# Discrete Dynamics of Cellular Machines: Specification and Interpretation

Stefano Nichele
Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Selandsvei 7-9 7491 Trondheim Norway
nichele@idi.ntnu.no

## ABSTRACT
This paper presents research on discrete dynamics of cellular machines, their specification and interpretation. It gives an overview of the fundamental issues related to the classification of Cellular Automata (CA) classes. In particular, the possible locations of various CA capable to achieve different degrees of complex behaviors are described. This work is mainly focused on the correlation between CA behavior and cellular regulative properties. A possible minimalistic experimental setup is presented, together with some preliminary results and ideas that can be investigated in future work.

## Categories and Subject Descriptors
I.2.m [**Artificial Intelligence**]: [Miscellaneous]

## General Terms
Design, Experimentation

## Keywords
Development, Cellular Computation, Emergence

## 1. INTRODUCTION
For the past 50 years computers have been based on the so called von Neumann architecture, where one complex processor sequentially performs a single task at each time-step. Recently, new computational paradigms have been explored and investigated. These new systems are based on a myriad of small and unreliable components called cells. Even if a single cell itself can do very little, the emergent behavior of the system as a whole is capable of complex dynamics. In cellular computing [12] each cell can only communicate with a few other cells, most or all of which are physically close by (neighbors). One implication of this principle is that none of the cells has a global view of the entire system, i.e., there is no central controller. Such systems can be modeled using specific computational machines called cellular automata. The metaphor with biology can be exploited on cellular systems because the physical structure is similar to biological multi-cellular organisms.



The main initial topic of this research is to investigate the complexity of cellular machines and their behavior, in relation with the development process. This work includes identification of favorable developmental properties, e.g. self–replication, self-adaptation, self-repair, toward machines capable of complex computation.

Section 2 gives motivations and background information. In Section 3 the main objectives are presented and in Section 4 Wolfram's CA classes are described. In Section 5, Langton's work on the computation at the Edge of Chaos is introduced. Section 6 discusses the connection between previous works and ongoing research. A possible minimalistic experimental setup for the investigation of correlation between CA behavior and cellular regulative properties is introduced in Section 7. Section 8 shows some preliminary results and finally, Section 9 concludes the paper.

## 2. MOTIVATION AND BACKGROUND
The work is going to be included in current projects of using biological inspiration from evolution and development towards hardware capable of unconventional computation. The approach in this project is to use a combination of artificial development and evolutionary algorithms (EvoDevo) [5]. In this work, the computational architectures targeted are architectures that can be viewed as sparsely connected networks, e.g. boolean networks or cellular automata. The computational output of such architectures can be seen as a discrete dynamic behaviour.

Even though such architectures have vast computational power, due to the massive parallel operation, the specification of input data and the interpretation of the dynamics of the system (the output) are not trivial. This project aims to gain knowledge of how specification of input data and the interpretation of the dynamics can be improved as to be able to exploit these architectures in real world problems. This involves work in the field of artificial development, unconventional computation and possible utilization of existing hardware technology or computational models of cellular machines.

Artificial developmental systems are analyzed and evaluated by viewing the system as a discrete dynamic system and the development process is treated as series of discrete events, each representing a point in time on the developmental path from zygote to multi-cellular organism. If the parallel nature and limited local communication of a cellular system is considered in relation with the discrete time update of the system, a developmental system (e.g. a CA), can be approached as a network of sparsely connected units (cells). Such networks can be modelled and analyzed using the same methods as for Boolean Networks and Random Boolean Networks (RBN) [16]. This opens for the possibility to generate and visualize attractor basins and the trajectories from initial state to attractors, which may represent the system behaviour.

