MULTIDIMENSIONAL APPROACH TO ROBOT CONTROL AND EFFECTIVE REPRESENTATION OF THE OPERATING ENVIRONMENT

KHELEMENDIK, R.; KUVSHINOV, S.;
KHARIN, K. & PRYANICHNIKOV, V.

Abstract: *The article develops a multidimensional approach to planning the motion of robots on a flat discrete field of dimension n x m, representing a visual, formalized and effective description of the operating environment. The application of the IGEC methodology in four-dimensional space for visualization, algorithmization, standardization and numerical analysis of sensor and control systems of robots is shown. An algorithm for transition from rectangular cell fields to special truncated n-dimensional cubes is proposed, the use of which allows for optimization of software modeling of robot motion. An algorithm for logical scaling and detailing of the planned trajectory of a robot is described, which, with an increase in the number of cells, provides significant savings in time and memory compared to the conventional representation of the field on a coordinate plane. Some ways of applying this approach in robotics are shown.*

Key words: *Mobile collaborative robots, presentation of information about the surrounding operating environment, n-dimensional cubes, using IGEC-methodology.*

Authors´ data: Univ.Prof. Dipl.-Ing. **Pryanichnikov**, V[alentin]*, * KIAM Russian Academy of Sciences, MSTU “Stankin” and IINET RSUH, Moscow 127055, v.e.pr@yandex.ru

This Publication has to be referred as: Khelemendik, R[oman]; Kuvshinov, S[ergey]; Kharin, K[onstantin] & Pryanichnikov, V[alentin] (2025). Multidimensional Approach to Robot Control and Effective Representation of the Operating Environment, Chapter 06 in DAAAM International Scientific Book 2025, pp.073-088, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-45-7, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2025.06

1. Introduction

Planning the trajectory of a mobile robot on a relatively large grid is a typical robotics problem. The trajectory is constructed from the current position (i_0, j_0) through adjacent cells to the target cell (i_g, j_g), with diagonal transitions considered forbidden. Obstacles between adjacent cells are modeled as «walls», prohibiting transitions between them based either on a priori information (a problem with a known map) or on current sensor readings. The paper will show that plotting trajectories using the n-dimensional cube formalism is significantly more efficient.

We assume that the number of cells is 2^n, and the field has dimensions 2^m x 2^{n-m}. Let's consider ways to number cells using Boolean sets of length n and numbers from 0 to 2^{n-1} so that adjacent (vertically or horizontally) fields differ by exactly one binary digit. This numbering helps save time and memory when formalizing movement on a discrete plane. This, in particular, allows us to change just one bit in software to move the robot one cell in a straight line; we also gain new capabilities for modeling robots in flat labyrinths.

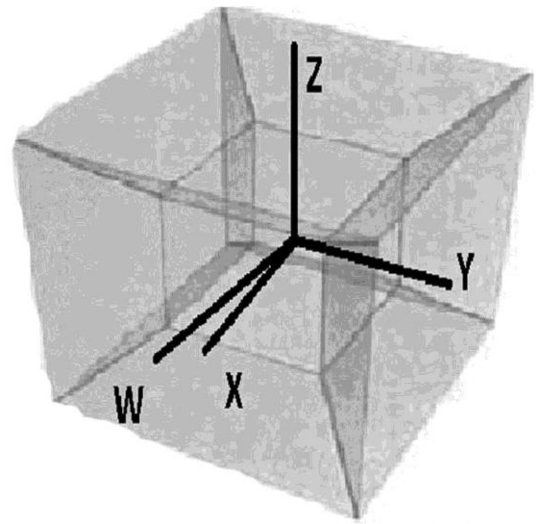
For problems where fast and efficient transitions between states of objects are important, Gray codes (Perezhogin & Bykov, 2022), (Gray) are used. Their distinctive feature is that only one position changes when transitioning from one code combination to another. Gray codes have been studied in terms of Hamiltonian cycles in graphs and n-dimensional boolean cubes (Perezhogin & Bykov, 2022), (Gilbert, 1958), (Perezhogin, 2021), in the development of combinatorial algorithms (Reingold et al., 1977), for eliminating communication errors over communication channels (Goddyn et al., 1988), in the design of hard disks and databases (Goddyn et al., 1988), (Faloustos, 1988).

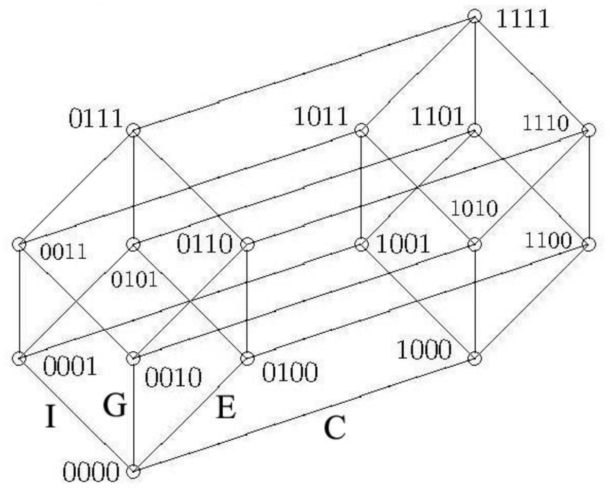
This paper proposes the construction of specialized truncated n-dimensional cubes, planar rectangular fragments, and their combinations for new applications of n-dimensional cubes and Gray codes, in particular for robot motion planning in cellular environments. This is important for representing the operating environment (such as a labyrinths) in the robot's memory. These principles can be used to construct an effective description of an algorithm for traversing, for example, hospital wards using a logistics robot.

2. Methodology of the IGEC in four-dimensional space

The IGEC methodology is a combination of the so-called «ancient Indo-Greek-Egyptian-Chinese» traditions and methods of constructing proofs (IGEC) (Kirilchenko & Pryanichnikov, 2013), (Pryanichnikov & Khelemendik, 2016), (Pryanichnikov et al., 2015), (Pryanichnikov et al., 2014). Let us agree to consider (Kirilchenko & Pryanichnikov, 2013), that the *ancient Indian* tradition relies on the senses and presents contemplation and intuition as methods of research and evidence. The *ancient Greek* tradition relies on logical inference, rational reasoning, and the methods of the natural sciences. The *ancient Egyptian* tradition relies on authority: its assertions are invariably true, and the precise execution of its principles is its method.

The *ancient Chinese* tradition relies on the meticulous execution of a large number of small operations, and its method involves numerous calculations and manipulations, which is characteristic of modern computing. A four-dimensional representation of the IGEC methodology is shown in Fig. 1.





