E11-2002-268

V. M. Severyanov

AN APPROACH TO FORMAL DESCRIPTION OF ONE CLASS OF AUTOMATA NETWORKS

1 Introduction

Automata Networks were originally introduced in [1, 2, 3, 4] as models for physical and biological phenomena. Automata networks are discrete dynamical systems [5]. The state space of an automata network is defined by a graph, where each vertex takes state in a finite set and arcs represent vertex dependencies. The state of a vertex is changed according to a transition rule which takes into account only the state of its neighbors. The global dynamics is determined by a strategy of the local transition rules application.

Two important classes of Automata Networks are Cellular Automata and Neural Networks.

Hyperbolic Cellular Automata (HCA) [12, 13] are Automata Networks based on Iterated Function Systems [6] and originally were designed for constructing fractal objects [7, 8]. However, they are in their own right and can be studied, for example, in the context of the computer simulation theory [9]. Hyperbolic Cellular Automata can be considered as a generalization of Cellular Automata: the main difference lies in the fact that a hyperbolic cellular automaton has non-regular structure of the cell neighborhood system. The *distinguishing feature* of a Hyperbolic Cellular Automata is that *its neighborhood system is determined by a Hyperbolic Iterated Function System.*

A *Hyperbolic Iterated Function System* (IFS) specifies a discrete dissipative dynamical system. An IFS consists of a set of contractive functions on a complete metric space which induces a more complex contractive function F acting on the set of compact subsets of the metric space. Due to the contractivity of F there is a unique attractor A_F that satisfies $A_F = F(A_F)$. Furthermore, any set A will be eventually mapped onto the attractor A_F under repeated application of the function F. Usually contractive affine transformations are used for the specification of an IFS.

Evolving Algebras (known also as Abstract State Machines) have been proposed by Yuri Gurevich [10, 11] as the models for arbitrary computational processes. They are finite many-sorted dynamic algebras representing state transitions and describing operational semantics of discrete dynamical systems. They may be tailored to any desired level of abstraction. System states are represented here as static algebras, the dynamics is described by a set of transition rules. Evolving Algebras provide a formal method for *executable* specifications.

Several approaches are possible to using Evolving Algebras for specification of Hyperbolic Cellular Automata. One of them is outlined in this talk as the first step on the way of applying Evolving Algebra formalism to Hyperbolic Cellular Automata.

2 Iterated Function Systems

Aa *Iterated Function System* is a structure

$$F = ((X, d), f_1, f_2, ..., f_N), \tag{1}$$

where (X, d) is a *complete* metric space, with metric d, $f_i : X \to X$, a contiguous function $\forall i$. An IFS F induces an *operator* $F : \mathcal{H}(X) \to \mathcal{H}(X)$, where $\mathcal{H}(X)$, the space of all compact subsets of X

$$F = \bigcup_{i=1}^{q} f_i. \tag{2}$$

The metric space $(\mathcal{H}(X), h)$, with *Hausdorff metric* h, is also complete. When the functions f_i are *contractions*, the IFS is *hyperbolic*. In this case, there is a compact subset A_F of X which is the fixed point of the operator F

$$\exists A_F \in \mathcal{H}(X) : \; A_F = F(A_F). \tag{3}$$

Such a set is called the *attractor* of the IFS F. The pair $(\mathcal{H}(X), F)$ is a *set* dynamical system whose attractive point is a set (rather than a point) - the attractor A_F. Realizing the dynamics of such a dynamical system, we can build a fractal set.
The Hausdorff metric is defined as follows:

$$h(A, B) = \max(d_s(A, B), d_s(B, A)), \qquad A, B \in \mathcal{H}(X), \tag{4}$$

where

$$d_s(A, B) = \max_{x \in A} \min_{y \in B}(d(x, y)).$$

We are dealing here with hyperbolic IFSs with all f_i *affine* transformations.

2.1 IFS Approximation of Sets

We can conveniently work in the unit square $S = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ $S = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ for the space X. The IFS under consideration is appropriately scaled to fit into S.

In practice, we are dealing with an *approximation* of the attractor of an IFS rather than with the attractor itself. We work not in the space X but in its pixel representation $\tilde{X}$ and, therefore, we deal not with $(\mathcal{H}(X), h)$ but with $(\mathcal{H}(\tilde{X}), h)$ and, correspondingly, with $\tilde{F} = (\tilde{f}_1, \ldots, \tilde{f}_q)$ and $\tilde{S}$. In what follows under S, F and f we mean $\tilde{S}$, $\tilde{F}$ and $\tilde{f}$, respectively.

