Chapter 33

Phase Transitions in the Computational Complexity of "Elementary" Cellular Automata

Sitabhra Sinha[1]
Center for Condensed Matter Theory, Department of Physics, Indian Institute of Science, Bangalore 560012, India
and
Condensed Matter Theory Unit, Jawaharlal Nehru Center for Advanced Scientific Research, Bangalore 560064, India
sitabhra@physics.iisc.ernet.in

A study of the computational complexity of determining the "Garden-of-Eden" states (i.e., states without any pre-images) of one-dimensional cellular automata (CA) is reported. The work aims to relate phase transitions in the computational complexity of decision problems, with the type of dynamical behavior exhibited by the CA time evolution. This is motivated by the observation of critical behavior in several computationally hard problems, e.g., the Satisfiability problem (SAT) [1]. The focus is on "legal" CA rules, i.e., those which obey the quiescence and symmetry conditions [2]. A relation exists between the problem of "Garden-of-Eden" states determination and the SAT problem. Several CA rules (e.g., CA rules 4, 22 and 54) are studied in detail to establish the occurrence of phase transition. Finite-size scaling exponents corresponding to the critical behavior are obtained. Based on these exponents, a new quantitative classification of "elementary" cellular automata into 5 classes is proposed.

[1] *Present address*: The Institute of Mathematical Sciences, C I T Campus, Taramani, Chennai 600113, India. (*E-mail*: sitabhra@imsc.res.in)

1 Introduction

The spontaneous emergence of complexity in various physical systems is one of the most remarkable features of the natural world. There have been several attempts to define and quantify "complexity" (in the dynamical sense). Measures such as information entropy, mutual information, etc., have been used with varying degrees of success. As yet, there does not seem to be an unambiguous indicator of the degree of complexity in a system. However, in the field of computer science, a measure of complexity, called *computational complexity* which measures the increase in system resources required to solve a problem as the problem increases in size, has been well-established since the 1960s [3]. A limitation of using this measure on actual physical systems is that it is defined only for processes taking discrete state values, as well as evolving in discrete space-time [4]. Further, in order to study the computational complexity of a process, we need to pose a non-trivial decision problem for the system in question. The simplest class of models that shows interesting dynamical behavior and for which "hard" decision problems can also be posed is that of *cellular automata* (CA). They exhibit these desired properties with the least number of theoretical assumptions and the least numerical effort. It is interesting to note that some CA are also the simplest systems capable of universal computation [2].

In this paper, we have studied the computational complexity of one-dimensional cellular automata. In particular, we have observed phase transitions in the computational hardness of a decision problem in CA. Previous work has indicated a relation between phase transition and the occurrence of computational complexity in other decision problems, such as the *Satisfiability* problem (SAT) [1] and the number-partitioning problem [5]. Whether the phase transition in relation to computational complexity has any connection to qualitative changes in the dynamical behavior of CA is of particular interest. For at least one of the CA we have studied, a relation exists between the computational phase transition and the existence of a dynamical phase transition [6].

In Section 2, we have defined cellular automata and described the particular type of one-dimensional CA that we have considered for our study. This section also discusses the computationally hard decision problem for CA which is the focus of our numerical investigations. The back-tracking algorithm used for obtaining many of the numerical results is discussed in Section 3. Section 4 contains the results of the computational study, with details for four particular CA rules. A new classification scheme of CA, involving 5 classes, has been proposed in this section. Finally, in Section 5, the relevance of this study to other systems is discussed.

2 Cellular Automata

Cellular automata (CA) are simple lattice models of dynamics evolving in discrete state space at discrete time intervals. One-dimensional CA is characterized by k, the number of states available to each automaton (located at each cell), and r, the radius of the neighborhood whose states determine the state of an automaton at the next time step. The mapping between a particular input, i.e., the state configuration of $(2r + 1)$ cells at time step n, to a particular output (at time step $n + 1$), is determined by the CA rule governing the system evolution. Different rules give rise to a wide variety of dynamical behavior, from a spatially homogeneous, time-invariant pattern, to spatially inhomogeneous, time-varying patterns. Examples of patterns generated by different CA rules are shown in Fig. 1. In the following we shall consider $k = 2, r = 1$ CA (the so-called "elementary" CA). In particular, the "legal" rules will be considered, i.e., those rules for which the state configuration 000...000 is a *null configuration* (i.e., invariant under application of the rule), and for which the rule gives identical output for symmetrically related input.