In continuous space	In the discrete case (4-dimensional cube)

Fig. 1. Representation of IGEC methodology in four-dimensional space

For a continuous four-dimensional space, its two-dimensional projections were considered (Pryanichnikov & Khelemendik, 2016) – studies of objects by different pairs of traditions. In this paper, we focus on a discrete space (the Boolean four-dimensional cube), without imposing general restrictions on the use of all four approaches. We denote the coordinate axes with the corresponding binary digits (in ascending order) by the letters I, G, E, and C, respectively, conventionally interpreting them as follows: I – clarity, G – reasoning, E – regulation, C – calculation. Then, for example, a description of the entire design complex for a mobile robot can be represented as follows. Let's assume that within one dimension, the following points on a 4-dimensional cube correspond to: <0001> (visualization of movement in a digital twin), <0010> (construction of movement algorithms), <0100> (construction of CAD drawings in accordance with standards), <1000> (numerical and in-kind experiments). The addition of the next dimension corresponds to a transition along the cube edge in the corresponding direction, and a comprehensive and all-encompassing study of the mobile device corresponds to the point <1111>. This methodology of study and research allows us to identify gaps and formalize contradictions, both for students studying robotics and for specialists. This methodology allows us to identify gaps and formalize contradictions, both for developers and students studying robotics. This is achieved, for example, by detailing the numerical experiments: according to the structure (Fig. 1 on the right), on a 4-dimensional cube, such calculations are divided into 8 groups (points of the right three-dimensional cube) and the possible discrepancy in the results for different groups allows us to focus attention on the causes of inconsistencies in the data, which may not be related to hardware and software features.

3. Definitions and designations

A simple path in a graph or *a simple chain* of length k>1 on a graph G will be called a sequence (set) $C=<v_1,u_1,\ldots,u_{k-1},v_k>$, where v_i belongs to the set of vertices V, and u_i to the set of edges U, and none of the elements of the sequence is repeated (if there are repeating elements in this sequence, the word “simple” is omitted). Similarly, a simple cycle in a graph is a simple chain C for which $v_1 = v_k$.

The set of all sets $<\alpha_1...\alpha_n>$ of binary (Boolean) values is called an *n-dimensional (Boolean) cube* of size n and is denoted by B^n. For sets of values we use the following notations: $\boldsymbol{\alpha}^n=<\alpha_1...\alpha_n>$. *The distance* $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)$ between vertices $\boldsymbol{\alpha}^n$ and $\boldsymbol{\beta}^n$ is the number of digits in which they differ: $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=\Sigma^n_{i=1}|\alpha_i-\beta_i|$. *The number* $\eta(\boldsymbol{\alpha}^n)$ *of a set* $\boldsymbol{\alpha}^n$ is the number whose binary representation is $\boldsymbol{\alpha}^n$: $\eta(\boldsymbol{\alpha}^n)=\Sigma^n_{i=1}2^{n-i}\alpha_i$. Geometrically, the Boolean cube B^n is a graph consisting of 2^n vertices, and any two of its vertices $\boldsymbol{\alpha}^n$ and $\boldsymbol{\beta}^n$ are connected by an edge if and only if $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=1$. *A Hamiltonian cycle* (HC) in a graph is a sequence of all its vertices, connected by edges and not repeating, in which the last vertex is connected by an edge to the first. If the vertices of B^n are denoted by binary words (sets) of length n, then the HC in B^n is also called the Gray code [2, 3]. Fig. 2 shows flat projections of n-dimensional cubes for n=1,2,3,4, as well as some HCs (arrows).

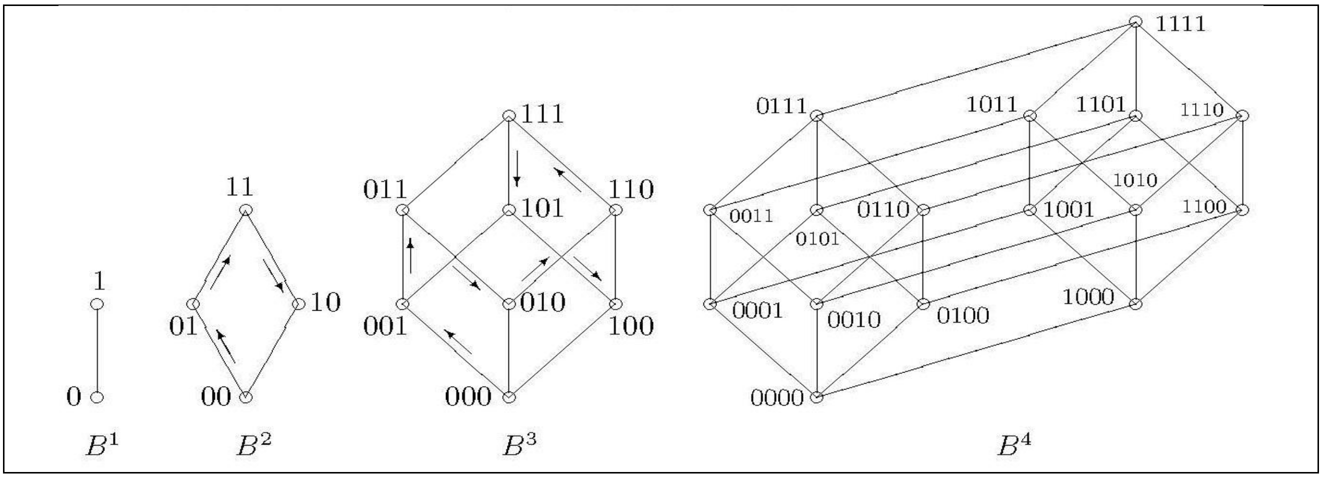


Fig. 2. n-dimensional cubes B^n, n≤4, see also [17]

A plane rectangular (m,n)-*fragment* ($PF_{(m,n-m)}$) of an n-dimensional cube is a subgraph B^n such, that its vertices are mapped to the cells of a rectangle of size 2^m x 2^{n-m}, the edges of the subgraph connect vertically and horizontally adjacent (neighbouring) cells of this rectangle, and the remaining edges from B^n are removed. In this case, the cells of a 2^m x 2^{n-m} rectangle can be numerated with numbers representing binary codes (of vertices). Thus, the HC in Fig. 2 for B^2 is the basis for $PF_{(0,2)}$ – a rectangle of size 1 x 4, the cells of which are sequentially numbered with the numbers 0, 1, 3, 2; and if we restore the edge <00>–<10> in this graph, we obtain $PF_{(1,1)}$ – a rectangle of size 2 x 2 (square), consisting of the rows [0,2], [1,3].

We note that in order to maintain the integrity of the presentation, the following sections 4-6 are presented close to the text of our publication [17], while these results are significantly supplemented and continued in sections 7-8.

4. Construction of the cellular environment and PF (plane fragments) for the first values of n.

Obviously, for n=1 there is only one PF, namely $PF_{(0,1)}$ with a corresponding rectangle of size 1 x 2, and “almost identically” (indistinguishably) numbered cells – in the order [0,1] or [1,0]. In the case of n=2, as shown above, there are only two PFs: $PF_{(0,2)}$ (strip) and $PF_{(1,1)}$ (square), for which the indistinguishable cell numberings are given above. We move on to the case of n≥3. The statement is obvious:

Statement 1. *Every* HC *in* B^n *uniquely corresponds to some* $PF_{(0,n)}$, *and vice versa*.

If in HC for B^n we can recover (n-1) edges, then we get $PF_{(1,n-1)}$.

Statement 2. *For* n≥3, *not every* HC *in* B^n *has a corresponding* $PF_{(1,n-1)}$.

