = 1$ for any time moment t. This means that directional movement of the automaton is uniform. This finished the proof. □

4. Capabilities of a single automaton upon traversing G^∞.

The infinite grid graph after constructing a minimal traversable deterministic labelling on it and fixing two pairs of opposite directions on it becomes an analogue of an infinite mosaic labyrinth without holes [1, 2]. The key problem for automata and labyrinths is the problem of constructing a finite automaton that traverse a given class of labyrinths, that is, an automaton in the initial state is placed at any vertex of any labyrinth from this class, and must visit all vertices of this labyrinth up to some moment of time [9]. We say that automaton traverse infinite graph G^∞ if it visits any randomly selected vertex of the graph in a finite time. In his paper [10], Andzhan prove that a collective of one automaton and three pebbles can traverse infinite mosaic labyrinth without holes and any collective with fewer pebbles cannot. We will show that methods and algorithms proposed in this work can be used to traverse infinite grid D-graph after modification associated with a changed concept of direction.

Theorem 4. *Any single automaton cannot explore D-graph G^∞.*

Proof. Let automaton A infinitely moves along D-graph G^∞. We first prove that there are at least two labels such that the automaton visits the vertices labelled with these labels infinitely many times whatever its trajectory. Without loss of generality we can assume $M = \{a, b, c, d, e\}$ and graph G^{∞} is marked as shown in Fig.4(a). Assume to the contrary that there is an infinite trajectory of automaton such that the number of entries in corresponding word is finite for any label. Let the automaton no longer visits vertices with label e from moment of time t_1 but visits vertices with labels a, b, c, and d (consult Fig. 4(b) for illustration). Next, let the automaton no longer visit vertices with label d from moment of time t_2 but visits vertices with labels a, b, and c (consult Fig. 4(c) for illustration). Finally, let the automaton no longer visit vertices with label c from moment of time t_3 but visits vertices with labels a and b (consult Fig. 4(d) for illustration). Since the automaton moves infinitely, we have that it will endlessly move between two adjacent vertices with labels a and b from moment of time t_3. No further number of labels reduction possible. We have arrived at a contradiction. Hence the automaton visits the vertices with the same label infinitely many times when moving along the graph. Among them there exists vertices v_i and v_j on which the automaton A are in the same internal state. Since the behaviour of A depends only on internal state it follows that A will move from vertex $v(j)$ in the same direction as from vertex $v(i)$ i.e. $\mu\left(v(j+1)\right) = \mu\left(v(i+1)\right)$. Then $v(j+1) - v(i+1) = v(j) - v(i)$. From the obvious equality $v\left(i + (j-i)\right) = v(i) + \left(v(j) - v(i)\right)$ it follows that $v\left(t + (j-i)\right) = v(t) + \left(v(j) - v(i)\right)$ holds for all $t > i$.

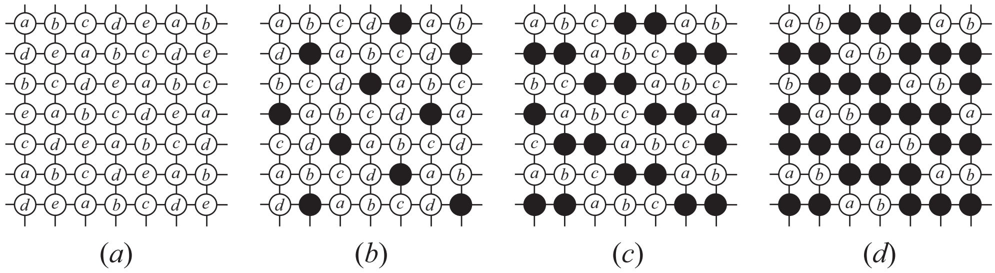

Fig. 4. Reduction of the number of labels using in automaton movement. (A) no reduction, (b) after the ban on use "e", (c) after the ban on use "d" and "e", (d) after the ban on use "c", "d", and "e".

