IC/85/92

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

EP

NESTED CELLULAR AUTOMATA

Uwe Quasthoff

INTERNATIONAL ATOMIC ENERGY AGENCY

UNITED NATIONS EDUCATIONAL SCIENTIFIC AND CULTURAL ORGANIZATION

1985 MIRAMARE-TRIESTE

International Atomic Energy Agency
and
United Nations Educational Scientific and Cultural Organization

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

NESTED CELLULAR AUTOMATA *

Uwe Quasthoff **
International Centre for Theoretical Physics, Trieste, Italy.

ABSTRACT

Cellular automata by definition consist of a finite or infinite number of cells, say of unit length, with each cell having the same transition function. These cells are usually considered as the smallest elements and so the space filled with these cells becomes discrete.

Nevertheless, large pictures created by such cellular automata look very fractal. So we try to replace each cell by a couple of smaller cells, which have the same transition functions as the large ones. There are automata where this replacement does not destroy the macroscopic structure. In these cases this nesting process can be iterated.

The paper contains large classes of automata with the above properties. In the case of one dimensional automata with two states and next neighbour interaction and a nesting function of the same type a complete classification is given.

MIRAMARE - TRIESTE
July 1985

* To be submitted for publication.

** Permanent address: Karl-Marx Universität Leipzig, Sektion Mathematik, 7010 Leipzig, German Democratic Republic.

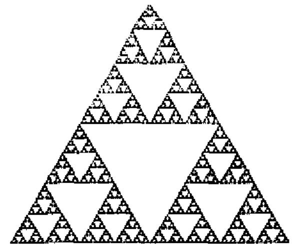

Figure 1.

the cells in a neighbourhood of the considered cell. The concrete structure of this neighbourhood has to be given by defining T. We now want to replace each unit cell by a couple of smaller cells without destroying the macroscopic structure. That means there has to be a nesting function N such that

(1) N maps the states of the smaller cells on the state of the replaced cell, and

(2) one has $T\cdot N=N\cdot T^k$ for some $k\geq 2$.

The precise definition is given in section 2.

The existence of such a nesting function has several consequences. The process described above can be iterated by considering the smaller cells as new unit cells and so on. Then property (2) tells that the time unit for a very small cell constructed in the n^{th} step is just the k^n-th part of the time unit of the original unit cell. So we can approximate continuous processes (continuous both in space and time) by such nested cellular automata.

In section 2 we describe the outlined construction in detail.

Section 3 gives a full list of possible transition functions T of two state automata and next neighbour interaction (both one- and twosided) which allow a nesting process of the same kind.

In section 4 examples of nesting processes with larger neighbourhoods and higher growth rates (defined in section 4) are presented. The example given in the beginning of the introduction is generalized. All examples allow generalization to arbitrary high dimension.

The last section deals with a more complicated neighbourhood structure. We show that there are nesting processes in the case of neighbourhoods described by Meshalkin [2].

2. Nesting functions

Throughout the paper we consider cellular automata with the cells placed in a d dimensional lattice Λ isomorphic to $\mathbb{Z}^d$. Every cell contains an automaton with a given finite state space Y and a transition function T (independent of the special cell). This transition function has as arguments the states of the cells of a finite neighbourhood at time t and returns the state of the cell at time t+1. The structure of these neighbourhoods is up to section 5 always assumed to be translation invariant in the following sense. If $U(x)=(y_1,y_2,\dots,y_m)$ is the neighbourhood of x then $U(x+x')=(y_1+x',y_2+x',\dots,y_m+x')$. In the nesting process we assume that each cell of the lattice is (simultaneously) replaced by a couple of cells so that the new lattice is

$$\Lambda'=\{(x_1+k_1/n_1,\dots,x_d+k_d/n_d),\ x\in\Lambda,\ n_i\geq 1,\quad k_i=0,1,\dots,n_i-1\}.$$

Call this nesting process of **order** $(n_1,\dots,n_d)$ or of order n_1 if d=1. This nesting process can be iterated by identifying the point $x'=(x'_1,\dots,x'_d)\in\Lambda'$ with $x=(n_1x'_1,\dots,n_dx'_d)\in\Lambda$. This mapping $\Phi:\Lambda\to\Lambda'$ defines neighbourhoods in Λ' by $U(x')=\Phi U(x)$.

A **nesting function** is a function $N:Y^{n_1\cdot\ldots\cdot n_d}\to Y$. By reversing this nesting process N assigns a state to a larger cell using the states of the smaller cells inside (some ordering of these smaller cells is always assumed).