Indeed, for n≥3 there is HC (2,0,4,5,1,3,7,6). However, to obtain $PF_{(1,2)}$, we can restore edges (2,6) and (5,1), but we cannot restore edges (0,7) and (4,3), since they are not in B^3. Similar, as well as more complex, situations arise when n>3.

A sequence of numbers $\{a_k\}$ of length s is called ↑-unimodal (↓-unimodal) if there exists k such that $a_1 < a_k > a_{k+1} > \ldots > a_s$ ($a_1 > a_k < a_{k+1} < \ldots < a_s$), 1<k<s. Thus, the sequences (0,1,3,2) and (6,7,5,4) are ↑-unimodal. Obviously, if the sequence $\{a_k\}$ ↑-unimodal (↓-unimodal), then its inverse (taken in reverse order) sequence is ↓-unimodal (↑-unimodal). By combining two oppositely directed unimodal sequences (0,1,3,2) and (4,5,7,6), we obtain $PF_{(1,2)}$, consisting of rows with these sequences (Fig. 3). Note that the ↑-unimodal sequence is simply unimodal (Yablonskii, 1986)., p. 178), so in our definition, unimodality is somewhat specified (clarified) for this sequence, whether it is increasing or decreasing.

4	5	7	6
0	1	3	2

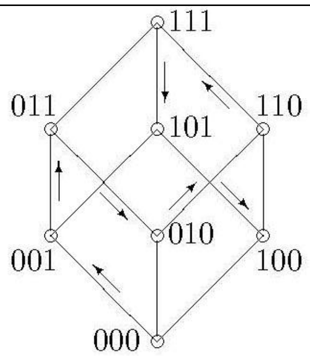


Fig. 3. $PF_{(1,2)}$, B^3 and HC in B^3

Further, unimodal sequences of numbers in rows (columns) will be called unimodal rows (columns). It is easy to see that both unimodal rows form a GC in the B^3 subgraphs, which are B^2 cubes (squares) and represent boolean functions depending only on the second and third variables.

Statement 3. *For* B^4 *there exists a* $PF_{(2,2)}$, *such that the sequences of all its rows and columns are unimodal*.

One of these $PF_{(2,2)}$ and its corresponding B^4, from which the edges unnecessary for the 4x4 square (a truncated four-dimensional cube) are removed, are shown in Fig. 4.

8	10	14	12
9	11	15	13
1	3	7	5
0	2	6	4

1000	1010	1110	1100
1001	1011	1111	1101
0001	0011	0111	0101
0000	0010	0110	0100

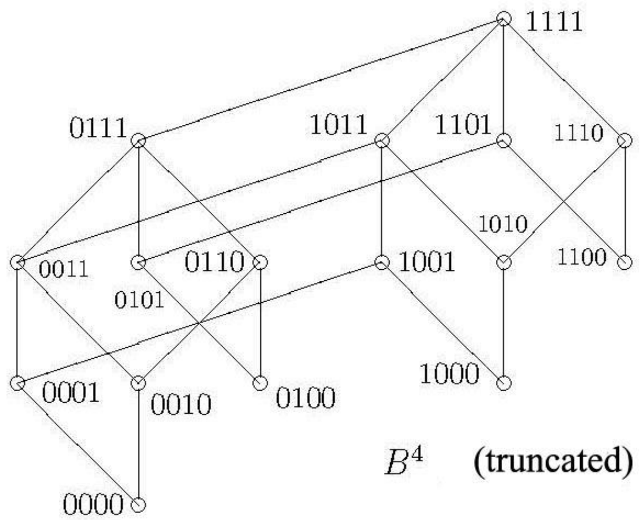


Fig. 4. $PF_{(2,2)}$ and truncated B^4

It is easy to see, that in this 4x4 square, all rows are unimodal in the second and third digits, and the columns are in the first and fourth. More specifically, from this we get, that the horizontal movement between the second and third verticals corresponds to a change in the second digit, and the movement between the first and second or third and fourth verticals corresponds to a change in the third digit. Similarly, a vertical movement between the second and third horizontals corresponds to a change in the first digit, and a movement between the first and second or third and fourth horizontals corresponds to a change in the fourth digit.

The weight $\omega(\boldsymbol{\alpha}^n)$ *of a Boolean set* $\boldsymbol{\alpha}^n$ is the number of its unit components: $\omega(\boldsymbol{\alpha}^n) = \Sigma^n_{i=1}\, \alpha_i$. Then the weights in the unimodal rows (x_2 and x_3) and columns (x_1 and x_4) also form a unimodal sequence: 0,1,3,2.Thus, instead of the $PF_{(2,2)}$ shown in Fig.3, other 4x4 squares can be constructed using forward and reverse unimodal sequences of weights for various pairs of digits. Note that to obtain $PF_{(1,3)}$ from $PF_{(2,2)}$, it is sufficient to rotate 270 degrees counter clockwise relative to the midpoint of the left side of the 4x4 square of the two upper rows. Then in $PF_{(1,3)}$ we obtain rows consisting of the concatenation of the unimodal rows (13,15,11,9,1,3,7,5) and (12,14,10,8,0,2,6,4), and in the truncated cube B^4 we remove 4 edges corresponding to the “gaps” (due to rotation), and restore (from the full B^4) the edge (8,0). Continuing this process (cutting along the central horizontal line, rotating 270 degrees and restoring one edge), from $PF_{(1,3)}$ we obtain $PF_{(0,4)}$.

5. Construction of the cellular environment and PF for n≤6.

To go from n=4 to n=5, simply duplicate the $PF_{(2,2)}$, mirroring it relative to the vertical centerline, increasing each number by 16, and positioning this PF to the right of the original. Then, for all adjacent cells in both $PF_{(2,2)}$, the condition $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=1$ remains satisfied, and for the fourth and fifth columns, horizontal adjacency means a difference of only one digit – the most significant one (Table 1).

Recording cells by natural numbers							
8	10	14	12	28	30	26	24
9	11	15	13	29	31	27	25
1	3	7	5	21	23	19	17
0	2	6	4	20	22	18	16
Recording cells as binary numbers							
01000	01010	01110	01100	11100	11110	11010	11000
01001	01011	01111	01101	11101	11111	11011	11001
00001	00011	00111	00101	10101	10111	10011	10001
00000	00010	00110	00100	10100	10110	10010	10000

Table 1. Cell numbering in $PF_{(2,3)}$ (rectangle of size 2^m x 2^{n-m}, m=2, n=5 , i.e. the size of rectangle is 4x8).

When moving from n=5 to n=6, we duplicate the constructed $PF_{(2,3)}$ and perform similar actions of mirror symmetry relative to the horizontal midline, increasing the numbers by 2^5=32, and positioning the new PF above the original. An example of such a $PF_{(3,3)}$ and the associated truncated cube B^6 are shown in Table 2 and Fig. 5.