We next prove that periodic trajectory of the automaton is inside some half-stripe. Let us draw a line l passing through the vertices v_i and v_j mentioned above. It follows from above that this line will passing through infinite amount of vertices belonging to the trajectory of the automaton A. Line l divides graph D^{∞} into two half-grid subgraphs. Let us choose any of these subgraphs and find on it a vertex that belongs to the trajectory of the automaton and is the most distant from line l. We will draw a line passing through this vertex and parallel to the line l. This line is the first bound of the desired half-stripe. Since the trajectory of automaton is periodic, it follows that there are infinitely many vertices belong to both first bound and trajectory of automaton. The bound of the second half-stripe is drawn on the other subgraph in a similar way. Finally let us draw a line perpendicular to the bounds and passing through initial vertex of the trajectory of automaton. The set of vertices bounded by these lines forms desired half-stripe. From the obvious fact that any half-stripe don't cover graph 1, it follows that the statement of the theorem is true. This finished the proof. □

Corollary 4.1. *Any single automaton cannot explore D-graphs $G^{(+\infty,\infty)}$ and $G^{(\infty,n)}$.*

5. Capabilities of automaton with one pebble upon traversing G^{∞}.

The impossibility to explore D-graph G^{∞} by a single automaton poses the problem of possible enrichment of the automaton model which is able to solve exploration problem. Several enrichments are suggested. One of the most natural approaches is to give the automaton an ability to place additional labels on the vertices of the graph (or paint the vertices in some colours) [9]. In essence, the ability to colour vertices means that the automaton possesses an unbounded external memory, which greatly increases its possibilities. Another enhancement of a single automaton is a system of interacting automata referred to as a collective. In contrast to a single automaton a collective analyses a graph with regard for positions of its members there. If some members of the collective are automata of the simplest form whose positions are completely determined by the remaining automata of the collective then these simplest automata are called pebbles.

Theorem 5. *Any collective of one automaton and one pebble cannot explore D-graph G^{∞}.*

Proof. Consider a collective of one automaton A_0 and one pebble A_1. The proof is by case analysis. There are two cases:

1. A_0 and A_1 move together all the time.

2. A_0 can move away from A_1.

Case 1: Suppose that automaton A_0 and pebble A_0 move together all the time. Then collective (A_0, A_1) operates like single automaton and cannot explore D-graph G^{∞} by Theorem 4.

Case 2: Suppose that automaton A_0 can move away from pebble A_1. This case splits into two subcases:

Case 2.1: A_0 and A_1 are together at the same vertex infinitely many times.

Case 2.2: A_0 and A_1 are together at the same vertex only a finite number of times.

Case 2.1: Suppose that automaton A_0 and pebble A_1 are together at the same vertex infinitely many times. Consider the infinite sequence of vertices on which A_0 and A_1 meet each other. In this sequence there is an infinite subsequence consisting of vertices with the same label. Among these vertices there are vertices $v(i)$ and $v(j)$, $i < j$, on which the automaton was in the same internal state. Since the behaviour of collective (A_0, A_1) depends only on the state of automaton A_0 it follows that collective (A_0, A_1) will move from vertex $v(j)$ in the same direction as from vertex $v(i)$. As in the proof of Theorem 4 we obtain that trajectory of collective (A_0, A_1) is periodic. Therefore this collective cannot explore D-graph G^{∞}.

Case 2.2: Suppose that automaton A_0 and pebble A_1 are together at the same vertex only a finite number of times. Let $v(k)$ be the last vertex on which A_0 and A_1 were together. Then there exists two moments of time t' and t'', $k < t' < t''$, such that automaton A_0 is in the same internal state on the vertices $v(t')$ and $v(t'')$, $\mu(v(t')) = \mu(v(t''))$. As in the proof of Theorem 4 we obtain that trajectory of automaton A_0 is periodic. Therefore collective (A_0, A_1) cannot explore D-graph G^{∞}.

This finished the proof. □

Corollary 5.1. *Any collective of one automaton and one pebble cannot explore any deterministic stripe graph.*

6. Capabilities of automaton with two pebbles upon traversing G^{∞}.

Theorem 6. *Any collective of one automaton and two pebble cannot explore D-graph G^{∞}.*

Proof. The idea of the proof is similar to that of Theorem 3 in [10]. Let automaton A_0 and pebbles A_1, A_2 are placed on the vertex $v(0)$ at the initial moment of time $t = 0$. Let predicate $R_{0,1}(t)$ denote that A_0 and A_1 are on the same vertex at the moment of time t, and predicate $R_{0,2}(t)$ denote the similar statement for A_0 and A_2. We will denote by $s(t)$ the inner state of automaton A_0 at the moment of time t. By $D_{1,2}(t)$ denote the distance between A_1 and A_2 at the moment of time t.

