Generalized Coordinates for Cellular Automata Grids

Lev Naumov

Saint-Peterburg State Institute of Fine Mechanics and Optics, Computer Science
Department, 197101 Sablinskaya st. 14, Saint-Peterburg, Russia
levnaumov@mail.ru

Abstract. After some cellular automata basics here is stated an approach of universal data organization for different automata grids. It allows to identify any cell with only one non-negative integer index even for multidimensional grids

1 Introduction

Cellular automata are simple models, which are used for studying complex systems behavior in different fields of science. They found applications in physics, mathematics, computer sciences, chemistry, psychology, meteorology, social sciences and others.

These automata are discrete dynamic systems, which work can be completely described with the terms of local interactions [1, 2]. One of the most important causes of interest to cellular systems is that they form the common paradigm of parallel computations as Turing machines do for the consecutive computations [3].

2 Cellular Automata Basics

Here we will use the following theoretical formalism of cellular automata: cellular automaton A is a set of four objects $A = \langle G, Z, N, f \rangle$, where

- G – set of cells, automaton's workspace and data storage. It is named as “grid”;
- Z – set of possible cells states;
- N – description of cells neighborhood. Neighborhood is a set of cells which have an influence on the currently considered one;
- f – next-state function. It can operate so $Z^{|N|+1} \rightarrow Z$ (if cells state on the current step has an influence on its state in the next moment, so it can be said that such automaton have cells “with memory”) or so $Z^{|N|} \rightarrow Z$ (only cells neighbors are significant, then cells can be named “memoryless”) [4].

The difference between these two variants of next-state functions operation isn't important on practice. It is just theoretical. For real tasks next-state function has been generally defined as a computable program on a programming language (may be, a

specialized language for cellular automata next-state function definition) [1], instead of a mathematical (or logical) expression. It can be also named as “rules” of cellular automaton.

The set Z is to be finite. Although, for example, in physics task solving it can represent continuum, an interval of possible physical magnitude values, but on practice, any floating-point variable has a finite set of values.

Grid G is an array of cells, each of them can contain a value (or “be in state”) from a set Z . Grids can be one-dimensional, two-dimensional or multidimensional, it depends on task. However here we will view two-dimensional grids, composed of regular polygons. There are only three such grids: a grid of triangles (Fig. 1), a grid of squares (Fig. 2) and a grid of hexagons (Fig. 3).

Fig. 1. Grid of triangles and cells nearest neighbors on it

Fig. 2. Grid of squares and cells nearest neighbors on it

Fig. 3. Grid of hexagons and cells nearest neighbors on it

Cells neighborhood N is to be common for all of grids cells. So, after application of a metrics for a grid, this set can be determined as a collection of displacements relative to the currently considered cell. As usual neighborhood of a cell is a subset of its nearest cells. On Fig. 1-3 they are shown for all three considered grids. The involved cell is dark, its nearest neighbors are darken a little. The cells, that have a common edge

with the involved one can be named as a “main neighbors” (they are showed with the hatching). The set of actual neighbors of the cell a , which can be found according to N , we will denote as $N(a)$.

Cellular automata basic properties can be formulated as following:

- *Laws are local.* All cells neighbors (elements of neighborhood) are to be on a finite distance from it;
- *System is similar for all the cells.* Neighborhood and laws are always the same. There are no two cells on the grid, which can be distinguishable by the landscape;
- *All cells get their new values simultaneously, at the end of the timestep,* after all new values were calculated for all grids cells.

In previous narration we several times mentioned that there is a use in metrics for a grid of cellular automata. Let us assume the nearest neighbors of the cell to be the cells “of first ring” of the involved one. For the current cell, the cells of the i -th ring are the cells of the first ring of all cells, which are the member of $(i-1)$ -th ring, excluding cells of $(i-1)$ -th and $(i-2)$ -th rings. Formally, if $R(a, i)$ is a set of cells of i -th ring of cell a , then we can write

$$R(a, i) = \{b | \exists c : b \in R(c, 1), c \in R(a, i-1), b \notin R(a, i-1), b \notin R(a, i-2)\} \quad (1)$$

On Fig. 4-6 cells of different rings are showed with the different hatching or color.

It is useful to assume $\{a\} = R(a, 0)$. The important property of ring concept is that if a is in $R(b, i)$ then b is in $R(a, i)$.

So now we can denote a distance function [5] between cells a and b , $D(a, b)$. For example, it could be done as follows:

$$D(a, b) = i : a \in R(b, i) \quad (2)$$

The notion of ring may be generalized for multi-dimensional grids. And the definition of distance function, given by formula (2), would remain the same.