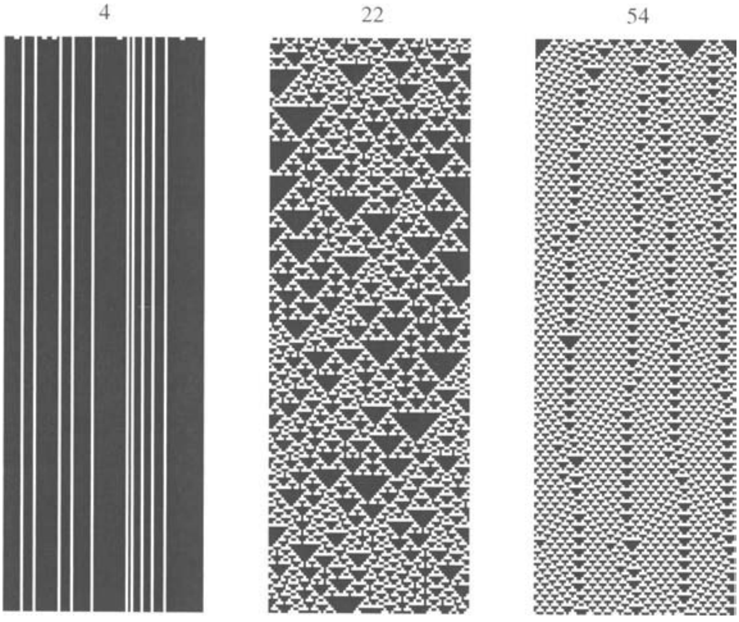

Figure 1: CA time evolution for CA Rules 4, 22 and 54.

Cellular automata models are studied as models for emergence of complex dynamical behavior. There have been some efforts at establishing the existence of a phase transition from ordered to chaotic behavior with complexity emerging at the transition [7]. One-dimensional CA have been classified morphologically into 4 classes by Wolfram [2], with Class IV corresponding to the most complex, marked by the presence of coherent, propagating structures (e.g., "gliders") [8]. Langton's work seemed to indicate that, by varying a parameter, one could get rules in Class I and II (which have simple dynamics), and then rules in Class III (which are "chaotic") through a transition, where the Class IV rules are obtained. However, as dynamical "complexity" is not well-defined, this has been subjected to criticism. As computational complexity is quite a well-defined concept, we have tried to use this concept to classify CA rules. The notion involves how computation resources used for solving a computational problem scales with problem size [3].

Recently, it has been shown that several NP-complete problems show phase transition behavior, such that, not all instances of these problems are uniformly hard. Typically, they show a transition from easy to hard to easy instances as a system parameter is varied [1]. This has motivated us to find similar behavior in the case of CA.

The particular computational question that we shall be focusing on is whether a given random binary string is a "Garden-of-Eden" configuration, i.e., one which has no pre-image. This is the simplest query that is very hard to answer, but given the answer, it is very easy to verify the correctness of the solution. The problem of finding a pre-image for a length N sequence under cellular automaton evolution is in the class NP [9]. Therefore, the problem can be written in terms of a known NP-complete problem, e.g., the Satisfiability (SAT) problem.

The rules for Cellular Automata are normally expressed in terms of Disjunctive Normal Form (DNF). However, to express them in terms of the K-SAT problem, one must convert these into Conjunctive Normal Forms (CNF). Immediately, one notices that the pre-image formula for any of the elementary CA can be expressed as a suitable combination of 1-SAT, 2-SAT and 3-SAT.

Take for example Rule 22. The existence of 0 at the i-th site implies that the pre-image x, if it exists, must satisfy the following condition: $(\neg x_{i-1} \vee x_i \vee x_{i+1}) \wedge (x_{i-1} \vee x_i \vee \neg x_{i+1})$, which is a 3-SAT with $\alpha = 3$, while the presence of 1 implies the logical expression $(x_{i-1} \vee x_i \vee x_{i+1}) \wedge (\neg x_{i-1} \vee \neg x_{i+1}) \wedge (\neg x_{i-1} \vee \neg x_i) \wedge (\neg x_{i+1} \vee \neg x_i)$, which is a mixture of 2-SAT and 3-SAT. For Rule 54, $z_i = 0 \Rightarrow (\neg x_{i-1} \vee x_i) \wedge (\neg x_{i+1} \vee \neg x_i) \wedge (x_{i-1} \vee \neg x_i \vee x_{i+1})$, and, $z_i = 1 \Rightarrow (x_{i-1} \vee x_i \vee x_{i+1}) \wedge (\neg x_{i-1} \vee \neg x_i) \wedge (\neg x_{i+1} \vee \neg x_i)$, which are a mix of 2-SAT and 3-SAT. Note that, the existence of a pre-image for evolution under Rule 90 can be shown to be equivalent to the existence of a satisfiable assignment for a pure 2-SAT ($\alpha = 2$) problem. This indicates that the problem is solvable using a polynomial time algorithm. Indeed, a simple linear-time algorithm can be used to determine the existence of a pre-image under CA Rule 90 evolution which takes $4N$ steps in the worst case.