One way to build a set, specified by IFS, looks as follows. We take an initial $A_0 \in \mathcal{H}(X)$ and define

$$A_{n+1} = F(A_n) \equiv \bigcup_{i=1}^{q} f_i(A_n), \qquad n = 0, \ldots, \infty. \tag{5}$$

The sequence $\{A_n\}_{n=0}^{\infty}$ converges to A_F in the Hausdorff metric. We do N iterations of F. When the number N of iterations becomes great enough, we have an approximation

$$F^N(A_0) \approx A_F. \tag{6}$$

3 Hyperbolic Cellular Automata

We define a *Hyperbolic Cellular Automaton* (HCA) σ as a structure

$$(C, A, S_0, N, \Delta), \tag{7}$$

where

- C is a set of *cells*;
- A is an *alphabet of states* (attributes);
- $S : C \to A$ is a *state*, S_0 an *initial* state;
- $N : C \to \mathcal{P}(C)$ is a *neighborhood system*; $\mathcal{P}(C)$ is the power-set of C, $\forall c \in C :\ N(c)$ is the *neighborhood* of c;
- $\Delta : \Sigma \to \Sigma$ is a *global dynamic rule* (GDR), $\Sigma = \{S \mid S \text{ is a state}\}$.

$$\Delta^n : S_{n-1} \longmapsto S_n \qquad n = 1, 2, \ldots, M \tag{8}$$

The global dynamic rule Δ comes about from the *local dynamic rules* (LDRs)

$$\delta_c : \Sigma_{N(c)} \to A, \tag{9}$$

where $S_{N(c)}$ is the restriction of S to $N(c)$. Every LDR δ_c gives a new state value to the cell c as a function of cell states from the neighborhood $N(c)$ of c. The action of Δ

$$\Delta(S) = \bigcup_{c \in C} S[S(c) \leftarrow \delta_c(S_{N(c)})], \tag{10}$$

is the union of the states obtained by replacing the state of each cell c accordingly to δ_c.

Hyperbolic Cellular Automata can be considered as a generalization of 'Cellular Automata', the main difference is our Hyperbolic Cellular Automata have a non-regular structure of the system of the cell neighborhoods. There is no common template of vicinity for cells, the neighborhoods of cells have variable cardinality and, hence, the LDRs have a variable arity. There are possible cases when $c \notin N(c)$ and even $N(c) = \{\emptyset\}$.

A hyperbolic cellular automaton σ works as follows. If an initial state S_0 is given, it iteratively applies the GDR Δ until it reaches a steady attractive state S_A:

$$S_0 \xrightarrow{\Delta} S_1 \xrightarrow{\Delta} S_2 \xrightarrow{\Delta} \ldots \xrightarrow{\Delta} S_A.$$

The cells work synchronously in parallel governed by some global synchronization signals.

4 From Iterated Function Systems to Hyperbolic Cellular Automata

When we want to realize the dynamics of a *digitized* hyperbolic IFS

$$F = ((X, d), f_1, f_2, ..., f_k)$$

with the help of a Hyperbolic Cellular Automaton σ, we have the following picture. As C, we take a regular lattice of some dimensionality m. In a practically interesting case, $m = 2$ and $C = \{(i, j) \mid i = 1, \ldots, p,\ j = 1, \ldots, q\}$. The alphabet of states $A = \{0, 1\}$. The cell neighborhoods are defined by the inverses of the IFS's functions

$$\forall c \in C: \quad N(c) = \bigcup_{i=1}^{k} f_i^{-1}(c). \tag{11}$$

The LDRs δ_c are the same for all cells c

$$\forall c \in C: \quad \delta_c(S_{N(c)}) = \begin{cases} 1 & \text{if } \exists c' \in N(c) \ \ S(c') = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We take as C the unit square S^u of pixels and obtain the neighborhoods $N(c)$ with the help of our Algorithm described in [7].
The whole process of going from an *initial* IFS

$$F = ((X, d), f_1, f_2, ..., f_k) \qquad \& \qquad F = \bigcup_{i=1}^{k} f_i$$

to the corresponding Hyperbolic Cellular Automaton σ looks like the following. Fist of all, we go to the dynamical system

$$((\mathcal{H}(X), h), F),$$

then, after digitization and scaling, to the dynamical system

$$((\mathcal{H}(S^u), h), F),$$

where S^u is the unit square.
Eventually we construct the Hyperbolic Cellular Automaton

$$(S^u, \{0,1\}, N, A_0, \Delta) \qquad \& \qquad N = \bigcup_{s\in S^u} \bigcup_{f\in F} f^{-1}(s).$$

We start the dynamical process, beginning from a compact set $A_0 \subseteq S^u$, and after a big enough number M of iterations get an approximation A_M of the IFS's attractor A_F:

$$\Delta^* : A_0 \longrightarrow A_M \approx A_F.$$

5 Evolving Algebras

Evolving Algebras (known also as Abstract State Machines) have been proposed by Yuri Gurevich in 1988 as the models for arbitrary computational processes. They are finite dynamic algebras representing state transitions and describing operational semantics of discrete dynamical systems. They may be tailored to any desired level of abstraction. System states are represented here as static algebras. Evolving Algebras provide a formal method for *executable* specifications.