Fig. 4. First four rings for the cell of the grid of triangles

Fig. 5. First four rings for the cell of the grid of squares

Fig. 6. First four rings for the cell of the grid of hexagons

3 Generalized Coordinates for Cellular Automata Grids

When the task is to deal with the n -dimensional array of data with the help of cellular automaton, the obvious way is to use the n -dimensional grid. Each cell can be defined with the help of n -dimensional coordinate (Cartesian in most cases). Usage of generalized coordinates is the way to manage with one-dimensional array always [6].

The idea is to enumerate all cells of the grid with, for example, non-negative integer numbers, which we will name as generalized coordinates. It must be done without any blanks. Each number is to have one and only one corresponding cell. Last sentence is correct with the assumption that the grid is infinite. In fact, of course, generalized coordinates have to be bounded above. But the bound can move aside if needed and new cells can be appended to the end of the array.

The n -dimensional arrays are very suitable for quick foundation of the nearest neighbors. If it would be shown that there is still an ability to find cells nearest neighbors, after applying generalized coordinates, it would mean that they can be used for cellular automata tasks solving. In case when N contains not nearest neighbors, they would be applicable too. Then cells could be found as nearest neighbors of nearest neighbors and so on.

The ways of associating cells with generalized coordinates can be different. The main aim is to introduce them in the way, which allows to get cells neighbors as fast as possible. Lets look at two examples of such ways. First – method of the spiral coordinates introduction for the grid of hexagons. Second is applicable for the grid of triangles, but it is based on the first approach. Other methods of general coordinates introduction and their introduction for other grids remain out of this paper [6].

3.1 Spiral Generalized Coordinates by the Example of the Grid of Hexagons

The rule of enumeration of cells is very simple: chose any zero-cell and enumerate in each its ring, for example, clockwise. Possible result is shown on Fig. 7.

Fig. 7. Spiral generalized coordinates for grid of hexagons. Rings corner cells and zero-cell are darken here

Definition of a cell, which gets zero coordinate is not important. The system is invariant relative to its position.

The task is to find cells nearest neighbors now. Lets enumerate first rings cells with indexes from 0 to 5 as it is shown of Fig. 8.

Fig. 8. Cells neighbors indexes for the grid of hexagons

Let $N_i(a)$ be the i -th neighbor of the cell a (i is from 0 to 5). We may find just four of six neighbors, because of the formulae (3), but we will show how to find all of neighbors without recursion.

$$\begin{aligned} N_0(a) &= N_5(N_1(a)) = N_1(N_5(a)) \\ N_3(a) &= N_4(N_2(a)) = N_2(N_4(a)) \end{aligned} \tag{3}$$

On Fig. 7 cells that are the corners of the rings were darked. Each ring has 6 corners, which are also enumerated from 0 to 5 as shown on Fig. 7. Let $C_{i,j}$ be the coordinate of j -th corner of the i -th ring (j is from 0 to 5). It could be shown [6] that

$$C_{i,j} = 3(i^2 - i) + 1 + i \cdot j \tag{4}$$

The number of ring containing cell a , $n(a)$, can be found using formula (5)

$$n(a) = \left\lceil \frac{\sqrt{1 + 8 \lceil a/6 \rceil} - 1}{2} \right\rceil \tag{5}$$

Calculation of the neighbors of cell x comes to using formula (5). So you can determine the position of the cell inside the ring. All possible 13 positions (it could be any corner cell or in the interval between two corners) have corresponding rows in table 1. Then evaluate the expression, which is situated on the necessary row of the table, in the column, which corresponds to a definite neighbor index.

Table 1. Expressions for foundation of cells neighbors for the grid of hexagons (n – the number of ring, which contains the cell (see formulae (5)), x – involved cell)

Position	Neighbor index					
	0	1	2	3	4	5
$C_{n,0}$	$C_{n+1,0}$	$C_{n+1,0} + 1$	$x + 1$	$C_{n-1,0}$	$C_{n-1,0} - 1$	$C_{n+2,0} - 1$
$(C_{n,0}; C_{n,1})$	$C_{n+1,0} - C_{n,0} + x$	$C_{n+1,0} - C_{n,0} + x + 1$	$x + 1$	$C_{n-1,0} - C_{n,0} + x$	$C_{n-1,0} - C_{n,0} + x - 1$	$x - 1$
$C_{n,1}$	$C_{n+1,1} - 1$	$C_{n+1,1}$	$C_{n+1,1} + 1$	$x + 1$	$C_{n-1,1}$	$x - 1$