If $R_{0,1}(t)$ holds only for a finite amount of natural t then after last rendezvous with A_2 the automaton A_0 and the pebble A_2 operate like a collective of one automaton and one pebble. Therefore this collective cannot explore D-graph G^{∞} by Theorem 5 and the statement of the theorem is satisfied. The same reasoning applies to the case where $R_{0,2}(t)$ holds only for a finite amount of natural t. Further, we assume that $R_{0,1}(t)$ and $R_{0,2}(t)$ hold for an infinite amount of natural t.

Suppose the distance between A_1 and A_2 does not exceed a certain constant C while the collective (A_0, A_1, A_2) operates on the graph G^{∞}, i.e. there exists an infinite amount of moments of time t_i, $i = 1, 2, \ldots$, such that $D_{1,2}(t_i) < C$. Let $v_1(t_i)$ and $v_2(t_i)$ denote the vertices where the pebbles A_1 and A_2 are placed at the moment of time t_i. Consider the pairs $(v_2(t_i) - v_1(t_i), s(t_i))$. Since $v_2(t_i) - v_1(t_i)$ is an integer pair and $|v_2(t_i) - v_1(t_i)| = D_{1,2}(t_i)$, it follows that amount of different pairs $v_2(t_i) - v_1(t_i)$ is finite. Hence the amount of different pairs $(v_2(t_i) - v_1(t_i), s(t_i))$ is finite too and there exists two equal pairs in the infinite sequence of such pairs. This means that the trajectory of the collective (A_0, A_1, A_2) is periodic (the proof is similar to the proofs of Theorem 4 and 5). Therefore this collective cannot explore D-graph G^{∞} in this case.

Suppose the distance between pebbles increases unlimitedly with increasing t. Consider the behaviour of automaton A_0 in very moment $t > T_0$ when $D_{1,2}(t)$ is greater than the number of internal states of A_0. From the above it follows that automaton A_0 moves from the pebble A_1 to the pebble A_2 and returns back. Let $R_{0,1}(t_1)$ and $R_{0,2}(t_2)$ holds, $t_1 < t_2$, automaton A_0 moves from A_1 to A_2 and does not meet the pebbles in the time interval between t_1 and t_2. Hence in the time interval from t_1 to t_2 automaton operates without pebbles. By Theorem 4 in this time interval trajectory of A_0 is inside some half-stripe. This half-stripe is said to be the *transition half-stripe*. The direction of transition half-stripe depends only on $s(t_i)$. Since the amount of inner states of A_0 is finite, we have the amount of transition half-stripes different in directions is finite too. The width of the half-stripes can be bounded by some constant C. The same conclusion can be drawn for the transition half-stripes from A_2 to A_1.

Let $R_{0,1}(t_3)$ holds and A_0 does not meet A_1 in time interval between t_2 and t_3 i.e. $R_{0,1}(t_3)$ is the first rendezvous A_0 and A_1 after $R_{0,1}(t_1)$. We shall show that during the time interval from t_1 to t_3 the pebble A_2 will move a distance no more than the amount n of inner states of A_0. In fact, A_2 can only move with A_0. If A_2 has moved a distance greater than n, then A_0 has moved with A_2 more than n times. Then at least twice A_0 has transited to the same inner state during the movements with A_2. Therefore, by Theorem 5, the trajectory of the collective (A_0, A_2) is periodic and A_0 does not depart from A_2 further than the distance n. Hence at the moment of time t_3 (for which $R_{0,1}(t_3)$ holds) the pebbles A_1 and A_2 are at a distance less or equal to n from each other, which is impossible by assumption.

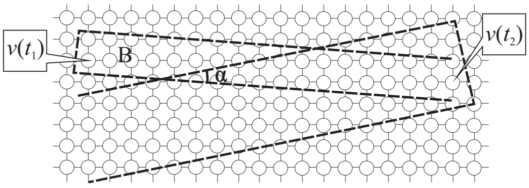

Fig. 5. A rational angle α between directions of half-stripes.