<u>Definition 1.</u> A pair (T,N) of a transition function and a nesting function is called **compatible** if there is a natural number k such that

$$T\cdot N = N\cdot T^k.$$

Such a pair is called nontrivial if $T\neq id$, $N\neq 0$ and $k\geq 2$. The constant k is called growth **rate**.

3. Complete analysis of isomorphic type 2 and type 3 automata with two states and next neighbour interaction in one dimension

<u>Definition 2.</u> A nesting process and a compatible pair (N,T) are both called of **isomorphic type** $(n_1,\dots,n_d)$ (or of isomorphic type n_1 if d=1) if the neibourhood U corresponding to T is a rectangular subset with the sides $n_1,\dots,n_d$.

This means that the isomorphism Φ maps for every $x\in\Lambda$ the domain of T onto the domain of N.

<u>Definition 3.</u> A transition function T is called having next neighbour interaction of type 2 or 3 if $U(x)=(x,x+1)$ or $U(x)=(x-1,x,x+1)$, respectively.

If $x\in\Lambda$, then s_x shall denote the state of the automaton at the point x. s (without index) shall denote the (ordered tuple of) states of automata in a subset of Λ clear from the context.

To give names to the transition functions of this type we use a method described in [5], for instance. Read the states (0's and 1's) in a neighbourhood (in their natural order) as a binary number b(s) (with s–the couple of states in U) and assign to a transition function T the name $\Sigma T(s)\cdot 2^{b(s)}$ where the summation is taken over all possible 2^n couples of states in the type n case (n=2,3).

For instance in the type 2 and type 3 case T=identity has the names 12 and 204, respectively. T=left shift has the names 10 and 170, respectively.

The nesting functions are numbered in the same way. We want the global state, where all automata are in state 0 to be stable ("where nothing is, nothing can happen"), that means we only consider T's and N's with even names. We do not restrict ourself to symmetric functions, as, for instance, [1], because we can combine asymmetric functions in a way described in section 5. We have the following result.

<u>Theorem 1.</u> In the type 2 case, the only compatible pairs are (6,6), (6,10), (6,12), (8,8), (10,n), (12,n) and (14,14) with n=2,4,...,14.

<u>Proof:</u> It is easily checked by hand that the above pairs are compatible. The proof that these are all is done by computer. ■

<u>Theorem 2.</u> In the type 3 case, all compatible pairs are (128,128), (136,128), (170,n), (192,128), (204,n), (238,254), (240,n), (252,254) and (254,254), where n=2,4,...,254.

<u>Proof:</u> By computer. ■

<u>Remarks:</u>

1. Both in Theorems 1 and 2 a transition function is compatible with all nesting functions iff it is the identity or a shift.

2. The nesting functions 8 and 14 in the type 2 case and 128 and 254 in the type 3 case seem to be of little interest because the binomial expension of these numbers have all but one digit equal.

3. So the only remaining interesting compatible pairs have the transition function 6 which can be described as addition modulo 2. This is just the example given in the introduction. It is also the only transformation leaving the global state with all cells in state 1 not invariant.

<u>4. Different groth rates and higher dimensions</u>

First we want to show that for arbitrary groth rate $k\geq 2$ there are compatible pairs in dimension 1.

<u>Theorem 3.</u> If T is the left shift, $k\geq 2$ is given and N is any nesting function of order k, then $T\cdot N=N\cdot T^k$, i.e. T and N are compatible with groth rate k. The analogous statement holds for the right shift.

<u>Proof:</u> First note that T shifts one unit to the left and that these units differ in both sides of the equation $T\cdot N=N\cdot T^k$. Without loss of generality we can assume $U(x)=\{x, x+1/k, \dots, x+(k-1)/k\}$. Then we have

$$(T\cdot N)(s)_x = N(s)_{Tx} = N(s)_{x-1} = N(s_{x-1}, s_{x-1+1/k}, \dots, s_{x-1/k}).$$

On the other hand

$$(N\cdot T^k)(s)_x = N(T^k s)_x = N(s_{x-1}, s_{x-1+1/k}, \dots, s_{x-1/k}).$$

■

<u>Remark.</u> The same result can easily be proved for higher dimensions.

Next we want to generalize the example of the introduction to arbitrary high growth rates.