Recording cells by natural numbers							
34	32	36	38	54	52	48	50
35	33	37	39	55	53	49	51
43	41	45	47	63	61	57	59
42	40	44	46	62	60	56	58
10	8	12	14	30	28	24	26
11	9	13	15	31	29	25	27
3	1	5	7	23	21	17	19
2	0	4	6	22	20	16	18
Recording cells as binary numbers							
100010	100000	100100	100110	110110	110100	110000	110010
100101	100001	100101	100111	111101	110101	110001	110011
101011	101001	101101	101111	111111	111101	111001	111011
101010	101000	101100	101110	111110	111100	111000	111010
001010	001000	001100	001110	011110	011100	011000	011010
001011	001001	001101	001111	011111	011101	011001	011011
000101	000001	000101	000111	011101	010101	010001	010011
000010	000000	000100	000110	010110	010100	010000	010010

Table 2. Cell numbering in $PF_{(3,3)}$

This $PF_{(3,3)}$ is not exactly an extension of the previously constructed $PF_{(2,3)}$, since in the starting $PF_{(2,2)}$ we chose oppositely directed unimodalities: rows are ↓-unimodal, and columns are ↑-unimodal.

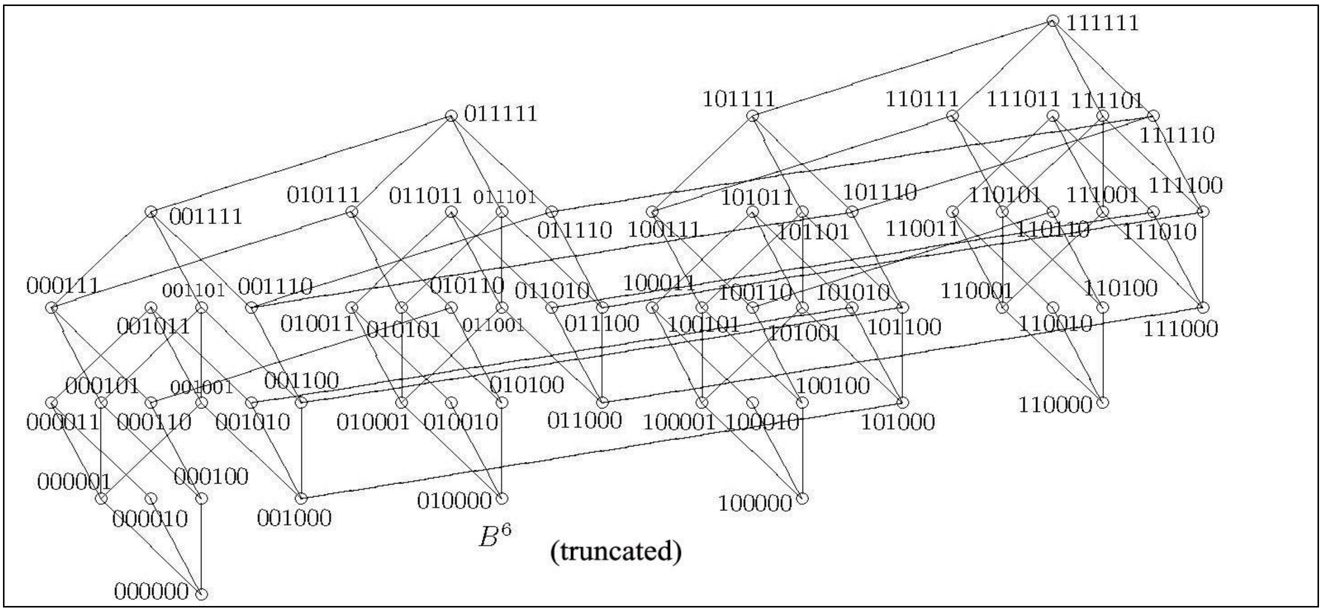


Fig. 5. Truncated six-dimensional Boolean cube

6. Construction of the cellular environment and PF for n>6.

We will construct the $PF_{(4,4)}$ for n=8, and its lower half will also be the $PF_{(3,4)}$ for n=7. First, we will slightly modify the original $PF_{(3,4)}$ for n=4, according to which we will construct the $PF_{(3,3)}$ for n=6 (see Table 3) as was done above.

Recording cells by natural numbers							
32	33	37	36	52	53	49	48
34	35	39	38	54	55	51	50
42	43	47	46	62	63	59	58
40	41	45	44	60	61	57	56
8	9	13	12	28	29	25	24
10	11	15	14	30	31	27	26
2	3	7	6	22	23	19	18
0	1	5	4	20	21	17	16
Recording cells as binary numbers							
100000	100001	100101	100100	110100	110100	110001	110000
100010	100011	100111	100110	110110	110111	110011	110010
101010	101011	101111	101110	111110	111111	111011	111010
101000	101001	101101	101100	111100	111101	111001	111000
001000	001001	001101	001100	011100	011101	011001	011000
001010	001011	001111	001110	011110	011111	011011	011010
000010	000011	000111	000110	010110	010111	010011	010010
000000	000001	000101	000100	010100	010101	010001	010000

Table 3. Unimodal cell numbering in $PF_{(3,3)}$

The main difference between this PF and the previous $PF_{(3,3)}$ (see Table 2) lies precisely in the lower left 4x4 square: previously, the rows were unimodal in digits 2 and 3, and the columns in digits 1 and 4, but now the rows are unimodal in even digits (2, 4), and the columns in odd digits (1, 3).

When moving to the new $PF_{(3,3)}$, unimodality in row concatenations is also preserved for even digits, and unimodality in column concatenations is preserved for odd digits. Continuing the transition from n=6 to n+1 in width, and then to n+2 in height, as was previously done for n=4, we arrive at $PF_{(4,4)}$ – a 16x16 square consisting of 16 4x4 squares (Table 4).

The cells in this field contain two-digit numbers written in hexadecimal notation. The first number represents the position of a 4x4 square, and the second represents the cell's position within the 4x4 square (Table 5). The cells of this field contain two-digit numbers written in hexadecimal notation. In this case, the first number characterizes the position of the 4x4 square, and the second characterizes the location of the cell already in the 4x4 square (Table 5, the symbols * denote arbitrary hexadecimal numbers, and $*^4$ – four arbitrary binary digits).

80	81	85	84	94	95	91	90	D0	D1	D5	D4	C4	C5	C1	C0
82	83	87	86	96	97	93	92	D2	D3	D7	D6	C6	C7	C3	C2
8A	8B	8F	8E	9E	9F	9B	9A	DA	DB	DF	DE	CE	CF	CB	CA
88	89	8D	8C	9C	9D	99	98	D8	D9	DD	DC	CC	CD	C9	C8
A8	A9	AD	AC	BC	BD	B9	B8	F8	F9	FD	FC	EC	ED	E9	E8
AA	AB	AF	AE	BE	BF	BB	BA	FA	FB	FF	FE	EE	EF	EB	EA
A2	A3	A7	A6	B6	B7	B3	B2	F2	F3	F7	F6	E6	E7	E3	E2
A0	A1	A5	A4	B4	B5	B1	B0	F0	F1	F5	F4	E4	E5	E1	E0
20	21	25	24	34	35	31	30	70	71	75	74	64	65	61	60
22	23	27	26	36	37	33	32	72	73	77	76	66	67	63	62
2A	2B	2F	2E	3E	3F	3B	3A	7A	7B	7F	7E	6E	6F	6B	6A
28	29	2D	2C	3C	3D	39	38	78	79	7D	7C	6C	6D	69	68
08	09	0D	0C	1C	1D	19	18	58	59	5D	5C	4C	4D	49	48
0A	0B	0F	0E	1E	1F	1B	1A	5A	5B	5F	5E	4E	4F	4B	4A
02	03	07	06	16	17	13	12	52	53	57	56	46	47	43	42
00	01	05	04	14	15	11	10	50	51	55	54	44	45	41	40