Those alternative computation paradigms such as cellular computation may offer massive computation power. However, the potential is hard to exploit. Logical design of the hardware and lack of programming methods makes it difficult to unleash the potential computation power. As an alternative to today's top-down design approach an adaptive approach e.g. an Evolutionary Algorithm (EA) [10], show promising results as a design tool for such systems. However, EAs alone lack the scalability required to solve the task of designing the hardware and setup the running conditions required for realistic computation problems. One solution to increase the level of design complexity of EAs is to take inspiration from nature's way of handling complexity. Applying biologically inspired design methods, such as evolutionary algorithms and artificial development, as a design tool for hardware capable of complex computation is not a trivial task. The approach is relatively new and there is little knowledge of how to design the biologically inspired methods toward functional hardware. However, the results found by applying biologically inspired design methods [17] are promising and it is our belief that pursuing this approach will lead to specialized hardware capable of effective computation.

## 3. OBJECTIVES

The main high level objective is to exploit Artificial Development to the creation of basic computational elements that fit in an unconventional computation paradigm. This topic has opened several research directions which aim at the following research questions:

- Which theoretical and experimental approaches can be used to find methods for the specification of input data and interpretation of the discrete dynamics of cellular machines?
- How to use those methods on problems where modeling on cellular machines is advantageous? Problems of interests are in the domain of complex systems, e.g. economical models, clima models or models of gene regulation networks.
- Which is the correlation between developmental and structural complexity of cellular machines? Which is the relation between complexity and the behavior of the system? Investigation, evaluation and comparison of such properties.
- Which is the impact of developmental properties on EvoDevo (Evolution and Development) machines capable of complex computational behavior?

## 4. CA CLASSES

The starting point is an in depth study of CA behaviors and classifications. John von Neumann [9] studied the first cellular automaton in the 1940s, basing his research also on some studies done by Stanislaw Ulam [8]. At that time, the research on cellular automata was performed through analysis of variations produced by a single automaton [11]. In the 1980s Stephen Wolfram changed this approach. He showed that one-dimensional cellular automata could be sufficient to investigate the rules' behavior. Instead of studying single rules, he divided and enumerated subclasses of CA rules in order to group the rules producing similar behavior and study different classes depending on the pattern that the rules were able to generate [2]. Wolfram found out that CA can be grouped in four classes, depending on the type of behavior they produce. In practice, with finite lattices, there is only a finite number of possible configurations and all rules lead to a periodic behavior. However, in theory, the lattice is supposed to be infinite.

### 4.1 Class 1

Almost all initial configurations relax after a transient period to the same fixed configuration. In this class, the outcome of the evolution is determined with probability 1 and it is not dependent on the initial state. In other words, the automata in this class die after one or few evolution steps and the process is not reversible because all the previous information is lost. No patterns, or very few patterns, are produced in the first evolution steps and, afterwards, the CA reaches homogeneous state every time.

### 4.2 Class 2

Almost all initial configurations relax after a transient period to some fixed point or some temporarily periodic cycle of configurations, which is dependent on the initial configuration. Some parts of the initial configuration are filtered out and others are propagated forever. The automata in this class will look like vertical bars or stair steps. The process is not reversible because information is partially lost during the evolution.

### 4.3 Class 3

Almost all initial configurations relax after a transient period to chaotic behavior. The evolution process is completely reversible since the previous state can be predicted analyzing the current state. This class of behaviors is chaotic but not random and the produced data are not noise.

### 4.4 Class 4

Some initial configurations result in complex localized structures that are sometimes long-lived. The information is propagated by the automata at variable speed. The process is non-reversible because the current site values have been influenced by more than one previous configuration [14]. This is the only class with non-trivial automata, which can produce complex behaviors instead of fixed dynamics (trivial by definition) or chaotic dynamics (chaotic behavior is considered to be trivial because it is not random and thereby it is completely reversible). Wolfram [2] proposed that the automata in the class 4 are capable of universal computation. However class 4 is not rigorously defined and thus this hypothesis seems impossible to verify [13] (Cook [18] proved that one dimensional CA rule 110 is capable of universal computation) .

