A Brief History of Cellular Automata

PALASH SARKAR

Indian Statistical Institute

Cellular automata are simple models of computation which exhibit fascinatingly complex behavior. They have captured the attention of several generations of researchers, leading to an extensive body of work. Here we trace a history of cellular automata from their beginnings with von Neumann to the present day. The emphasis is mainly on topics closer to computer science and mathematics rather than physics, biology or other applications. The work should be of interest to both new entrants into the field as well as researchers working on particular aspects of cellular automata.

Categories and Subject Descriptors: F.1.1 [**Conputation by abstract devices**]: Models of Computation; K.2 [**Computing Milieux**]: History of Computing

General Terms: Theory

Additional Key Words and Phrases: Cellular automata, cellular space, homogeneous structures, systolic arrays, tessellation automata

1. INTRODUCTION

Cellular automata were originally proposed by John von Neumann as formal models of self-reproducing organisms. The structure studied was mostly on one- and two-dimensional infinite grids, though higher dimensions were also considered. Computation universality and other computation-theoretic questions were considered important. See Burks [1970] for a collection of essays on important problems on cellular automata during this period. Later, physicists and biologists began to study cellular automata for the purpose of modeling in their respective domains. In the present era, cellular automata are being studied from many widely different angles, and the relationship of these structures to existing problems are being constantly sought and discovered.

Next, we would like to clarify the purpose of this survey as compared to other related work. There is an excellent survey of CA by A.R. Smith III [Smith III 1976]. However, it is more than twenty years old. There are also two other surveys [Vollmar 1977; Aladyev 1974] which are quite old. Currently, it is perhaps quite impossible to survey the whole of CA research. There is a good survey on computation theoretic aspects of CA by Culik II et al. [1990]. There are also books on CA [Garzon 1995; Chaudhuri et al. 1997; Wolfram 1986] which cover specific topics of CA research. In this survey we try to cover the major questions asked about CA as opposed to the use of CA in

Author's address: Applied Statistics Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta, India 700035; email: palash@isical.ac.in.

CONTENTS

1. Introduction
2. Classical
   2.1 Beginnings
   2.2 Variants of Cellular Automata
   2.3 Biological Connection
   2.4 Fault-Tolerant Computing
   2.5 Language and Pattern Recognition
   2.6 Invertibility, Surjectivity and Garden of Eden
3. CA Games
   3.1 Firing Squad Problem
   3.2 Game of Life
   3.3 $\sigma(\sigma^{+})$ -Game
4. Modern Research
   4.1 Empirical Study
   4.2 Classification of CA
   4.3 Limit Sets and Fractal Properties
   4.4 Dynamics of CA
   4.5 Computational Complexity
   4.6 Finite CA and its Applications
5. Conclusion

modeling of natural phenomena. We focus on topics which are closer to computer science and mathematics rather than physics or other applications. We believe that such a survey has not been previously attempted, and will prove to be useful to both fresh entrants into this field and to experts working on particular aspects of CA. However, we would like to point out that any review of CA is bound to be incomplete. We have been motivated in choosing topics based on our knowledge and interest. The aforementioned surveys by A.R. Smith III and Culik II et al. have helped us greatly in preparing this work. The bibliography associated with this article is not comprehensive, though we believe that there are sufficient links to almost all aspects of CA. Additional bibliographies can be found in the books mentioned above. An online bibliography on CA is also available at

http://alife.santafe.edu/alife/topics/cas/ca-faq/ca-faq.bib

At this point we would like to make a few remarks on the problem of trying to write a history of any scientific topic. A chronological ordering of ideas is difficult to adhere to, since an idea may be introduced at some point in time, is pursued vigorously for a while, and may disappear from the literature for quite some time, only to be taken up again at a later point. There is almost no final statement on any idea. A thematic grouping of topics is possible and is mostly used. However, in such an approach one might have to include work from different decades under the same group, and this presents its own problems. The scientific temper varies across time, which leads to a distinct difference in the approach to a problem. So even though the topic may be the same, the method and questions may vary considerably. In this paper we try to take a chronological view of work done in the area of cellular automata over the past forty years, and we order the topics based upon their first appearance in the literature. We have divided the work into three broad categories.