The CA system we have considered for our numerical studies is a chain of N sites, with periodic boundary conditions. All sites in the lattice are updated according to the same CA rule. A variable parameter p ($0 \leq p \leq 1$) is introduced, which gives the probability for each site to independently assume the value 1; else the site has value 0.

3 Backtracking Algorithm

To check if a randomly generated binary string has a pre-image under a given rule is in general a difficult task. For most ($k = 2, r = 1$) CA rules a simple algorithm exists. However, in general, we have to use a modification of the CA Reverse Algorithm [8]. This is a backtracking method where a possible pre-image is gradually constructed, while checking for consistency at each step of the construction. To find whether a given configuration C has a pre-image, a partial pre-image P is constructed, where $2r$ bits, up to and including the starting site P_i, is given. The next unknown bit to the right, P_{i+1}, is chosen consistent with the CA rule-table. If this cannot be done, then P_i is rejected, and another possible partial pre-image is chosen. The process is continued until either, a valid pre-image sequence P has been found, or, all possible partial pre-images have been exhausted. Note that the choice of P_{i+1} may not be unique, as many pre-images can lead to the same string C. If multiple alternatives are present, a branching is made at this point, and the process is continued along one of the branches until a full, valid pre-image sequence is obtained. If at some point, the string can no longer be continued, the algorithm returns to the branch point and resumes one of the remaining alternative choices. This procedure is continued until the algorithm arrives at the bit P_{i-1} preceding the starting bit. It is checked whether the periodic boundary conditions are satisfied. If not, the procedure is begun anew with a new partial pre-image, unless all such alternatives have been exhausted; in which case, the string C is a Garden-of-Eden state. In the worst case, the algorithm will still have to go through 2^N possible pre-images, as is the case with exhaustive search. However, in most cases, the number of possibilities to be checked is much smaller in the backtracking algorithm. This is because a candidate string is quickly rejected if it is found that it can no longer be continued.

4 Results

In the numerical studies of phase transitions, the variable is p, the fraction of 1s in the given binary string whose pre-image (if it exists) is to be determined. The variable p can also be looked upon as the probability of occurrence of 1 at any given site within the string. The order parameter is the fraction of Garden of Eden (GOE) configurations, $f(G)$. This is analogous to the *magnetization* parameter (M) observed in models of magnetic systems, while the variable in such systems is the *temperature* (T). Note that, in such systems, when $T \rightarrow T_c$ (T_c is the *critical temperature*, where the phase transition occurs) the order parameter scales with the variable according to the relation

$$M \sim |T - T_c|^{\beta}$$

where β is the *scaling* or *critical exponent* characterizing the phase transition. Another parameter that shows similar characteristic scaling relation is the *correlation length* (ξ). Near the critical point (i.e., $T \rightarrow T_c$),

$$\xi \sim |T - T_c|^{-\nu}$$

We characterize the phase transitions seen in the problem studied in this paper by obtaining the critical exponents ν and β, through finite-size scaling analysis

[10].

Finite-size scaling is done by observing the order parameter for various system sizes, N. In cases where a simple polynomial algorithm for finding pre-images could not be obtained, we used the backtracking algorithm described in the previous section. The median number of computational steps required to obtain the solution using this algorithm gave a measure of the computational cost involved in solving the problem. This is analogous to the *susceptibility* (χ) parameter observed in magnetic systems, which shows the following scaling relation near the critical point:

$$\chi \sim |T - T_c|^{-\gamma}$$

In the following examples, we have obtained the value of γ for CA Rule 22.