<u>Definition 3.</u> For every natural number n define T_n in the following way. Take $\Lambda=\mathbb{Z}$ and for $x\in\Lambda$ let $U(x)=\{x-[(n-1)/2], x-[(n-1)/2]+1, \dots, x+[n/2]\}$ where [y] denotes the integer part of y and define

$$(Ts)_x = \sum_{y \in U(x)} s_y \bmod n.$$

Define a nesting function N_n of type n also by summing up its arguments modulo n.

For $n=2$, the pair (T_2, N_2) is compatible by Theorem 1. We have the following more general result.

<u>Theorem 4.</u> For every prime p, we have the compatible pair (T_p, N_p) satisfying $T_p \cdot N_p = N_p \cdot T_p^p$.

<u>Proof:</u> *For simplicity of notation we shift the neighbourhoods* $[(p-1)/2]$ units and take $U(x)=(x, x+1, \ldots, x+p-1)$.

First we have

$$\begin{aligned}(T_pN_ps)_x &= T_p((N_ps)_x, (N_ps)_{x+p}, \ldots, (N_ps)_{x+p(p-1)}) \\ &= T_p(s_x+\ldots+s_{x+p-1} \bmod p, \ldots, s_{x+p(p-1)}, \ldots, s_{x+p^2-1} \bmod p) \\ &= \sum_{k=0}^{p^2-1} s_{x+k} \bmod p.\end{aligned}$$

Next we define constants $a_k^{(l)} \in \{0, 1, \ldots, p-1\}$ by

$$(T_p^l s)_x = \sum_{k=0}^{l(p-1)} a_k^{(l)} s_{x+k} \bmod p$$

and $a_k^{(l)}=0$ for all $k>l(p-1)$.

This implies

$$\begin{aligned}(N_p \cdot T_p^p s)_x &= N_p((T_p^p s)_x, (T_p^p s)_{x+1}, \ldots, (T_p^p s)_{x+p-1}) \\ &= \sum_{k=0}^{p^2-1} (a_k^{(p)}+a_{k-1}^{(p)}+\ldots+a_{k-p-1}^{(p)}) s_k \bmod p \\ &= \sum_{k=0}^{p^2-1} a_k^{(p+1)} s_k \bmod p.\end{aligned}$$

So it sufficies to prove $a_k^{(p+1)}=1 \bmod p$. Note that we have $p_0^{(j)}=1$ for all j. Therefore $a_k^{(p+1)}=1 \bmod p$ is equivalent to

$$a_0^{(p)}=1, \quad a_k^{(p)}=0 \bmod p \text{ for } k=1, \ldots, p-1.$$

This in turn is equivalent to

$$a_0^{(p-1)}=1, \quad a_1^{(p-1)}=-1, \quad a_k^{(p-1)}=0 \bmod p \text{ for } k=2, \ldots, p-1.$$

By induction one has

$$a_k^{(j)} = (-1)^k \cdot \binom{p-j}{k}.$$

But for any prime number p we have $p \mid \binom{p}{k}$ for $0<k<p$, i.e.

$$a_k^{(0)} = (-1)^k \cdot \binom{p}{k} = 0 \bmod p \text{ for } 0<k<p.$$

So $a_k^{(p+1)}=1 \bmod p$ is equivalent to $a_0^{(0)}=1$, $a_k^{(0)}=0 \bmod p$ for $1<k<p$. The last condition is fulfilled by construction. This completes the proof. ■

<u>Remarks:</u>

1. Theorem 4 can be generalized to higher dimensions in the following way. Let T_p, U, and N_p be as in Theorem 4. To define the analogous automata in higher dimensions, take $\Lambda'=\mathbb{Z}^d$, $U'=U\times\ldots\times U$ (d factors) and T'_p and N'_p as before the functions summing up its arguments modulo p. Then for any prime p, the pair (T'_p, N'_p) is compatible. The proof is similar to that of Theorem 4, using one more induction argument.

2. The fact that the only interesting function of isomorphic type 2 (see Remark 3 after Theorem 2) can be generalized to arbitrary large neighbourhoods, many state automata, high growth rates and high dimensions sheds another light at the result of Theorem 2. There seems to be two classes of pairs (n,U) of n-state automata with neighbourhood U: In the first class we have interesting compatible pairs in the above sense, in the other class not. The compatible pairs (T_p, N_p) of Theorem 4 belong to the first class, all pairs of Theorem 2 belong to the second class. This phenomenon needs further classification.

5. Meshalkin type examples

First we want to recall the original neighbourhood structure given by Meshalkin [2], see also [3,4].