Table 4. Unimodal cell numbering in $PF_{(4,4)}$

8*	9*	D*	C*	*8	*9	*D	*C
A*	B*	F*	E*	*A	*B	*F	*E
2*	3*	7*	6*	*2	*3	*7	*6
0*	1*	5*	4*	*0	*1	*5	*4
Binary numbering				Binary numbering			
1000$*^4$	1001$*^4$	1101$*^4$	1100$*^4$	$*^4$1000	$*^4$1001	$*^4$1101	$*^4$1100
1010$*^4$	1011$*^4$	1111$*^4$	1110$*^4$	$*^4$1010	$*^4$1011	$*^4$1111	$*^4$1110
0010$*^4$	0011$*^4$	0111$*^4$	0110$*^4$	$*^4$0010	$*^4$0011	$*^4$0111	$*^4$0110
0000$*^4$	0001$*^4$	0101$*^4$	0100$*^4$	$*^4$0000	$*^4$0001	$*^4$0101	$*^4$0100
Big square				Little square			

Table 5. Numbering of cells of the large and small square in $PF_{(4,4)}$

Thus, any horizontal movements correspond only to a change in even digits, and any vertical movements correspond only to a change in odd digits. In this case, a change in the 1st (2nd) digit is a transition between horizontals (verticals) 8 and 9; a change in the 3rd (4th) digit is a transition between horizontals (verticals) 4-5, 12-13; a change in the 5th (6th) digit is a transition between horizontals (verticals) 2-3, 6-7, 10-11, 14-15; a change in the 7th (8th) digit is a transition between horizontals (verticals) 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16. The algorithm for constructing a PF from $n=2k$ to $n+1$ in width, and then to $n+2$, also works in the general case $n>8$.

7. Transition from traditional to multidimensional representation of the cellular environment.

It is common to represent a robot's position in a cellular environment as a pair of coordinates <x,y>, where x is the vertical axis and y is the horizontal axis. The simplicity and clarity of this representation are beyond doubt. Moving to an adjacent cell then means adding or subtracting one from the x or y variable. It should be noted, that repeated execution of such simple arithmetic operations in computing devices with a multidimensional representation of the cellular environment requires significantly less time and memory. Let us consider the important case $n=8$ (generalizable for $n=2k$, $k>4$), when one byte is sufficient to store the current position of an object (robot) for both representations of the cellular environment. In the traditional numbering (from left to right, and then from bottom to top), the number η of a cell <x,y> in the form of a byte is represented as yx, where $\eta=16y+x$, $0 \le x,y \le 15$, from which it follows that $y = \lfloor \eta/16 \rfloor + x$, $x = \eta \bmod 16$. In the multidimensional case (Table 5) for the field $\boldsymbol{\alpha}^8$ with the number $\eta(\boldsymbol{\alpha}^8)$ we obtain $y = 4[2\alpha_1+(\alpha_1\oplus\alpha_3)]+[2\alpha_5+(\alpha_5\oplus\alpha_7)]$, $x = 4[2\alpha_2+(\alpha_2\oplus\alpha_4)]+[2\alpha_6+(\alpha_6\oplus\alpha_8)]$, which allows (if necessary) to quickly move from a multidimensional coordinate system to a normal one, and vice versa (the symbol $\oplus$ denotes addition modulo 2).

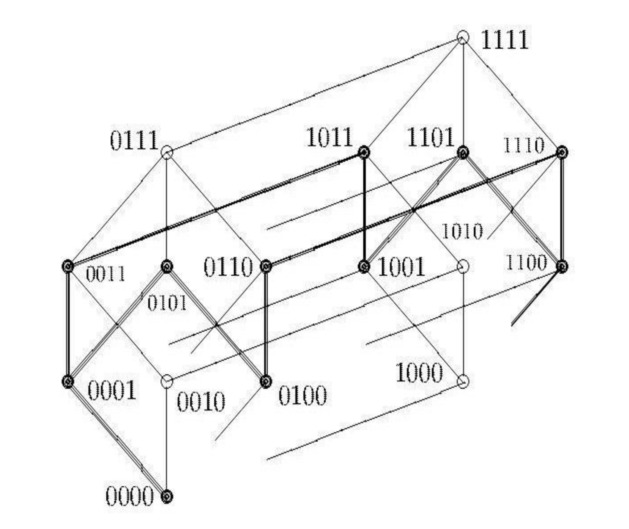
In the next actual task, the mobile robot needs to find a path from cell <0,0> to cell <15,15>. Any shortest path from <0,0> to <15,15> consists of 15 elementary vertical and 15 horizontal movements. In the multidimensional representation, an elementary movement results in a speed savings of some δ, since changing one bit is faster than adding or subtracting one. For 30 movements, this already yields a savings of 30δ. We achieve memory savings if we want to remember a specific route from <0,0> to <15,15>. With the traditional representation, we need to remember a sequence of coordinates (bytes), i.e., 30 bytes, whereas in the multidimensional representation, 30x3 bits are sufficient. At the same time, the hierarchy of bits by seniority allows us to obtain instant detailed information about the robot even when only one bit changes: for example, the inversion of one of the most senior bits immediately means that the robot has moved to the other half of the square (vertical or horizontal).

8. Formalism of representing a labyrinth in the robot's memory.

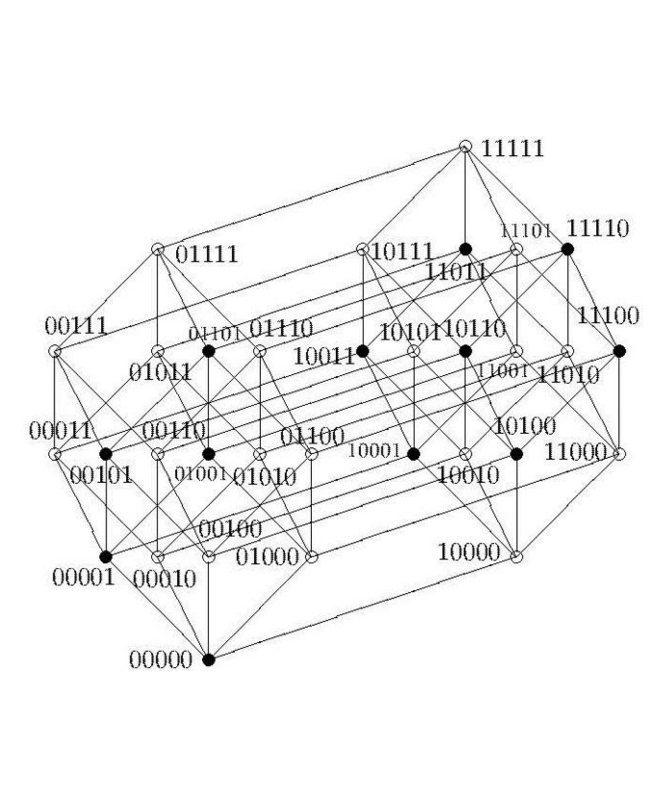
We will represent the labyrinth in the cellular environment as a rectangular fragment of the PFin which transitions between some adjacent cells may be forbidden. Let's first consider a field of size $2^{n/2}$ x $2^{n/2}$, which we represent as a $PF_{(n,n)}$, as shown above. Then a specific labyrinth is defined by listing all pairs of adjacent cells between which transitions are allowed.