Rule 0. This is possibly the simplest CA, where all input configurations give the output 0. The limit set is the null configuration $(\ldots 000 \ldots)$ which is reached in one iteration starting from any initial string. It follows that all strings except the null configuration are GOE states. Note that, under evolution with CA Rule 0, any output string in which the block '1' occurs anywhere is forbidden. So '1' is the smallest irreducible forbidden word in the formal language generated by the CA, or it is the 'kernel'. As the probability of the non-occurrence of 1 anywhere within a string of length N is $(1-p)^N$, the fraction of GOE states is given by

$$f(G) = 1 - (1-p)^N \sim Np$$

Finite-size scaling data for CA Rule 0 obtained using system sizes $N = 100$, 200 and 400, showed that the value of the critical exponent, $\nu = 1$ ($\beta = 0$). This agrees with the simple argument described above which shows how $f(G)$ scales with N.

Rule 4. This simple rule only outputs 1 for the input triplet 010, all other input sets resulting in an output of 0. Wolfram classified it in Class II. The limit set of this CA rule consists of configurations with only isolated 1s, and is achieved in one iteration from any initial configuration. In particular, any configuration is either an attractor (i.e., invariant with respect to CA evolution) or a GOE configuration. The decision problem of whether a particular binary string has a pre-image or not under CA Rule 4 evolution, can be decided by simply iterating it once in the forward direction and checking whether the string remains invariant under CA evolution. If it is indeed invariant, then it is an attractor, and therefore its own pre-image; if not, it is a GOE configuration. Therefore, the algorithm for solving the decision problem is a simple linear time algorithm, belonging to the class P.

Fig. 2 (left) shows the result of finite-size scaling for system sizes $N = 100$, 200 and 400. The value of the finite-size critical exponent obtained from this scaling is $\nu = 2$ ($\beta = 0$). This is not surprising if we notice that under CA Rule 4, the block "11" is excluded. In other words, it is the smallest irreducible forbidden word under the formal language defined by CA Rule 4. Therefore, $f(G)$ will grow as a function of Np^2 with higher order terms. Alternatively, the order parameter increases as a function of $p\sqrt{N}$, which is observed numerically.

As mentioned before, the decision problem of whether a binary string has a pre-image under a given CA evolution can be posed as a SAT problem. If the i-th bit of the given string is represented by $z(i)$ and that of its pre-image (if it exists) by $x(i)$, then the SAT corresponding to Rule 4 is

$$z_i = 1 \Rightarrow \neg x_{i-1} \wedge x_i \wedge \neg x_{i+1} (1 - \mathrm{SAT}, \alpha = 3)$$

and

$$z_i = 0 \Rightarrow x_{i-1} \vee \neg x_i \vee x_{i+1} (3 - \mathrm{SAT}, \alpha = 1)$$

Therefore, the existence of a pre-image for an arbitrary string under evolution with CA Rule 4 is equivalent to the existence of a satisfiable assignment for a corresponding mixed (1-SAT + 3-SAT) problem. The relative fraction of 1-SAT clauses in the formula is

$$f(1 - \mathrm{SAT}) = 3\mathrm{p}/(1 + 2\mathrm{p})$$

For $p = 0$, the problem is a pure 3-SAT, and hence, satisfiable, whereas for $p = 1$, it is a pure 1-SAT, and unsatisfiable (UNSAT). For $p > 0$, at the thermodynamic limit ($N \to \infty$), the problem is always UNSAT because of the 1-SAT component.

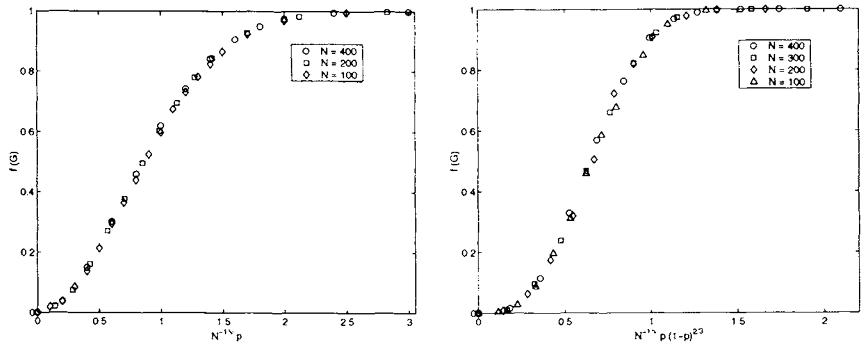

Figure 2: Finite-size scaling for fraction of Garden-of-Eden configurations for (left) CA Rule 4 ($\beta = 0, \nu = 2$) and (right) CA Rule 54 ($\beta = 0, \nu = 3$).