Let $\Lambda = \mathbb{Z}$ and consider three state automata with state space $Y=\{1,a,b\}$. Define a transition function $T: Y \times Y \to Y$ by

$$T(x,x') = x' \quad \text{for } x,x' \in Y$$

(i.e. T projects on the second argument). This simple transition function yields to difficult behaviour with the following neighbourhood structure. We assume that the states 1, a, and b appear in Λ with densities 1/2, 1/4 and 1/4, respectively. Next we construct pairs (a,1) and (b,1) in the following way. If a state a or b appears as left neighbour of an 1, then they build a pair. Next cut out all cells which are part of pairs already built and continue the same procedure. With probability one all cells can be put into pairs. Figure 2a gives an example with the pairs indicated. If s is the (global) state of Λ, then the neighbourhood of an $x \in \Lambda$ is $U(x)=(x,x')$ where x' is the other cell of the pair containing s_x. Note that we use the ordering (x,x') for the arguments of T. Figure 2b gives the transformed of Figure 2a.

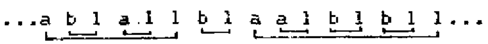

Fig. 2a

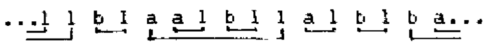

Fig. 2b

This neighbourhood structure is no more translation invariant in the sense described in section 2 because $U(x)$ depends on the state s. But we have a weaker translation invariance: Let S be

the (left) shift. Then for any state s we have $SU(x)=U(Sx)$. The structure of the pairs is invariant under the action of T^2.

Define another neighbourhood $U'(x)=T^{-1}U(Tx)$. Then U and U' assign to any cell a left and right neighbour. So we get (finitely or infinitely many) sequences of neighboured cells. Each sequence has the form

$$\ldots, c_{-1}, 1, c_0, 1, c_1, 1, \ldots \quad \text{with } c_k \in \{a,b\}.$$

Next note that T acts as right shift on each of these sequences and T^2 acts as right shift on the sequences $\ldots, c_{-1}, c_0, c_1, \ldots$

Corresponding to this neighbourhood structure we define the following nesting procedure. N cuts out the subsequence $c=(\ldots, c_{-1}, 1, c_0, 1, c_1, 1, \ldots)$ containing the cell at $x=0$. To give N a form similar to that of section 2 we can built pairs (x_k, y_k) *with* $x_k \in c$, $y_k \in c$ *and define* $N: Y \times Y \to Y$ by $N(x_k, y_k)=y_k$ *taking its value* at the cell of the second argument.

<u>Theorem 5.</u> The pair (T^2,N) is compatible, we have $T^2 \cdot N = N \cdot T^2$.

<u>Proof:</u> The Theorem follows from the fact that T^2 leaves the structure of the neighbourhoods invariant. So in both cases N cuts out the same pairs (on different places). ■

In our last example we use a more complicated state space $Y=\{1,a,b,R,A,B\}$ with 6 states and densities (1/4, 1/8, 1/8, 1/4, 1/8, 1/8). Let the neighbourhood relation of the 1's, a's and b's be as before. For the R's, A's and B's we do the same but take pairs with A, B as right (instead of left) neighbours of R. Then in the same way as before we get subsequences

$$\ldots c_{-1}, 1, c_0, 1, c_1, 1, \ldots \quad c_k \in \{a,b\}$$

$$\ldots C_{-1}, R, C_0, R, C_1, R, \ldots \quad C_k \in \{A,B\}$$

While T acs as right shift on the first kind of sequences, it acts as left shift on the second. So there is no reason to exclude "asymmetric " functions as shifts from our discussion because they can be put together to build a function with no distinguished direction.

ACKNOWLEDGMENTS

The author would like to thank Professor Abdus Salam, the International Atomic Energy Agency and UNESCO for hospitality at the International Centre for Theoretical Physics, Trieste.

References:

[1] Ingerson, T.E.;Buvel, R.L.:Structure in asynchronous cellular automata, Physica 10D (1984), 91-99

[2] Meshalkin, L.D.: A case of isomorphism of Bernoulli schemes, Dokl. Akad. Nauk SSSR 128 (1954), 41-44 (in Russian)

[3] Quasthoff, U.: to appear in Proceedings of the Workshop "Fractals in Physics", Trieste 1985

[4] Weiss, B.: The isomorphism problem in Ergodic Theory, Bull. AMS 78 (1972), 668-694

[5] Wilson, S.J.: Cellular Automata can generate fractals, Discrete Appl. Math.8 (1984), 91-99

[6] Wolfram, S.: Statistical Mechanics of cellular automata, Rev. Mod. Phys. 55(1983), 601-644