An evolving algebra is a structure

$$\psi = (\Sigma, S, I_0, T),$$

where Σ is a signature (a set of operational (or functional) symbols with their arities); S is a superuniverse (a union of all sorts), Boolean universe $\mathbf{2} \equiv \{0,1\} \subset S$ (a universe (a sort) is represented by its indicator function $U : S \rightarrow \mathbf{2}$); I_0 an initial interpretation (a finite static many-sorted algebra); T a set of transition rules.

I_0 gives an initial interpretation of the signature's operational symbols:

$$I_0 : \Sigma \rightarrow \bigcup_{n\geq 0} (S^n \rightarrow S), \qquad I_0(f) : S^{\mathrm{ord}(f)} \rightarrow S, \quad \forall f \in \Sigma.$$

NB: The interpretation $I_0(f)$ is a partial function on S^n but it is total on the corresponding universes.

There are four kinds of transition rules (or updates):

1. **Function updates**

$$f(t_1, \ldots, t_n) := t_0 \qquad f \in \Sigma, \; n \geq 0,$$

where t_i are terms, the new function

$$f' \equiv f\,[f(t_1, \ldots, t_n) \leftarrow t_0]$$

2. **Conditional**

$$if\ b\ then\ C$$

3. **Extension of universes**

$$extend\ U \subset S\ by\ x_1, \ldots, x_m$$

4. **Contraction of universes**

$$discard\ t\ from\ U$$

Iterative application of evolving algebra ψ to sequentially arising states (static algebras) I_i, starting from the initial state I_0, may give a terminated computation

$$I_0 \xrightarrow{\psi} I_1 \xrightarrow{\psi} I_2 \xrightarrow{\psi} \ldots \xrightarrow{\psi} I_M.$$

6 Hyperbolic Cellular Automata as Evolving Algebras

It is pertinent to note that there may be several approaches to using Evolving Algebras for specification of Hyperbolic Cellular Automata. In the simplest case of *one* Hyperbolic Cellular Automaton one can directly go from its mathematical definition and a given IFS to an Evolving Algebra. When it is required *to construct* an IFS and the corresponding Hyperbolic Cellular Automaton, some complications arise associated with the need to appeal to some second order constructions.

As the first step on the way of applying Evolving Algebra formalism to Hyperbolic Cellular Automata we give here a sketch of an Evolving Algebra for a Hyperbolic Cellular Automaton. Instead of speaking in terms of superuniverse it is convenient to use specific sorts (i.e. their indicator functions), and to extend the LDRs δ_c onto the whole set of cells

$$\tilde{\delta}_c(c) = \text{undef}, \quad \forall c \notin N(c).$$

The information of the IFS under consideration is reflected in the set of neighborhoods. The initial interpretation is given for the 2D-case.

- **Sorts**
  - Cells
  - Attributes
  - {N_c | $c \in$ *Cells*}   (set of neighborhoods)
  - undef
- **Static Functions**
  - {*delta_c* : Cells $\longrightarrow$ Attributes | $c \in$ Cells & *delta_c*(c) = undef, $\forall c \notin$ N_c}

- **Dynamic Functions**
  - *State* : Cells $\longrightarrow$ Attributes
- **Transition Rules**
  - $\{State(c) = delta\_c(c) \mid c \in \text{Cells}\}$
- **Initial Interpretation**
  - Cells $\longrightarrow \{(i,j) \mid i = 1,\ldots,p;\ j = 1,\ldots,q\}$
  - Attributes $\longrightarrow \{0,1\}$
  - $\left\{\text{N\_c} \rightarrow \{0,1\}^{\{(i,j) \mid i=1,\ldots,p;\ j=1,\ldots,q\}} \mid c \in \text{Cells}\right\}$
  - $\{delta\_c : \{(i,j) \mid i = 1,\ldots,p;\ j = 1,\ldots,q\} \rightarrow \{0,1\}\}$
  - $\{State : \{(i,j) \mid i = 1,\ldots,p;\ j = 1,\ldots,q\} \rightarrow \{0,1\}\}$

7 Conclusion

Hyperbolic Cellular Automata were designed for constructing fractal objects but they are in their own right. Evolving Algebra specifications are mathematically well founded and directly executable. In the talk, an Evolving Algebra approach to formal description of Hyperbolic Cellular Automata has been outlined. At least two steps of further study should be mentioned. The first one is to develop a way for specification of variable Hyperbolic Cellular Automata. The second one is to give an Evolving Algebra specification of our Algorithm of constructing a Hyperbolic Cellular Automaton for a given Iterated Function System [7].

References

[1] McCulloch W., Pitts W. - A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bull. Math. Biophysics*, 5, 1943, 115-133.