Rule 54. This CA is probably the only representative of Wolfram Class IV type automata among the legal, elementary CA rules. Its behavior is transitional between the simple, regular Classes I and II and the chaotic Class III rules [14]. Like the CA belonging to Classes I and II, any arbitrary initial configuration tend towards simple spatio-temporally periodic states under evolution with Rule 54, but unlike them, starting from these initial states, extremely long transients (with "chaotic" behavior) is observed before a periodic configuration is reached. For small times, therefore, Rule 54 resembles a Class III rule in its dynamical behavior. It appears to possess a natural measure, but as mentioned, this is approached very slowly from typical initial configurations. Further, the spatio-temporal patterns it produces are characterized by large coherent regions, punctuated by slowly moving defects. It has been conjectured that these defects convey information from one region of the CA to another.

Although the dynamics seems to indicate that Rule 54 is in some sense a marginal or critical CA, the decision problem is computationally easy to solve in this case. This is because the set of irreducible forbidden words for evolution under Rule 54 is a finite set: 10110, 10101, 01101 and 10111101. To decide whether a string has a pre-image or not, one simply has to check whether any of the above blocks occur anywhere within the string. Therefore, the problem can be solved using a polynomial time algorithm. In Fig. 2 (right) one sees the result of finite-size scaling which gives the value of the critical exponent $\nu = 3$ ($\beta = 0$).

Rule 22. This rule is a member of the class of additive CA rules, as the rule outputs a 1 if the sum of the bits in the input triplet is 1 (i.e., for the input triplets 001, 010 and 100); else it outputs 0. It has a unique stable, invariant measure called the *natural measure* which is approached rapidly by any initial measure under evolution with CA Rule 22. The dynamical behavior of the CA is placed under Wolfram Class III, as the irregular spatio-temporal patterns generated by it are typical of the other members of this class.

In spite of the simple appearance of patterns created by this CA, non-trivial long-range effects have been reported, similar to critical phenomena [11]. The resulting patterns were shown to be neither periodic nor random, but somewhere in between, due to the presence of subtle correlations. Later, extensive Monte Carlo simulations have shown that the system behaves like a normal one-dimensional statistical ensemble with a critical point $p_c = 0$ [6]. As $p \to p_c$, a critical slowing down to the asymptotic value of the average number of sites with 1 is observed, having a dynamical critical exponent of unity.

The results of finite-size scaling can be seen in Fig. 3 (left). Note that unlike the other rules for which results are given here, this CA has two finite critical exponents, $\nu = 4$ and $\beta \simeq 1/3$. The latter exponent was zero for the other CA, implying that, those were first order phase transitions, as the existence of a non-zero value of β is indicative of a second order phase transition. This can be related to the fact that the set of irreducible forbidden words for this CA has an infinite number of elements. The smallest of such excluded blocks are 10101001 and 10010101. The number of such blocks or 'kernels' increase rapidly with the block length being considered [12], so that, simple algorithms based on checking for the existence of such blocks cannot be devised to solve the decision problem.

The non-zero value of β can possibly be also related to the fact that the sequences generated by one iteration of Rule 22 do not constitute a finite-complement regular language [13], i.e., a regular language with a finite number of excluded blocks. Note that these excluded blocks are the building blocks of GOE states. Further, the finiteness of the number of excluded blocks would have implied the existence of a computationally easy algorithm to identify GOE states, namely, by simply checking for the existence of each of the words of the regular language. The presence of any one would mean that the configuration necessarily lacks a predecessor and is therefore a GOE state.