## 5. COMPUTATION AT THE EDGE OF CHAOS

Langton tried to find a relation between the CA behavior and a parameter $\lambda$ [1]. He observed that the basic functions required for computation (transmission, storage and modification of information) are achieved in the vicinity of phase transitions between ordered and disordered dynamics (edge of chaos). He also guessed the location of Wolfram classes in the λ space. Classes 1 and 2 constitute the ordered phase and class 3 constitutes the disordered phase. Class 4 is located somewhere between these two phases of dynamical behavior.

It turned out that it is more likely to find rules capable of complex computation in a region where the value of a parameter (λ) is critical [1]. λ is defined as the fraction of "non-quiescent" states in the rule-table used for the development of the CA. When λ is equal to zero all the rules lead to a quiescent state and when λ = 1-1/k (k is the number of different possible states of a cell) the rule-table is the most heterogeneous. For example, considering a 2D CA with 3 cell types (empty cell, cell of type A and cell of type B), if all the rules constructed with all the possible neighborhood configurations lead to the empty cell, the value of λ is 0. On the other hand, if only 1/3 of the rules generate the empty cell, the value of lambda is 1-1/3=2/3. Incrementing λ from 0 to 1-1/k, it is possible to observe all the types of CA behavior described by Wolfram, crossing phase transitions and going from ordered to chaotic behavior. For certain λ critical values the CA tend to show complex and long-lived patterns. Langton [1] claimed also that class 4 is located somewhere around this critical value of λ, at the Edge of Chaos.

Studies on entropy [1], as a measure of the information carried by each cell during the CA development, and mutual information between the cell and itself at the next time step, support the hypothesis that for λ values in the proximity of phase transitions, it is more likely to have well balanced conditions to support computation. For example, information storage involves low entropy and, on the other hand, information transmission requires increasing entropy. If a system needs to do both in order to perform computation, there must be a tradeoff between levels of entropy. Again this happens in the proximity of phase transitions.

## 6. DISCUSSION

The previous work regarding the edge of chaos [1] [3] [4], may not be conclusive. More investigation is required in order to understand the relation between computation performed by CA and different λ values, the relationship between complexity and computation, the intrinsic properties of the development process of CAs. So far, it seems that there is a correlation between phase transitions in the CA regime and the ability to perform computation.

A developmental system is a system in which an organism can develop (e.g. grow) from a zygote to a multicellular organism (phenotype) according to specific local rules, represented by a genome (or genotype), and the interactions with the environment. A CA can be considered as a developing organism, where the developmental process is itself a computation. The genome specifications and the gene regulation information control the cells' growth and differentiation. The behavior of the CA is represented by the emergent phenotype, which is subject to size and shape modifications, according to the cellular changes along the developmental process. Such dynamic developmental systems can show adaptation, self-modification or plasticity properties. In the next chapter we present a possible minimalistic experimental setup.

## 7. POSSIBLE EXPERIMENTAL SETUP

Our current work consists on an investigation with a minimalistic developmental model based on a two dimensional cellular automata. The number of cell types is set to three instead of two in order to keep the property of multicellularity (two types of cells plus the empty or dead cell). A single cell is placed in the centre of the development grid and it will develop using a developmental table based on Von Neumann's neighborhood. All the possible regulatory input combinations are explicitly represented in a development table which consists of 243 ($3^5$) configurations (with three types of cells and a neighborhood of size five, the number of possible cellular states is 243). The table is generated using a "random table method" [1]. For each λ value from 0 to 1 (and intervals of 0.01), the relative developmental table is generated as follows: with probability 1-λ, the cell type at the next developmental step is quiescent; with probability λ, the cell type at the next developmental step is a generated by a uniform random distribution among the other two cell types. This method is computationally efficient compared to the "table walk through method" [1] but, on the other hand, it does not assure that the required value of λ is precisely reflected in the development table. However, it does guarantee enough accuracy. To ensure that cells will not materialize where there are no other cells around, a restriction has been set: if all the neighbors of a quiescent cell are quiescent, the cell will be quiescent also in the following development step. A more detailed description of the development model is given in [15].