Let us prove that there exists $T_1 > T_0$ such that the directions of all arising transition half-stripes coincide with an accuracy of 180° for all $t > T_1$. Suppose the contrary, the directions of two consecutive transition half-stripes are not opposite. Let B denote the half-stripe arising from movement of A_0 from A_1 to A_2. Assume A_1 stays on the vertex $v(t_1)$ and A_0 meets with A_2 on the vertex $v(t_2)$ for the first time after t_1, $t_1 < t_2$. From the above it follows that automaton A_0 can move the pebble A_2 at a distance of no more than n from its current vertex $v(t_2)$. Consider all vertices at a distance of no more than n from $v(t_2)$. One of these vertices belongs to the second of the considered transition half-stripes. Since the width of all transition half-stripes is bounded by constant C, we have that trajectory of A_0 back to A_1 is entirely inside a half-stripe with a width at most $2C + 2n$ and a direction not opposite to that of the half-strip B. It is clear that this half-stripe include the vertex $v(t_1)$ iff $D_{1,2}(t_1)$ does not exceed an upper bound which depends only on a rational angle α between directions of half-stripes (consult Fig. 5 for illustration). Since automaton A_0 is finite, we have that amount of half-stripes different in direction is finite too. Hence the amount of different angles α is also finite. It follows that automaton A_0 will not return to A_1 with a sufficiently large $D_{1,2}(t_i)$. The resulting contradiction proves that directions of all transition half-stripes coincide with an accuracy of 180° from some moment of time.

Further, we will consider the behaviour of the collective (A_0, A_1, A_2) only for $t > T_1$. The only direction of the transition half-stripes at this time is called the *main direction*.

Let $R_{0,1}(t_1)$, $R_{0,2}(t_2)$, $R_{0,1}(t_3)$, $R_{0,2}(t_4)$ hold, $t_1 < t_2 < t_3 < t_4$, and during time intervals (t_1, t_2), (t_2, t_3) and (t_3, t_4) the automaton A_0 does not meet the pebbles A_1 and A_2. Assume that $s(t_1) = s(t_3)$. The automaton A_0 leaves the pebble A_1 on its current vertex and moves along periodic trajectory to the pebble A_2. A rendezvous of A_0 and A_2 can occur generally anywhere in the period of trajectory. Hence the states of the automaton can differ at time moments t_2 and t_4. It follows that interaction of A_0 and A_2 and return to A_1 can differ too. The same departure leads to the same arrival iff the automaton trajectory differs only by an integer number of periods in both cases. Here we say that *period* is the automaton A_0 displacement vector and denote by $\vec{f}$. This vector is parallel to the main direction. Let $\vec{F}$ be a vector which connects current vertices of the pebbles A_1 and A_2. We will subtract $\vec{f}$ from $\vec{F}$ until we get a vector with minimal projection on the main direction. This vector we will call *the remainder of vector $\vec{F}$ from dividing by vector $\vec{f}$* and denote by rest(t). Since the beginning and the end of vector $\vec{f}$ have integer coordinates, we have that the amount of remainders is finite. From the foregoing it follows that state $s\,(t_2)$ is completely determined by state $s\,(t_1)$ and the remainder of vector $\vec{F}$ from dividing by vector $\vec{f}$.

We will build a normal h to the main direction. Let us fix on h positive direction, starting point and scale coinciding with the scale on the coordinate axes. By $r(t)$ denote the difference between the numerical values of the projections of the vertices $v_1(t)$ and $v_2(t)$ on the axis h. Since for $t > T_1$ the direction of the half-stripe transition is constant, their width is bounded, the main direction forms with the coordinate axes an angle with a rational tangent, automaton A_0 moves the pebble at a distance of less than n for one approach, we have that $r(t)$ can take only a finite amount of different values.