—*Classical*: The themes which were more or less influenced by the initial work of von Neumann.

—*Modern*: The themes which were influenced by the work of Wolfram on one hand, and by developments of other branches of computer science on the other hand. In this part we restrict ourselves to topics closer to computer science than physics.

—*Games:* Apart from the Game of Life and σ-game we have also included the Firing Squad problem in this section. The problem formulation of the Firing Squad problem has more of the flavor of a game than a synchronization problem. Also, this problem somehow does not fit into any of the above two classes.

In the rest of the article we abbreviate both cellular automata and cellular automaton by CA. We consider different varieties of CA, but the exact structure meant will always be clear from the context.

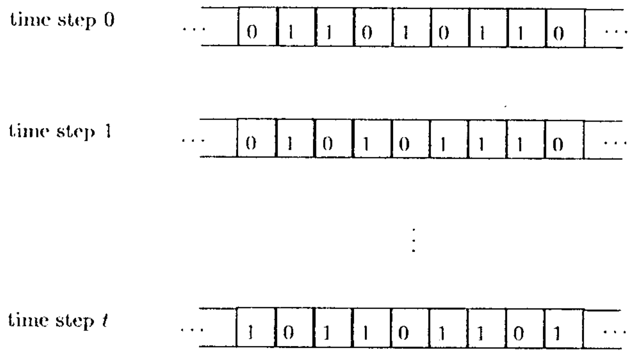

Figure 1. Evolution of an 1-d CA.

2. CLASSICAL

2.1 Beginnings

The simplest description of a CA is a one-dimensional array (possibly two-way infinite) of cells. Time is discrete, and at each time point each cell is in one of a finite set of possible states. The cells change state at each clock tick, and the new state is completely determined by the present state of the cell and its left and right neighbors. The function (called the local rule) which determines this change of state is the same for all cells. The automaton does not have any input, and hence is autonomous. The collection of cell states at any time point is called a configuration or global state of the CA, and describes the stage of evolution of the CA. At time $t = 0$, the CA is in some initial configuration, and henceforth proceeds deterministically under the effect of the local rule, which is applied to each cell at each clock tick (see Figure 1).

Application of the local rule to each cell of the CA results in a transformation from the set of all configurations into itself. This transformation is called the global map, or global rule of the CA. This is a very simple description of a CA, although it is perhaps the most studied structure.

The automaton originally described by von Neumann is a two-dimensional infinite array of uniform cells, where each cell is connected to its four orthogonal neighbors (see Figure 2).

This was originally called a cellular space, but the term CA is more popular now. It was introduced by von Neumann [1966] as a formal model of self-reproducing biological systems. Key ideas of the construction can be traced back even earlier to his talk on modeling of biological systems [von Neumann 1963a]. The main purpose of von Neumann was to bring the rigor of axiomatic and deductive treatment to the study of "complicated" natural systems. The basic idea of a self-reproducing automaton is presented in von Neumann [1963a], and is a beautiful adaptation of the idea of constructing a universal Turing Machine (TM). Here we present a brief sketch of the idea.

2.1.1 *Self-Reproducing Automata*. First, let us note that it is not very difficult to imagine the following two kinds of automata. The first kind is an automaton A which when given an instruction I can use it to construct an automaton (or machine) which is encoded by I. In fact, I can be considered to be composed of simpler instructions, each of which is used to construct the basic parts along with instructions which specify how to put these basic parts together. The second automaton (say B) is even simpler. It copies an instruction I into the con-

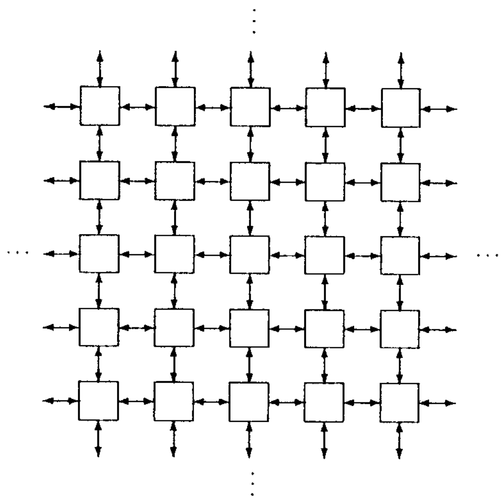

Figure 2. A 2-d CA with von Neumann (orthogonal) neighbourhood.