With this configuration it is possible to investigate the developmental process from a zygote to a multicellular organism with development rules that cover a wide spectrum of the λ space. Depending on the λ value, it may be possible to find a correlation between properties of the developmental mapping and the behavior of the automata in terms of developmental complexity, structural complexity, CA attractor length, CA trajectory length, different transient phases.

## 8. PRELIMINARY RESULTS

The first experiment has been performed on a 3 by 3 CA. With 9 cells, the maximum number of possible states is $3^9$=19 683. We immediately noticed that every cell has a development process which is completely dependent on the neighborhood configuration of each other. In other words, many neighbors are overlapping and the development process is annihilated by this superimposing, there is not enough space for a "free" development. This situation is shown in Figure 1a, where the less overlapping neighborhoods have 2 cells out of 5 in common (in green, 1/2). Moreover, there is not enough space for structure growth and signal propagation.

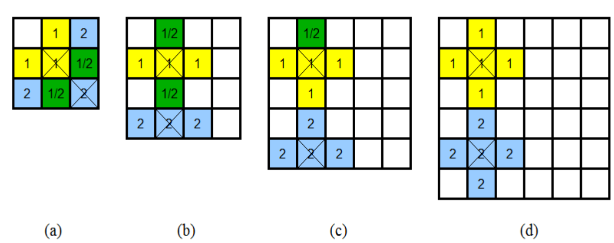

**Figure 1. Representation of overlapping neighborhood with different 2D CA sizes. Cells 1 and 2 are marked with a cross. Neighbors are marked with 1 and 2 respectively.**

The same mechanism is shown in Figure 1b, 1c and 1d, for a 4 by 4, 5 by 5 and 6 by 6 CA, respectively. In this last case, several cells have no overlapping neighborhood.

In [15] we have presented the results for 4 by 4 and 5 by 5 CA. At the moment, we are running experiments using 6 by 6 arrays. Unfortunately the experiments are not finalized. However, early results show a correlation between genomic composition and developmental properties, e.g. trajectory, attractor or transient length. In Figure 2, we illustrate some early results on a 3 by 3 cellular array. It is possible to identify that average trajectory and attractor length have higher values in the proximity of a "critical" value of λ region, around 0.66. As we have said earlier, this result is slightly reduced and mitigated by the overlapping scenario. Anyway, there is a clear correlation between the value of λ and the behavior of the resulting organism, observed as length of the trajectory and the attractor. It might be possible to find specific characteristics or properties of the genotype in order to predict how the phenotype will develop. For more exhaustive and detailed results, please refer to [15].

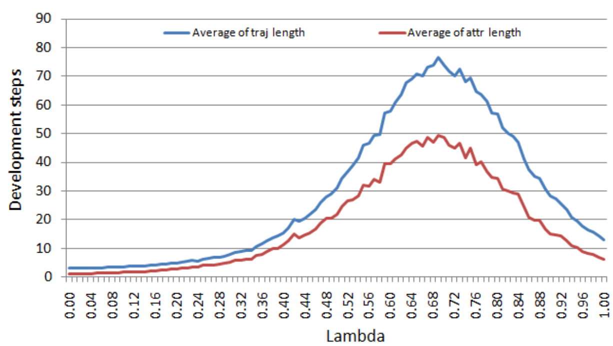


**Figure 2. Results of 3x3 organism, average trajectory and attractor length plotted as function of Lambda.**

## 9. CONCLUSIONS

In this paper, Wolfram's CA behavior classification is described together with the "edge of chaos", a region in the rule space where there is a transition between ordered and chaotic regimes. An experimental setup for a minimalistic developmental system that can be exploited to investigate correlations between CA behaviors and cellular regulative properties is presented.