Consider the moments of time $t_1 < t_2 < t_3 \ldots$ at which $R_{0,1}\,(t_1)$, $R_{0,1}\,(t_2)\,,\ldots$ hold. Let $P(i) = (s\,(t_i)\,, r\,(t_i)\,, \mathrm{rest}\,(t_i))$, $i = 1, 2, \ldots$. From the above it follows that $P(i+1)$ is completely determined by $P(i)$. Since every component of triple $P(i)$ takes only a finite amount of values, we have that there are a finite amount of such triples. Hence there exists two equal triples among them.

Let $P(k) = P(l)$, $k < l$, and $v_0(t)$ denote the vertex where the automaton A_0 is at moment of time t. The following two cases are possible.

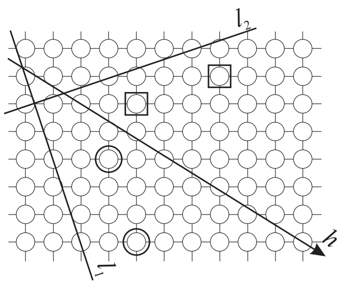

Fig. 6. An angle that automaton with two pebbles never exit. Circles indicate the vertices $v_1\,(t_q)$ and $v_1\,(t_u)$. Squares indicate the vertices $v_2\,(t_q)$ and $v_2\,(t_u)$

Case 1: the projections of the vertices $v_0\,(t_k)$ and $v_0\,(t)$ on the axis h are coincide with each other. Let us project the trajectory of automaton A_0 in the time interval from t_k to t_l on the axis h. It is clear that the trajectory will fit inside some finite segment. Let us draw lines through the ends of this segment parallel to the main direction. Since the automaton A_1 at moment of time $t_l + 1$ will move in the same direction as at moment of time $t_k + 1$, at moment of time $t_l + 2$ – in the same direction as at moment of time $t_k + 2$ etc., we have that afterwards automaton A_0 will not move outside the stripe bounded by these lines.

Case 2: the projections of the vertices $v_0\,(t_k)$ and $v_1\,(t_l)$ on the axis h are differ from each other. It is clear that $P\,(l + m(k-l)) = P(k)$ holds for all natural n. Hence there exists q and u, $q < u$, such that $P(q) = P(u)$ and projections of $v_0\,(t_q)$ and $v_0\,(t_u)$ are at a distance greater than $2n$. Let projection of $v_0\,(t_u)$ on the axis h be to the right of projection of $v_0\,(t_q)$. Therefore projection of $v_1\,(t_u)$ be to the right of projection of $v_1\,(t_q)$ and projection of $v_2\,(t_u)$ be to the right of projection of $v_2\,(t_q)$. This follows from the fact that automaton A_0 moves a pebble at a distance of less than n for one approach (consult Fig. 6 for illustration). Let us draw a line l_1 parallel to the line passing through the vertices $v_1\,(t_q)$ and $v_1\,(t_u)$ and another line l_2 parallel to the line passing through the vertices $v_2\,(t_q)$ and $v_2\,(t_u)$ so that all vertices visited by A_0 until t_v are inside angle formed by lines l_1 and l_2. Automaton A_0 will never exit from this angle.

This finished the proof. □

Corollary 6.1. *For every deterministic stripe graph, there exists a collective of one automaton and one pebble, which explores this graph.*

Corollary 6.2. *For every deterministic angle graph provided that its value less than 180°, there exists a collective of one automaton and one pebble, which explores this graph.*

7. Capabilities of automaton with three pebbles upon traversing G^{∞}.

Theorem 7. *There exists a collective of one automaton and three pebbles which explore D-graph G^{∞}.*

Proof. As the proof, we present algorithm for D-graph G^{∞} exploration by automaton with three pebbles. The algorithm works as follows. In lines 1-5 automaton places pebbles to initial locations (consult Fig. 1(a) for illustration). During infinite loop of lines 6-27 automaton moves from pebble to pebble and places pebbles to new locations.

Algorithm 1. D-graph G^{∞} exploration by automaton with three pebbles.