Since the total number of possible transitions already at n=8 amounts to hundreds, listing forbidden pairs of adjacent cells as pairs of their coordinates is only suitable for a small number of prohibitions; moreover, it is not visually clear. Using a multidimensional representation of the cellular environment, forbidden pairs are specified by specifying the bits responsible for the transitions. Thus, for n=8, a set of values $\boldsymbol{\beta}^{\mathbf{8}}$=<$\beta_1,\dots,\beta_8$> is first formed such that $\beta_j=1$ if and only if for the case of at least one change of the j-th digit there is an allowed transition, and then an additional set ξ is formed, which allows us to specify all such cases. The representation of permissions becomes clear if the labyrinths are represented as truncated cubes, in which the permitted transitions (edges) are highlighted with bold lines (or in green). Further refinement of permitted transitions and formalization of routes can be achieved using Boolean functions with a small number of variables and propositional logic. Let's perform this refinement and formalization using a specific 4x4 square as an example (Fig. 6).

We call a *semi-open* PF if exits are possible for all cells in the top row and right column. It is easy to see that for a semi-open $PF_{(n,n)}$, the total number of transitions between adjacent cells, including exits, is 2^{n+1}. In a labyrinthfor a semi-open $PF_{(4,4)}$, the robot initially finds itself in cell <0,0> and he needs to get out of the labyrinth, taking into account restrictions on certain transitions between adjacent cells. The equivalent of a description of restrictions is an indication of permitted transitions between adjacent cells, which is given on the left side (Fig. 6) – both as a flat map and as a labeled «semi-truncated» boolean cube B^4.



y↑	no	no	no	yes	
3	1000	1001	1101	1100	no
2	1010	1011	1111	1110	no
1	0010	0011	0111	0110	no
0	0000	0001	0101	0100	no
x→	0	1	2	3	



Allowed transitions in B^4 | Allowed transitions in B^5 (points)

Fig. 6. Transition from B^4 to B^{4+1} to describe the allowed transitions in the labyrinth

In this cube, the edges from vertices numbered 4, 6, 8-9, 12-14 are not completely removed, since they correspond to exits from the upper horizontal and the right vertical. Allowed transitions (edges) in the labyrinth are highlighted in bold. Each transition (including the forbidden one) between cells $<\boldsymbol{\alpha}^4,\boldsymbol{\alpha}'^4>$, where $\boldsymbol{\alpha}'^4$ is located to the right of $\boldsymbol{\alpha}^4$, or $\boldsymbol{\alpha}^4$ is located on the rightmost vertical, is associated with the binary set $\boldsymbol{\beta}^5=<0,\alpha_1,\ldots,\alpha_4>$; and if $\boldsymbol{\alpha}'^4$ is located above $\boldsymbol{\alpha}^4$, or $\boldsymbol{\alpha}^4$ is located on the upper vertical, then the transition is associated with the binary set $\boldsymbol{\beta}^5=<1,\alpha_1,\ldots,\alpha_4>$.

Then the 12 allowed transitions in the flat labyrinth (Fig. 6 on the left) exactly correspond to the 12 points of the boolean cube B^5. These points are unit values of a boolean function of 5 variables and can be defined analytically: for example, by the formula $f(z^5) = \neg z_1\neg z_4(\neg z_2\neg z_3\neg z_5 \vee z_2 z_5) \vee z_1(\neg z_3 z_5(\neg z_2 \vee z_4) \vee z_2 z_3\neg z_4\neg z_5)$, which defines a map of the labyrinth. There are only two suitable (and non-redundant – not containing cycles) routes, which, in particular, can be defined by sequences of points from B^5. However, it is clearer and more economical to write them as sequences of changes in digit numbers when transitions along edges in B^4: <4243131> (3 cells to the right, 4 cells up) and <4313241> (to the right, 3 cells up, 2 cells to the right, up). By decreasing the digit numbers by one and moving to binary representation, we obtain boolean sets of length 14, i.e., the values of partially defined boolean functions of 4 variables. Thus, by defining in the first case the set <11011110 001000**> with the values 0 and 1 in the penultimate and last digits, we obtain a compact definition of the required route by the formula $f(z^4) = z_1\oplus(\neg z_3 \vee (z_2\oplus z_4))$.

For n>4, we represent the labyrinth as nested semi-truncated boolean cubes (PFs) as follows. We call a $2^{n/2}$ x $2^{n/2}$ labyrinth *semi-open* if its base is a semi-open $PF_{(n,n)}$. Obviously, if all exit restrictions are present along the top horizontal and right vertical, the semi-open labyrinth (and PF) becomes a regular labyrinth. We will describe the algorithm for formalizing the labyrinth for n=8 and note that to modify it for other n it is sufficient to construct a suitable PF (see section 6) of the appropriate size in the role of a large semi-truncated Boolean cube (see Fig. 8 below). For n=8, the labyrinth has dimensions 16x16 and can be represented by a tile of sixteen 4x4 squares (see Section 6, Table 4, Table 5). Specifying the allowed transitions for each square completely determines the allowed transitions in the labyrinth. For clarity, let's consider a specific labyrinth(Fig. 7).

Our task now is to describe the map of this labyrinth—all permitted transitions between adjacent cells (including exits upward from horizontal F and to the right from vertical F). We number the cells of each of the 16 squares with hexadecimal digits in the usual way (left to right, then bottom to top). Each such square is a small labyrinth, and its map can be specified by two binary sets of length 16 (see Section 8). For example, for the leftmost square, we have the sequences <0011 0100 1010 0001> (rightward transitions) and <1110 1111 1111 1100> (upward transitions), and we write the concatenation of these sequences as a set of 4 bytes: <34 a1 ef fc>. Repeating this procedure for the remaining 15 squares and considering them as the vertices of a large square, we obtain a map of the entire labyrinth in the form of a semi-truncated 4-dimensional cube, whose vertices indicate the maps of the small labyrinths (Fig. 8).