$(C_{n,1};$ $C_{n,2})$	$x-1$	$C_{n+1,1}-C_{n,1}$ $+x$	$C_{n+1,1}-C_{n,1}$ $+x+1$	$x+1$	$C_{n-1,1}-C_{n,1}$ $+x$	$C_{n-1,1}-C_{n,1}$ $+x-1$
$C_{n,2}$	$x-1$	$C_{n+1,2}-1$	$C_{n+1,2}$	$C_{n+1,2}+1$	$x+1$	$C_{n-1,2}$
$(C_{n,2};$ $C_{n,3})$	$C_{n-1,2}-C_{n,2}$ $+x-1$	$x-1$	$C_{n+1,2}-C_{n,2}$ $+x$	$C_{n+1,2}-C_{n,2}$ $+x+1$	$x+1$	$C_{n-1,2}-C_{n,2}$ $+x$
$C_{n,3}$	$C_{n-1,3}$	$x-1$	$C_{n+1,3}-1$	$C_{n+1,3}$	$C_{n+1,3}+1$	$x+1$
$(C_{n,3};$ $C_{n,4})$	$C_{n-1,3}-C_{n,3}$ $+x$	$C_{n-1,3}-C_{n,3}$ $+x-1$	$x-1$	$C_{n+1,3}-C_{n,3}$ $+x$	$C_{n+1,3}-C_{n,3}$ $+x+1$	$x+1$
$C_{n,4}$	$x+1$	$C_{n-1,4}$	$x-1$	$C_{n+1,4}-1$	$C_{n+1,4}$	$C_{n+1,4}+1$
$(C_{n,4};$ $C_{n,5})$	$x+1$	$C_{n-1,4}-C_{n,4}$ $+x$	$C_{n-1,4}-C_{n,4}$ $+x-1$	$x-1$	$C_{n+1,4}-C_{n,4}$ $+x$	$C_{n+1,4}-C_{n,4}$ $+x+1$
$C_{n,5}$	$C_{n+1,5}+1$	$x+1$	$C_{n-1,5}$	$x-1$	$C_{n+1,5}-1$	$C_{n+1,5}$
$(C_{n,5};$ $C_{n+1,0}$ $-1)$	$C_{n+1,5}-C_{n,5}$ $+x+1$	$x+1$	$C_{n-1,5}-C_{n,5}$ $+x$	$C_{n-1,5}-C_{n,5}$ $+x-1$	$x-1$	$C_{n+1,5}-C_{n,5}$ $+x$
$C_{n+1,0}$ -1	$C_{n+2,0}-1$	$C_{n,0}$	$C_{n-1,0}$	$C_{n,0}-1$	$x-1$	$C_{n+2,0}-2$

In [6] formalism of spiral generalized coordinates is applied also to grid of triangles and squares.

3.2 Example of Generalized Coordinates for Grid of Triangles Based on Spiral Generalized Coordinates for Grid of Hexagons

Another example of introduction of generalized coordinates will be made for the grid of triangles. It is based on the previous method.

Every hexagon can be presented as if it consists of the six triangles how it is shown on Fig. 9.

Fig. 9. Hexagon consists of six triangles

Fig. 10. Generalized coordinates for grid of triangles based on spiral generalized coordinates for grid of hexagons (small numbers – triangles coordinates, big numbers – hexagons coordinates)

Fig. 11. Cells neighbors enumeration for the grid of triangles

Generalized coordinates may be introduced in this way: after applying of generalized coordinates for such hexagons, we can get each triangles coordinate as a coordinate of hexagon, multiplied by 6 with the added index of triangle inside the hexagon (these indexes are presented on Fig. 9). The result is shown on Fig. 10 (note that grid

of hexagons was turned there in comparison with Fig. 7). Neighbor cells can be easily computed for this method. Lets enumerate cells neighbors as showed on Fig. 11 (there are to cells orientation variants, so we need two numerations).

In this case the algorithm of computing of neighbor cells for cell x is following: let h equals residue of division of x by 6 and H equals integer part of division of x by 6. So h would be triangle index inside the hexagon and H would store hexagons generalized coordinate. In table 2 we adduced the accordance between h and the neighbor indexes (Fig. 11). Each cell can contain one or two numbers, separated with hyphen. If there is only one number then it means that desired cell has the specified index inside the same hexagon H . If there are two number then the first is neighbor index of hexagon (Fig. 8) to which neighbor cell belongs, and second is a triangle index in the founded hexagon. So all numbers are from 0 to 5.