Require: A_0, A_1, A_2, A_3 are placed on arbitrary vertex of D-graph G^{∞}
Ensure: the trail of A_0 visiting every vertex of G^{∞} at least once

1: A_0, A_1, A_2, A_3 move to the 'northern' vertex
2: A_0, A_2, A_3 move to the 'southern' vertex
3: A_0, A_2, A_3 move to the 'western' vertex
4: A_0, A_3 move to the 'eastern' vertex
5: A_0, A_3 move to the 'eastern' vertex
6: **loop**
7:     **while** A_1 isn't found on the current vertex **do**
8:         A_0 moves to the 'northern' vertex
9:         A_0 moves to the 'western' vertex
10:     **end while**
11:     A_0, A_1 move to the 'northern' vertex
12:     A_0 moves to the 'southern' vertex
13:     **while** A_2 isn't found on the current vertex **do**
14:         A_0 moves to the 'western' vertex
15:         A_0 moves to the 'southern' vertex
16:     **end while**
17:     A_0, A_2 move to the 'western' vertex
18:     A_0, A_2 move to the 'southern' vertex
19:     A_0, A_2 move to the 'western' vertex
20:     **while** A_3 isn't found on the 'northern' vertex **do**
21:         A_0 moves to the 'eastern' vertex
22:     **end while**
23:     A_0 moves to the 'northern' vertex
24:     A_0, A_3 move to the 'eastern' vertex
25:     A_0, A_3 move to the 'southern' vertex
26:     A_0, A_3 move to the 'eastern' vertex

27: **end loop**

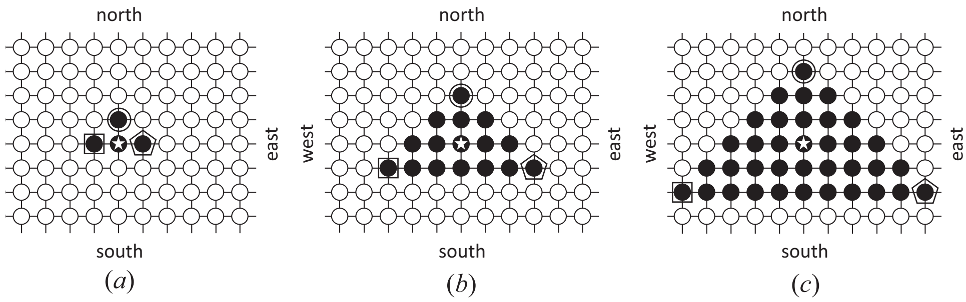

Fig. 7. The operation of Algorithm 1 on D-graph G^∞. (A) after lines 1-5, (b) after first iteration, (c) after second iteration. Star indicates the initial vertex. Circle, square and pentagon indicate the current locations of A_1, A_2, A_3, respectively.

To analyse the algorithm we need some notation. We will denote by $v_1(i)$, $v_2(i)$ and $v_3(i)$ the vertices on which the pebbles A_1, A_2 and A_2 are placed after the algorithm's i iteration. Also we will denote by $[v_1(i), v_2(i)]$ the segment of the line passing through the vertices $v_1(i)$ and $v_2(i)$. The line segments $[v_1(i), v_3(i)]$ and $[v_2(i), v_2(i)]$ are defined similarly.

The main idea of the algorithm is to maintain an explored subgraph to which the newly explored parts of the graph are merged. After some iteration of algorithm let the automaton has already explored a connected subgraph of G^∞ such that its inner faces do not contain unvisited vertices. The set of all not-visited neighbours of visited vertices is called the fringe. In the next iteration the automaton should visit all vertices in the fringe and add them to explored subgraph. Not-visited neighbours of visited vertices form a new fringe etc. It is clear that an arbitrary fixed vertex of the graph will be visited over time proportional to the distance from this vertex to the initial vertex.