trol part of some other automaton. Now consider A and B along with a control automaton C, which operates as follows. Given an instruction I, C runs A to create an automaton A_1 corresponding to I and then runs B to copy the instruction I into the control part of A_1. Let D consist of A, B and C. Then, clearly, D is an automaton which requires an instruction I to operate. Let I_D be the instruction which codes D. Let E be an automaton formed from D by copying I_D into the control portion of D. Now it is easy to see that E constructs itself, and hence is capable of self-reproduction. This simple description ignores the coding and other formal details. These were later formalized by von Neumann [1966] himself, in which he describes a cellular space where each cell can be in any one of 29 possible states. The structure is capable of non-trivial self-reproduction in the sense that it can support a universal computer. The process of self-reproduction can be visualized as follows [Smith III 1976]. Initially, the machine is placed in an environment where in each direction there is any amount of hardware available (a "hardware soup"). Following local rules, the initial configuration goes through a sequence of steps whereby it extends an "arm" into the hardware soup and creates a copy of itself, and then detaches the newly created machine from itself. The original proof of von Neumann was simplified and reformulated several times [Arbib 1966; Banks 1970] (see also Smith III [1976]).

The notion of self-reproduction introduced by von Neumann is asexual, in the sense that the offspring is derived from a single parent. In this form of reproduction, the offspring is constructed from a single "genetic" tape which contains an encoding of the machine. Sexual reproduction have also been considered, and Vitanyi [1973] contains a description of a machine which constructs an automaton from two "genetic" tapes, where the resulting offspring is not an exact copy of either parent.

It is important to note that a self-reproducing machine is to be nontrivial, in the sense of being capable of universal computation. Otherwise, a 1-d array with a single quiescent cell and a local rule copying this cell to the left and right neighbors can be considered to be self-reproducing. This brings up the question of CA capable of universal computation and universal constructors. If a machine can construct a set of automata, then it is called an universal constructor over this set. If this set contains the automaton itself, then it is self-reproducing. Before we discuss the question of universal computation, we briefly mention the general problem of pattern replication.

Amoroso and Cooper have described in an interesting paper [Amoroso and Cooper 1971] 1-d and 2-d CA, which after many steps finitely reproduces its initial pattern. The rule used is very simple. For 1-d, it is the sum of the left neighbor and itself modulo k, where k is the number of states a cell can assume. For 2-d, the rule is modified to include the neighbor vertically above the cell. A generalisation to higher dimensions is conjectured in Amoroso and Cooper [1971] and proved in Ostrand [1971]. Moreover, the pattern "reproduces" in a quiescent environment if k is prime. The CA rule used is linear, and is one of the early examples of linear CA.

2.1.2 *Computation Universality.* It is not very difficult to see that a CA is capable of universal computation. The basic idea is that a CA can perform a step by step simulation of a single tape Turing Machine (TM). For convenience, assume that the tape of the TM is two-way infinite. Each cell of the simulating CA will have two components. The first component stores the tape symbol of the corresponding cell of the TM tape, and the second component indicates whether the head is scanning the corresponding cell of the TM. Then, from the TM's transition function, it is easy to derive the local rule for the CA. The essential idea is the following.

(1) If the head is not scanning the cell or its left or right neighbor, the contents of the cell do not change.

(2) If the head is scanning the left cell and there is a right move, then in the next step the head scans the present cell. Similarly for the other direction.

(3) If the head is scanning the cell, then at the next clock tick, the contents of the first component of the cell is updated and the head does not scan the cell anymore.