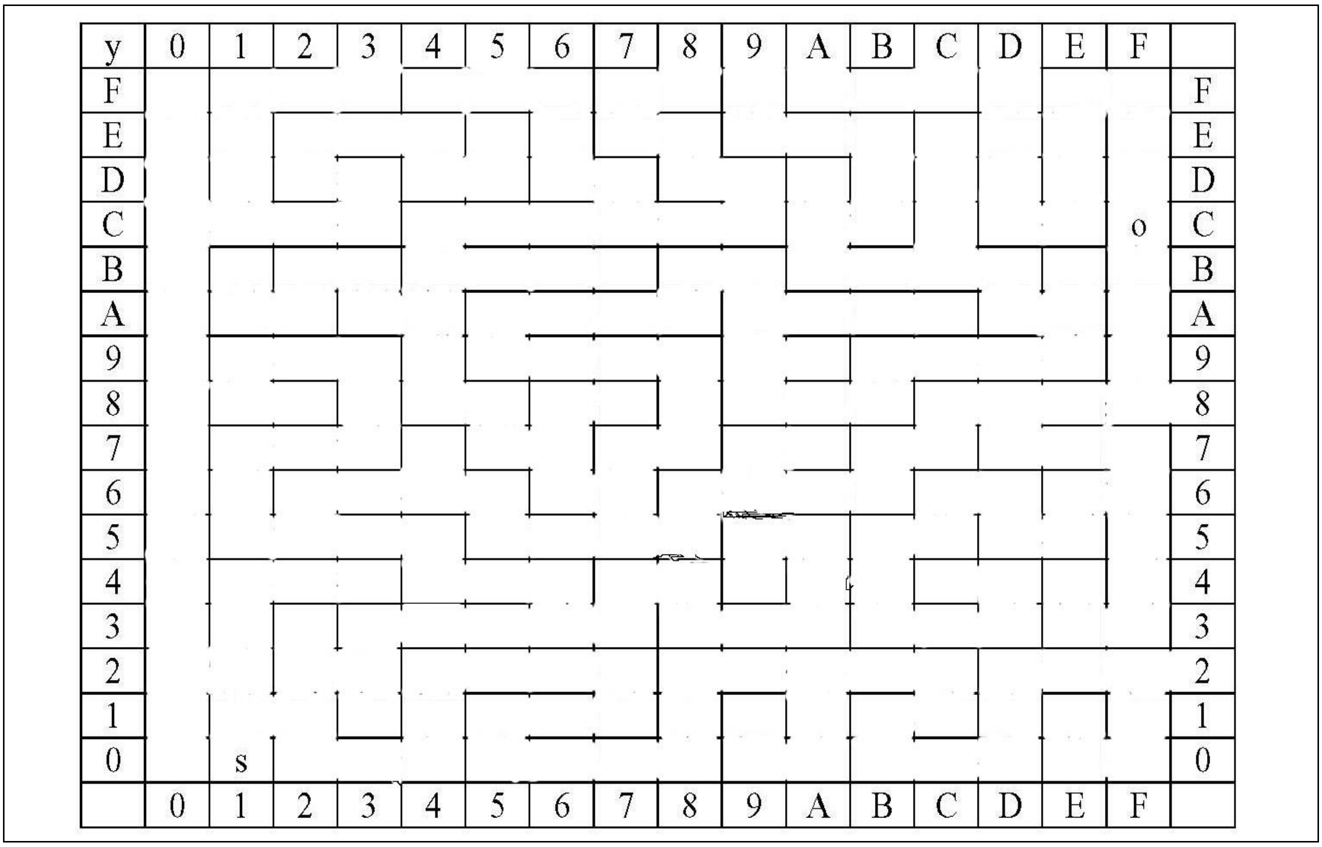


Fig. 7. Labyrinth of size 16x16.

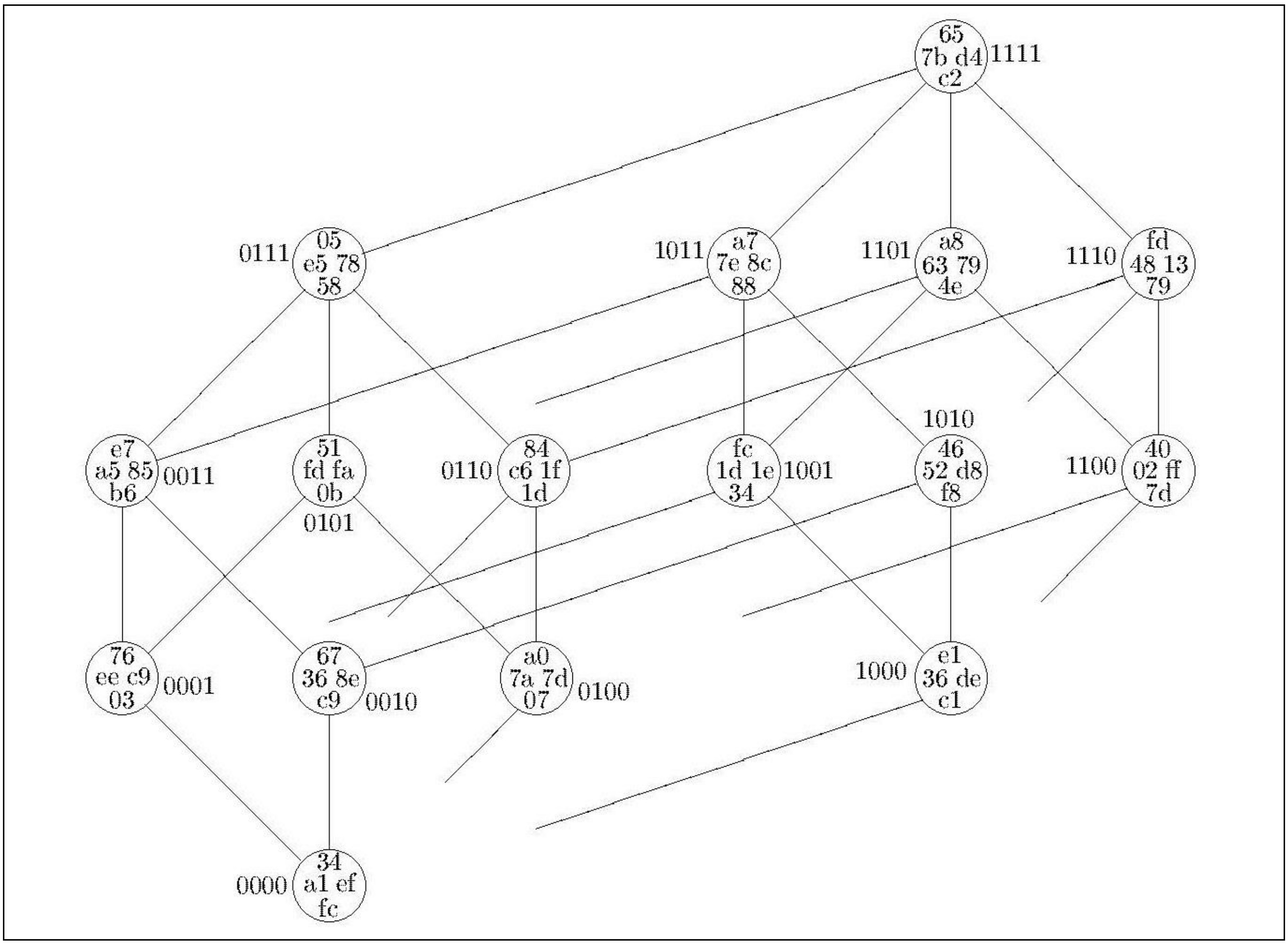


Fig. 8. Semi-truncated B^4 with embedded maps of small labyrinths

Note that in this semi-truncated B^4 (unlike the previously constructed small semi-truncated B^4, see Fig. 6), the edges represent the «gluing» points of squares and cannot always be highlighted in bold as «allowed». Indeed, for example, for the transition <0001>–<0101> in the large B^4, we can only move from square number 1 in the labyrinth to square 3 from cell <6,3>, which within this square is accessible only to three other cells on the same horizontal, while the remaining 12 cells in the square have no such access.