We proceed by induction on the number of algorithm's iterations. As a base case observe that after the first iteration all vertices inside and on the sides of the triangle bounded by $[v_1(i), v_2(i)]$, $[v_1(i), v_3(i)]$ and $[v_2(i), v_3(i)]$ are already visited by automaton A_0 (consult Fig. 7(b) for illustration). For the inductive step, let $k > 1$ be an integer, and assume that after the k iteration all vertices inside and on the sides of the triangle bounded by $[v_1(k), v_2(k)]$, $[v_1(k), v_3(k)]$ and $[v_2(k), v_3(k)]$ are already visited by automaton A_0. We want to show that the similar statement holds for $k+1$ iteration. All vertices on the line segment $[v_1(k+1), v_3(k+1)]$ are 'northern' neighbours of vertices on the line segment $[v_1(k), v_3(k)]$ except the vertex $v_3(k+1)$ and 'eastern' neighbour of the vertex $v_3(k)$. A_0 visits all these vertices due to the lines 7-11 and 24-26. All vertices on the line segment $[v_1(k+1), v_2(k+1)]$ are 'northern' neighbours of vertices on the line segment $[v_1(k), v_2(k)]$ except the vertex $v_2(k+1)$ and 'western' neighbour of the vertex $v_2(k)$. The vertex $v_1(k+1)$ has been visited earlier. A_0 visits all remaining vertices due to the lines 13-19. All vertices on the line segment $[v_2(k+1), v_3(k+1)]$ are 'southern' neighbours of vertices on the line segment $[v_2(k), v_3(k)]$. The vertex $v_2(k+1)$

has been visited earlier. A_0 visits all remaining vertices due to the lines 20-26. This completes the proof. □

8. Conclusion.

This work proposes the vertex labelling of the infinite square grid graph, due to which the graph-walking automaton can move along it in any arbitrarily chosen direction. It is shown that a collective of one automaton and three pebbles can explore infinite square grid graph with such labelling and any collective with fewer pebbles cannot. The results regarding to exploration of the infinite square grid graph coincide with the results of A.V. Andzhan (Andzans) regarding to traversal of the infinite mosaic labyrinth without holes. It is shown that infinite grid graph after constructing this labelling and fixing two pairs of opposite directions on it becomes an analogue of an infinite mosaic labyrinth without holes. For further investigation, the question of the minimum amount of different label types needed for the labeled square grid graph to be traversable by graph-walking automaton is of interest.

Цитированная литература

1. *Kilibarda G., Kudryavtsev V.B., Ušćumlić Š.* Independent systems of automata in labyrinths // Discrete Mathematics and Applications. – 2003. – Vol. 13, Is. 3. – P. 221–255.
2. *Kilibarda G., Kudryavtsev V.B., Ušćumlić Š.* Collectives of automata in labyrinths // Discrete Mathematics and Applications. – 2003. – Vol. 13, Is. 5. – P. 429–466.
3. *Dudek G., Jenkin M.* Computational Principles of Mobile Robotics. – Cambridge University Press, 2010.
4. *Blum F., Kozen D.* On the power of the compass (or, why mazes are easier to search than graphs) // Proceedings of the 19th Annual Symposium on Foundations of Computer Science (SFCS '78). IEEE Computer Society, Washington, DC, USA. 1978. –P. 132–142.
5. *Donald B.R.* The Compass That Steered Robotics. Logic and Program Semantics. – Lecture Notes in Computer Science, Vol. 7230. Springer, 2012. – P. 50–65.
6. *Курганский А.Н., Сапунов С.В.* О направленном перемещении коллектива автоматов без компаса на одномерной целочисленной решетке // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. – 2016. – Т. 16, вып. 3. – С. 356–365.
7. *Bondy J.A., Murty U.S.R.* Graph Theory. – Springer–Verlag London, 2008. – 663 p.
8. *Грунский И.С., Сапунов С.В.* Восстановление графа операционной среды мобильного робота путем разметки вершин, пригодной для дальнейшей навигации // Искусственный интеллект. – 2012. – № 4. – С. 420–428.
9. *Голубев Д.В.* Об обходе графов автоматами с одной нестираемой краской // Интеллектуальные системы. – 1999. – Т. 4, вып. 1–2. – С. 243–272.
10. *Анджан А.В.* Возможности автоматов при обходе плоскости // Проблемы передачи информации. – 1983. – Т. 19, вып. 3. – С. 78–89.

References

1. Kilibarda, G., Kudryavtsev, V.B., Ušćumlić, Š. (2003) Independent systems of automata in labyrinths. *Discrete Mathematics and Applications, 13*(3), 221-255.
2. Kilibarda, G., Kudryavtsev, V.B., Ušćumlić, Š. (2003) Collectives of automata in labyrinths. *Discrete Mathematics and Applications, 13*(5), 429-466.