Table 2. Indexes for foundation of cells neighbors for the grid of triangles

H	Neighbor index											
	0	1	2	3	4	5	6	7	8	9	10	11
0	5-3	1	5	5-2	0-5	0-4	2	3	4	4-2	4-1	5-4
1	0-4	2	0	0-3	1-0	1-5	3	4	5	5-3	5-2	0-5
2	1	1-5	3	0-4	0-3	1-0	1-4	2-1	2-0	4	5	0
3	2	2-0	4	1-5	1-4	2-1	2-5	3-2	3-1	5	0	1
4	5	3	3-1	0	1	2	2-0	2-5	3-2	3-0	4-3	4-2
5	0	4	4-2	1	2	3	3-1	3-0	4-3	4-2	5-4	5-3

We used rather successful neighbors numeration (Fig. 11). The success is that there is no need in two separate tables of different cells orientations.

Tests shows [6], that this way of introducing of generalized coordinates for the grid of triangles allows to calculate the nearest neighbor cells several times faster than a spiral variant for the triangle grid [6]. It is caused by complexity of a ring for the grid of triangles and by recursion which was used for spiral coordinates.

4 Resume

Usage of generalized coordinates can be rather useful and gives several opportunities:

1. Generalized coordinates provide a universal way of data storage for different grids. The only thing need to be changed for transformation from one cellular automaton to another one – the table of expressions of the cells nearest neighbors;
2. Grid may be easily enlarged if it is necessary. Appending cells to the end of the chain is much more easier than, for example, reallocation of multidimensional block structures. It can found application for the tasks like diffusion processes studying, when data propagate in all directions;
3. Serial data is easier to serialize and store;
4. Independency from the zero-cells position gives the opportunity to move it as it would be more useful. May be – to the center of activity, to operate with the smaller

numbers. May be – far from the activity, so the majority of calculations would be on the long edges of the rings, so some neighbors could be found by addition or subtraction of unity to cells coordinates (it is said for the spiral method);

5. Generalized coordinated is just a concept, so it could be adopted for the definite task, for example, according to the internal symmetry of system.

Cellular automata are the models of parallel computations with infinite extent of parallelism. It could be achieved with the help of infinitive amount of Turing machines. Using the generalized coordinates the system with infinite extent of parallelism can be emulated with the help of finite quantity of Turing machines. A single machine used to work with data storage (the tape) and finite quantity of machines is used for neighbors calculation and synchronization. Their exact amount depends on realization (on functions programmed in machines for neighbor cells calculations).

Main disadvantage of the offered approach is that calculations of neighbors are, obviously, slower than, for example, for the Cartesian case.

For investigations of cellular automata a project CAME&L was started in Saint-Petersburg State Institute of Fine Mechanics and Optics. CAME&L is Cellular Automata Modeling Environment & Library. It is a Windows-based software package, that is desired to be simple, extensible workspace for complicated cellular calculations. Each automaton is projected to be represented by a trinity of components:

- grid – that implements visualization of grid and cells navigation;
- metrics – that provides the relationship of neighborhood and, optionally, it can allow to calculate distance between cells (may be useful for rules writing);
- data – that maintains data storage and some aspects of data visualization.

This union is to cooperate with two other components:

- rules – defines the rules of automaton and controls the iterations;
- analyzer – allows to keep an eye on definite properties of automaton, draw diagrams, draw up the reports and all of this kind.

A part of CAME&L, named CADLib (Cellular Automata Developing LIBrary) is designing to present an easy-to-use and rich set of instrument for developers and researchers. Its base is a class hierarchy for further enlarging by the users. CAME&L is planned to be provided with the set of examples of different classes and some components.

References

1. Toffoli, T., Margolus, N.: Cellular Automata Machines. Mir (World), Moscow (1991)
2. Sarkar, P.: A Brief History of Cellular Automata // ACM Computing Surveys, Vol. 32. N1. (2000)
3. von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press (1966)
4. Naumov, L., Shalyto, A.: Cellular Automata: Realization and Experiments. In preparation
5. Korn, G., Korn, T.: Mathematical Handbook for Scientists and Engineers. McGraw-Hill Book Company, London (1968)
6. Naumov, L.: Applying and Using of Generalized Coordinates in Cellular Automata with the Grid of Triangles, Squares and Hexagons. In preparation