However, such access is possible if the exit from square 1 (via the <3,0> entrance) to square 0 is taken into account, followed by a route through it to cell <3,3> and an exit to the right of this square (to cell <4,3> of square 1). This pattern also holds true in a large labyrinth for the important practical task of constructing a route for a robot. Let's say a robot is located in cell (1,0) (marked with the symbol s), and its task is to reach cell (F,C), marked with the symbol o. Then, in a large semi-open labyrinth, it can exit and enter through arbitrary exit points. Finding such (alternate) routes significantly increases the reliability of mobile robots in extreme situations. We will also record the robot's routes through the labyrinth as sequences of numbers of the large squares they visit, as well as the exits (entrances) of the labyrinth. For our labyrinth, between cells (1,0) and (F,C), there is a single (non-redundant) route without exits: [0132376EC], and there are also routes with exits, for example: [02A8(3,F)(D,F)C], [0154(F,2)(F,8)EC].

As the dimension increases, for example with n=9, we can leave the large labyrinth the same (4x4), but then the small labyrinths will be 5-dimensional cubes, or we can leave the small labyrinths as 4x4 squares, but consider the large labyrinth as a 5-dimensional cube. Further formalization of labyrinths for n>8 can occur either by increasing the dimension of large (or small) truncated boolean cubes, or by using several levels of nesting (for example, 4-dimensional cubes).

9. Conclusion

The key point of the article is the application and development of the IGEC methodology in the multidimensional case and the proposal of a new logical representation to describe the operating environment of mobile service robots. The practical significance of these results lies in the possibility of maximum data compression (close to the theoretical limit) based on a very promising method for determining the neighborhood on a two-dimensional cellular field. Finding routes in the proposed generalized labyrinth model (semi-open labyrinth) increases the reliability of robots in extreme situations. The proposed device opens up new tools for creating effective on-board software for collaborative robots based on microprocessor implementations. It should also be noted, that the work was carried out mainly at the Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences as part of research related to the “Intelligent robotronics” project (see also the works (Davydov et al., 2017), (Pryanichnikov et al., 2021), (Aryskin et al., 2019), (Khelemendik & Pryanichnikov, 2020), (Kirilchenko & Pryanichnikov, 2013), (Pryanichnikov & Khelemendik, 2016), (Pryanichnikov et al., 2015), (Pryanichnikov et al., 2014), (Khelemendik & Pryanichnickov, 2025).

10. References

Yablonskii S.V. (1986). Vvedenie v diskretnuiu matematiku / M.: Nauka, 1986.-384s.
Perezhogin A. L., Bykov I. S. Review of constructions and properties of Gray codes // Mathematical Issues of Cybernetics, no. 20, A collection of papers, ed. by O.B.Lupanov. Moscow: Fizmatlit, 2022, pp. 41-60. URL: http://library.keldysh.ru/mvk.asp?id=2022-41 DOI: 10.20948/mvk-2022-41
Gray F. Pulse code communications.—Patent no. 2632058 US.
Gilbert E. N. Gray codes and paths on the n-cube //The bell system technical journal.— 1958.—V. 37, No 3.—P. 815–826.
Perezhogin A. L. On the spectrum of Hamiltonian cycles in the n-cube // Journal of Combinatorial Theory, Series B.—2021.—V. 151.—P. 435–464.
Reingold E. M., Nievergelt J., Deo N. Combinatorial Algorithms: Theory and Practice. — Prentice Hall College Div., 1977.
Goddyn L., Lawrence G. M., Neme h E. Gray codes with optimized run lengths // Utilitas Mathematica.—1988. — V. 34.—P. 179–192.
Faloustos C. Gray codes for partial match and range queries // IEEE Transactions on Software Engineering.— 1988.—V. 14, No 10.—P. 1381–1393.
Davydov D.V., Eprikov S.R., Kirsanov K.B., Pryanichnikov V.E. (2017). Service Robots Integrating Software and Remote Reprogramming, Proceedings of the 28th DAAAM International Symposium, pp.1234-1240, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/28th.daaam.proceedings.172.
Pryanichnikov, V[alentin]; Kolesov, Y[uriy]; Tarasov, R[adomir] & Plotnikov, A[leksey] (2021). Integration Software for a Distributed Group of Mobile Service Robots, Proceedings of the 32nd DAAAM International Symposium, pp.0311-0315, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978- 3-902734-33-4, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/32nd.daaam.proceedings.045
Aryskin A.A., Bogdanovich A.V., Davydov O., Khelemendik R., Petrakov M., Pryanichnikov V.E., (2019). Control and Interaction Algorithms for Industrial and Service Robots, Chapter 33 in DAAAM International Scientific Book 2019, pp.377-384, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-24-2, ISSN 1726-9687, Vienna, Austria DOI: 10.2507/daaam.scibook.2019.33
Khelemendik R., Pryanichnikov V. E. (2020). Contradictory Reasoning Connecting with the Problems of Intelligent Robotronics, Proceedings of the 31st DAAAM International Symposium, pp.0276-0281, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-29-7, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/31st.daaam.proceedings.038
Kirilchenko A. A., Pryanichnikov V. E., Rogozin K. V. Limits of reliability and reliability of evidence. Skepticism in mathematics, functions, traditions // Information-measuring and control systems. - 2013. - Vol. 11, No. 4. Pp. 57-65. ISSN 2070–0814.

Pryanichnikov V. E., Khelemendik R. V. IGEC Information technology and pentalogics // Computer science and information technology: Proceedings of the international conference. science. Conf. — Saratov: Publishing center «Science», 2016. Pp. 331-333.

Pryanichnikov V. E., Bielic T., Vican D., Katalinich B., Kirsanov K. B., Kuvshinov S. V., Marzanov Yu., Poduraev J. V., Khelemendik R. V., Prysev E. A., Uglesic A., Kharin K.V. Development of educational technology and the network of associated laboratories of robotarium // Information-measuring and control system. - 2015. — Vol. 13, No. 7. — Pp. 7-25. ISSN 2070–0814.

Pryanichnikov V.E., Katalinic B., Kirilchenko A.A., Khelemendik R.V., Kuvshinov S.V., Vician D., Uglesic A. New Creative Educational Technologies for Interuniversity Network. Elsevier. // Procedia Engineering, 2015. — Vol.100, — Pp.259-268. // 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014.

Khelemendik, R[oman] & Pryanichnickov, V[alentin] (2025). Cellular Environment and 4-Dimensional Cubes for Effective Representation of Robot in Labyrinth, Proceedings of the 36th DAAAM International Symposium, pp.0321-0327, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978- 3-902734-47-1, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/36th.daaam.proceedings.045