Early results show that it is possible to measure properties of the genome composition (here measured with λ parameter) as an indicator of how resulting organism will develop. In many studies regarding CA and in particular their development process and the produced computation, artificial organisms have shown remarkable abilities of self-repair, self-regulation [6] and phenotypic plasticity [7] [19].

Our guess is that many of the CA rules which lead to these behaviors are in the frozen or ordered region (Wolfram's classes 1 or 2) and not in the "edge of chaos".

## 10. REFERENCES

[1] C. Langton. Computation at the Edge of Chaos: Phase Transitions and Emergent Computation. Physica D Volume 42 (1990) pp. 12-37

[2] S. Wolfram. Universality and Complexity in Cellular Automata. Physica D Volume 10 Issue 1-2 (1984) pp. 1-35

[3] N. Packard. Adaptation Toward the Edge of Chaos. Dynamic Patterns in Complex Systems. Kelso, Mandell, Shlesinger, World Scientific, Singapore Press (1988) pp. 293-301

[4] M. Mitchell, J. Crutchfield, P. Hraber. Dynamics, Computation and the "Edge of Chaos": A Re-Examination. Santa Fe Institute Working Paper 93-06-040, Complexity: Metaphors, Models and Reality, Addison-Wesley (1994) pp.497-513

[5] G. Tufte. The discrete dynamics of developmental systems. IEEE Congress on Evolutionary Computing, IEEE Press Piscataway, NJ, USA, (2009) pp. 2209-2216

[6] J. Miller. Evolving a Self-Repairing, Self-Regulating, French Flag Organism. Gecco 2004. Springer-Verlag Lecture Notes in Computer Science 3102, (2004) pp. 129-139

[7] G. Tufte and P. Haddow. Extending Artificial Development: Exploiting Environmental Information for the Achievement of Phenotypic Plasticity. Springer Verlag Berlin Heidelberg, ICES 2007 – LNCS 4684, (2007) pp. 297-308

[8] S. Ulam. Los Alamos National Lab 1909-1984. Los Alamos: Los Alamos Science. Vol. 15 special issue, (1987) pp. 1-318

[9] J. Von Neumann. Theory and Organization of complicated automata. A. W. Burks, (1949) pp. 29-87 [2, part one]. Based on transcript of lectures delivered at the University of Illinois in December 1949.

[10] J. H. Holland. Genetic Algorithms and Adaptation. Technical Report #34 Univ. Michigan, Cognitive Science Dept. (1981)

[11] E. Berlekamp, J. H. Conway, R. Guy. Winning ways for your mathematical plays. Academic Press, New York, NY (1982)

[12] M. Sipper, The Emergence of Cellular Computing, Computer, vol. 32, no. 7, (1999) pp. 18-26

[13] M. Mitchell, P. T. Hraber and J. T. Crutchfield. Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computation. Complex Systems, vol. 7, (1993) pp. 89-130

[14] A. Turing. On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, Series 2, 42, (1936) pp 230–265

[15] G. Tufte and S. Nichele. On the Correlation Between Developmental Diversity and Genomic Composition, to appear in GECCO '11: Proceedings of the 20th annual conference on Genetic and evolutionary computation, New York, NY, USA (2011). ACM. (IN PRESS)

[16] C. Gershenson. Introduction to Random Boolean Networks, Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems, ALife IX, (2004) pp. 160-173.

[17] S. Nichele and G. Tufte. Trajectories and Attractors as Specification for the Evolution of Behavior in Cellular Automata, IEEE Congress on Evolutionary Computation. IEEE conference proceedings (2010) pp. 4441-4448

[18] M. Cook. Universality in Elementary Cellular Automata, Complex Systems 15 (2004) pp. 1-40

[19] P. J. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem, in GECCO '99, Genetic and Evolutionary Computation Conference (1999), pp. 35-43