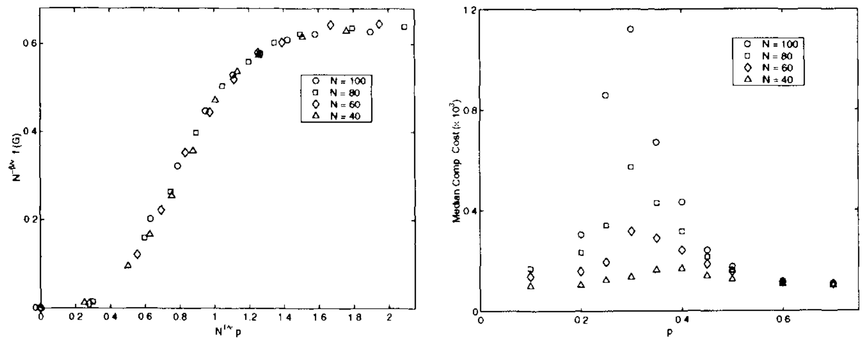

Figure 3: (left) Finite-size scaling for fraction of Garden-of-Eden configurations for CA Rule 22 ($\beta = 1/3, \nu = 4$) and (right) the median computation cost for identifying GOE configurations in CA Rule 22 using backtracking algorithm.

Fig. 3 (right) shows how the median computational cost of using the backtracking algorithm for identifying GOE states varies with p. The peak computational cost increases with N rapidly, while, the value of p at which the peak occurs tends to $p = 0$. This suggests that, at the thermodynamic limit, the peak computational cost will occur at $p_c = 0$. Finite-size scaling of the computational cost data shows that the ratio of the critical exponents $\frac{\gamma}{\nu} = 2$. This implies that there is no anomalous dimension for this system.

The above examples gave a detailed illustration of the nature of the phase transition observed in these systems. The same analysis can be extended to other CA rules. In fact, all the 32 one-dimensional "legal" CA rules have a characteristic value for the critical exponent ν, which fall in one of the following classes:

$\nu = 1$: Rules 0, 94, 122, 126, 200, 222, 250 and 254.
$\nu = 2$: Rules 4, 32, 36, 128, 160, 178, 182, 232, 236.
$\nu = 3$: Rules 50, 54, 72, 76, 108, 132, 218.
$\nu = 4$: Rules 22, 104, 146, 164.
$\nu \to \infty$: Rules 90, 150, 204.

This suggests an extremely simple quantitative classification of one-dimensional CA into *five* classes.

5 Discussion

The Wolfram classification scheme for one-dimensional CA, although the most well-known one, has fundamental problems. Not only is it entirely qualitative in nature, but it is also extremely dependent on the choice of the initial configuration. Based on Wolfram's work, Culik and Yu [15] had suggested a more specific definition of the four classes of CA according to their behavior on finite configurations. It was shown that the membership of a CA to a class in this scheme is formally undecidable in general, for each of the four classes.

Other classification schemes have also been proposed, which include, the classification according to mean field approximation [16], the types of pre-image formula [17], the properties of finite sequences without pre-images [12], etc. The classification proposed in this paper is the *simplest quantitative* scheme among all those present in the literature so far. Furthermore, the number of classes comes out naturally from the value of the exponents, rather than being imposed arbitrarily, as is the case for most such schemes. The connection with criticality also makes the scheme appealing.

The relation of the classification on the basis of computational hardness to the dynamical behavior of a CA is not obvious. In fact, the connection of dynamics to the existence of Garden-of-Eden states have been explored before [18], where the concept of a "topological skeleton" was used to relate the two. CA rules which exhibit mostly skeletal structure in their state-transition graph (e.g., Rule 30) generally show chaotic behavior while having very few GOE states. At the other end, rules which have mostly surface structure (e.g., Rule 4) show more ordered behavior and also have most configurations as GOE states. However, it remains an open question as to whether this suggested relation between the state-transition graph structure and the dynamical properties hold generally in the space of cellular automata.

Note that, a very interesting connection can be made between the dynamical and computational aspects of CA, through mapping them to equivalent neural network models. This can also be linked to Kaneko's work on information theory for multi-attractor dynamical systems, with focus on one-dimensional cellular automata [19]. When viewed from the dynamical systems point of view, the two aspects of information processing that are important are information generation and information storage. A dynamical system with chaos can be viewed as an information source because it amplifies microscopic fluctuations into macroscopic information. Information can be stored in the large number of attractors which are usually found in spatially extended dynamical systems. An example is the Hopfield neural network model of associative memory [20], whose multiple attractors are used to store a large number of patterns, which can be recalled when the network is given a partial or corrupted pattern as input. From this viewpoint of creation and storage of information, a CA can be placed in one of four classes, the classification being essentially the same as that of Wolfram. It will be interesting to see whether a similar connection can be drawn between the classification scheme proposed above and the properties of information generation and storage by CA, in view of the connection to neural network models.