[2] Ulam S. - On Some Mathematical Problems Connected with Patterns of Growth of Figures, in *Essays on Cellular Automata*, A. W. Burks (ed), Univ. of Illinois Press, 1970, 219-243.

[3] Von Neumann J. - Theory of Self-Reproducing Automata, A. W. Burks (ed), Univ. of Illinois Press, Chicago, 1966.

[4] Von Neumann J. - The General and Logical Theory of Automata, in *Hixon Symposium Proc.*, 1948 in *J.N. Neumann Collected Works*, A.H. Taub (ed), Pergamon Press, New York, Vol. 5, 288-328, 1963.

[5] Goles E., Martinez S. - Neural and Automata Network: Dynamical Behavior and Applications, Kluwer Academic Publishers, Dordrecht, 1990.

[6] Barnsley, M. F. - Fractals Everywhere. Springer-Verlag, Berlin, 1986

[7] Severyanov, V. M. - Automata Network Dynamical Systems for Construction of Fractal Objects, *Mathematics and Computers in Simulation* 57 (2001) 317-324.

[8] Severyanov, V. M. - Parallel Computation of Fractal Sets with the Help of Neural Networks and Cellular Automata, In *Parallel Computing Technologies, Proc. IV Int. Conference, PACT-97*, Yaroslavl, Russia, September 1997, Victor Malyshkin (ed), Springer-Verlag, Berlin, 1997, 109

[9] Barret, C. L., Reidys, C. N. - Elements of a theory of computer simulation I: Sequential CA over random graphs, *Applied Mathematics and Computation* 98 (1999) 241-259.

[10] Yuri Gurevich - Logic and challenge of computer science. In *Specification and Validation Methods*,E. Börger (ed), Oxford University Press, 1995

[11] Yuri Gurevich - Evolving Algebras: a tutorial introduction, *Bulletin of the European Association for Theoretic Computer Science*, 43:264-284, 1991

[12] Severyanov, V. M - An Evolving Algebra Approach to Formal Description of one class of Automata Networks, in *VIII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT'2002), Book of abstracts*, Moscow, 2002, 83.

[13] Severyanov, V. M - Deterministic hyperbolic cellular automata as evolving algebras, in *5 International Congress on Mathematical Modeling, Book of abstracts*, Vol. 1, Moscow, 2002, 173.

Received on November 28, 2002.

Северьянов В. М. E11-2002-268
Подход к формальному описанию одного класса автоматных сетей

Гиперболические клеточные автоматы (специальный класс автоматных сетей) основаны на системах итерируемых функций и первоначально были разработаны для построения фрактальных объектов. Однако они имеют самостоятельное значение и могут изучаться, например, в контексте теории компьютерного моделирования. Гиперболические клеточные автоматы могут рассматриваться как обобщение клеточных автоматов. Основное отличие заключается в том, что гиперболический клеточный автомат имеет нерегулярную структуру системы окрестностей клеток. Эволюционирующие алгебры (известные также как абстрактные машины состояний) были предложены Ю. Гуревичем в качестве моделей для произвольных вычислительных процессов. Это конечные многосортные динамические алгебры, представляющие переходы состояний и описывающие операционную семантику дискретных динамических систем. Системные состояния представлены статическими алгебрами, динамика описана набором правил перехода. Эволюционирующие алгебры предоставляют формальный метод для исполнимых спецификаций. Представлен подход к формальному описанию гиперболических клеточных автоматов посредством эволюционирующих алгебр.

Работа выполнена в Лаборатории информационных технологий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 2002

Severyanov V. M. E11-2002-268
An Approach to Formal Description of One Class of Automata Networks

Hyperbolic Cellular Automata, a special class of Automata Networks, are based on Iterated Function Systems and originally were designed for constructing fractal objects. However, they are in their own right and can be studied, for example, in the context of the computer simulation theory. Hyperbolic Cellular Automata can be considered as a generalization of Cellular Automata. The main difference lies in the fact that a hyperbolic cellular automaton has a nonregular structure of the cell neighborhood system. Evolving Algebras (known also as Abstract State Machines) have been proposed by Yuri Gurevich as the models for arbitrary computational processes. They are finite many-sorted dynamic algebras representing state transitions and describing operational semantics of discrete dynamical systems. System states are represented here as static algebras, the dynamics is described by a set of transition rules. Evolving Algebras provide a formal method for executable specifications. An evolving algebra approach to formal description of Hyperbolic Cellular Automata is presented.

The investigation has been performed at the Laboratory of Information Technologies, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 2002

Макет *Т. Е. Попеко*

Подписано в печать 24.12.2002.
Формат 60 × 90/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 0,68. Уч.-изд. л. 1,07. Тираж 310 экз. Заказ № 53681.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.
E-mail: publish@pds.jinr.ru
www.jinr.ru/publish/