3. Dudek, G., Jenkin, M. (2010). *Computational Principles of Mobile Robotics.* Cambridge University Press.
4. Blum, F., Kozen, D. (1978). On the power of the compass (or, why mazes are easier to search than graphs). In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science (SFCS '78)* (pp. 132–142). IEEE Computer Society, Washington, DC, USA.
5. Donald, B.R. (2012). The Compass That Steered Robotics. In *Logic and Program Semantics* (pp. 50-65). Lecture Notes in Computer Science, Vol. 7230. Springer.
6. Kurganskyy, O.M., Sapunov, S.V. (2016). On the directional movement of a collective of automata without a compass on a one-dimensional integer lattice. *Izv. Saratov Univ. (N.S.), Ser. Math. Mech. Inform., 16*(3), 356-365 (in Russian).
7. Bondy, J.A., Murty, U.S.R. (2008). *Graph Theory.* Springer-Verlag London.
8. Grunsky, I.S., Sapunov, S.V. (2012). Reconstruction of the Graph of Operating Environment of Mobile Robot by Vertex Labeling Sufficient for Further Navigation. *Iskusstvennyj intellekt, 4*, 420-428 (in Russian).
9. Golubev, D.V. (1999). On graph traversal by automata with single indelible paint. *Intellektualnye sistemy, 4*(1–2), 243-272 (in Russian).
10. Andzhan, A.V. (1983). Capabilities of Automata upon Traversing a Plane. *Problems of Information Transmission, 19*(3), 235-244.

С.В. Сапунов

Колективи автоматів на детермінованому нескінченому графі решітки.

Автомати, які пересуваються по графах, є математичною формалізацією автономних мобільних агентів з обмеженою пам'яттю, що функціонують у дискретних середовищах. В рамках цієї моделі виникла та інтенсивно розвивається велика область досліджень поведінки автоматів в лабіринтах (лабіринти є орієнтованими графами спеціального виду, які укладено на двовимірній цілочисловій решітці). Дослідження з цього напрямку отримали великий спектр застосувань, наприклад, в задачах аналізу зображень та навігації мобільних роботів. Автомати, що функціонують у лабіринтах, можуть розрізняти напрямки, тобто вони мають компас. У цій роботі розглядається вершинна розмітка графа квадратної решітки, завдяки якій скінчений автомат без компаса може пересуватися по графі у довільному напрямку. Автомат отримує на свій вхід позначки усіх вершин із замкненого околу поточної вершини та пересувається поміж суміжними вершинами, обираючи цільову вершину за її позначкою. У роботі запропоновано так звану мінімальну детерміновану прохідну розмітку, яка задовольняє шуканій властивості. Розмітка називається детермінованою, якщо усі вершини із замкненого околу будь-якої вершини графа мають різні позначки. Доведено, що мінімальна детермінована прохідна вершинна розмітка графу квадратної решітки потребує п'ять різних типів позначок. Також мінімальні детерміновані прохідні розмітки підграфів графу квадратної решітки, ступінь яких менше чотирьох. Основною задачею про автомати та лабіринти є задача про побудову скінченого автомата, який обходить заданий клас лабіринтів. Казатимемо, що автомат обходить нескінчений граф, якщо він відвідує будь-яку довільно обрану вершину графа за скінчений час. Доведено, що колектив, який складається з одного автомата та трьох каменів, обходить нескінчений граф квадратної решітки з заданою на ньому мінімальною детермінованою прохідною розміткою, а ніякий колектив з меншим числом каменів цього зробити не може. Каміння розглядається як автомати найпростішого виду, пересування яких цілком визначається іншими автоматами колективу. Результати стосовно обходу нескінченого позначеного графа квадратної решітки збігаються з результатами

А.В. Анджана стосовно обходу нескінченого мозаїчного лабіринту без дірок. Таким чином граф квадратної решітки після побудови на ньому мінімальної детермінованої прохідної розмітки та фіксації двох пар протилежних напрямків стає аналогом нескінченого мозаїчного лабіринту без дірок.

Ключові слова: *граф квадратної решітки, графохідний автомат, розмітка вершин, колектив автоматів.*

Institute of Applied Mathematics and Mechanics
of the NAS of Ukraine, Slavyansk
sapunov_sv@yahoo.com

Received 19.06.19