The problem of characterizing the complexity of CA through computational hardness might have relevance to the question of applying CA for secure communication, i.e., for public-key encryption. This kind of application requires the key to be generated in such a manner that is very easy to code but extremely difficult to decode. It is obvious that NP problems are good candidates for generating such keys. CA have been proposed as a possible mechanism for generating such keys. However, for successful application the decoding problem must not only be very hard to solve in the worst case, but should also be hard in the general case.

The scheme of using computational hardness as a measure for complexity can also have implication for other spatially extended dynamical systems. The next higher step is to characterize the complexity of couple map lattices (CML). However, this will involve having a proper definition of complexity classes for computation over real numbers [21]. It will be interesting to see whether broad universality classes will emerge, encompassing a large variety of dynamical systems.

Acknowledgments

I would like to thank Chandan Dasgupta, Vitaliy Mykhaylovskyy, Peter Grassberger, Andreas Engel and Bikas Chakrabarti for helpful discussions. Financial support from JNCASR is acknowledged.

Bibliography

[1] MONASSON, R., R. ZECCHINA, S. KIRKPATRICK, B. SELMAN and L. TROYANSKY, "Determining computational complexity from characteristic 'phase transitions' ", *Nature* **400** (1999), 133–137.

[2] WOLFRAM, S., "Statistical Mechanics of Cellular Automata", *Rev. Mod. Phys.* **55** (1983), 601–644.

[3] GAREY, M. R., and D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman (1979).

[4] Recently, a complexity theory for computation over real numbers has been developed. See, e.g., L. BLUM, F. CUCKER, M. SHUB and S. SMALE, *Complexity and Real Computation*, Springer Verlag (1998).

[5] MERTENS, S., "Phase transition in the number partitioning problem", *Phys. Rev. Lett.* **81** (1998), 4281–4284.

[6] ZABOLITZKY, J. G., "Critical properties of rule 22 elementary cellular automata", *J. Stat. Phys.* **50** (1988), 1255–1262.

[7] LANGTON, C. G., "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation", *Physica D* **42** (1990), 12–37.

[8] WUENSCHE, A. E., "Classifying Cellular Automata Automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter", *Complexity* **4**(3) (1999), 47–66.

[9] WOLFRAM, S., "Computational Theory of Cellular Automata", *Comm. Math. Phys.* **96** (1984), 15–57.

[10] PRIVMAN, V. (ed.), *Finite Size Scaling and Numerical Simulation of Statistical Systems*, World Scientific (1990).

[11] GRASSBERGER, P., "Long-range effects in an elementary cellular automaton", *J. Stat. Phys* **45** (1986), 27–39.

[12] VOORHEES, B. and S. BRADSHAW, "Predecessors of cellular automata states III. Garden of Eden classification of cellular automata", *Physica D* **73** (1994), 152–167.

[13] JEN, E., "Enumeration of preimages in cellular automata", *Complex Systems* **3** (1989), 421–456.

[14] BOCCARA, N., J. NASSER and M. ROGER, "Particlelike structures and their interactions in spatiotemporal patterns generated by one-dimensional deterministic cellular-automaton rules", *Phys. Rev. A* **44** (1991), 866–875.

[15] CULIK, K., L. P. HURD and S. YU, "Computation theoretic aspects of global cellular automata behavior", *Physica D* **45** (1990), 357–378.

[16] GUTOWITZ, H. A., "A hierarchical classification of cellular automata", *Physica D* **45** (1990), 136–156.

[17] JEN, E., "Preimages and forecasting for cellular automata", *Pattern Formation in the Physical and Biological Sciences* (H. F. NIJHOUT, L. NADEL and D. STEIN eds.), Addison-Wesley (1997), 157.

[18] GUTOWITZ, H. A. and C. DOMAIN, "The topological skeleton of cellular automaton dynamics", *Physica D* **103** (1997), 155–168.

[19] KANEKO, K., "Attractors, basin structures and information processing in cellular automata", *Theory and Applications of Cellular Automata* (S. WOLFRAM ed.), World Scientific (1986), 367–399.

[20] HOPFIELD, J. J., "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. USA* **79** (1982), 2554–2558.

[21] SIEGELMANN, H. T., A. BEN-HUR and S. FISHMAN, "Computational Complexity for Continuous Time Dynamics", *Phys. Rev. Lett.* **83** (1999), 1463–1466.