Note that this step for step simulation of TM by CA destroys the inherent parallelism of CA. There have been attempts to bring out the power of this parallelism [Smith III 1972]. Later work has shown how to simulate TM by reversible CA [Dubacq 1995]. There exists a universal CA A_U with 14 states, which can simulate step by step any CA whose initial configuration and local rule are encoded as an initial configuration of A_U (see Culik II et al. [1990]). Computation universality of one-way CA and totalistic CA (see Section 2) have also been proved [Culik II et al. 1990]. The problem of deciding whether a CA is computation-universal based on the local rule is undecidable, since otherwise the problem of deciding whether a Turing machine is universal would be decidable. See Martin [1994] for additional work on universal CA and its *S-m-n* form.

2.1.3 *CA Tradeoffs.* An early technical question regarding CA was the different kinds of tradeoffs

—between the size of cell (number of possible states) and the size of the neighborhood and

—between the size of cell and the speed of computation.

The idea of tradeoff is an immediate consequence of reformulation of von Neumann's original proof of self-reproducing machines. The original CA described by von Neumann used 29 states per cell. Codd [1968] gave an 8-state machine. Arbib [1966] provided a simple description where each cell can execute a short program–and hence the number of states per cell is large. Banks [1970] provided a 4-state cell which could be used to build a self-reproducing CA. Each of these constructions are for 2-d infinite CA and uses the so-called von Neumann or 5-cell (orthogonal ones and itself) neighborhood.

Generalization of these tradeoff ideas for construction and computation universal machines is natural and has been studied in some depth. The simplest known construction universal machine with 4 states per cell and von Neumann neighborhood is that of Banks [1970]. He has also described the simplest known computation universal 2-d CA (3 states per cell and von Neumann neighborhood). However, for 9 cell or unit square neighborhood (also called Moore neighborhood), 2 states per cell is sufficient and a particular local rule called "Game of Life" (see Section 3) has been shown to be computation-universal [Smith III 1976]. Smith III [1971] provides a list of neighborhood size versus state set size tradeoff results for computation-universal 1-d CA capable of self-reproduction.

The other kind of tradeoff results is related to simulation of a CA by another CA, which is a basic technique for proving results on CA. Specialization of such results to computation-universal CA yields the results just described. It has been observed (but not proved) that the cost of reducing neighborhood or increasing speed leads to an increase in the size of the state set. For a neighborhood of M cells and n states per cell, the size of the state set increases to about M^n when reduction is to Moore neighborhood (a generalization of the 9 neighborhood for 2-d CA). Reduction of Moore neighborhood to von Neumann neighborhood is difficult, and increases the state set size from n to n^V, where V is the volume (number of cells) in a d-dimensional sphere of radius $2d^{3/2}$ [Smith III 1971]. For the 2-d and 3-d cases, this cost can be significantly reduced [Butler 1974; Hamacher 1971]. Simulations can be carried out with neighborhoods smaller than von Neumann. For example, a neighborhood consisting of the cell itself and a neighbor in each dimension suffices for a step by step simulation of an arbitrary CA. In fact, the cell itself can also be left out [Smith III 1971]. If a strict step by step simulation is not required, then the initial encoding may be omitted, and the CA can itself perform the initial encoding. The reverse tradeoff decreasing the state set size by increasing the neighborhood is also possible [Smith III 1971]. See also Mazoyer and Reimen [1992] for later work on CA speed-up.

Given a CA, it is possible to design another CA which simulates the given CA k times faster at a cost of increase of state set size, assuming Moore neighborhood before and after simulation [Smith III 1971]. Both decrease in neighborhood and speed-up can also be achieved at a cost of increase in the state set size. But there seem to be no theoretical results on the limits of the tradeoff possible. For example, assuming finite neighborhood, what is the maximum speed-up possible at a cost of increase in state set size? Investigation of this and similar questions can lead to interesting results.

2.2 Variants of Cellular Automata

A CA is characterized by four features: the geometry of the underlying medium which contain the cells; the local transition rule; the states of the cell; and the neighborhood of a cell. In the following paragraphs we briefly discuss different types of CA that can arise by varying the four features mentioned above. To the best of our knowledge, these cover the several variations considered in the literature.

2.2.1 *Cell States*. The cells of a CA can assume one of a finite number of possible states at any point of time. Usually there is one particular state, called the quiescent state, such that the local rule takes a cell to the quiescent state, if all its neighbors are in the quiescent state.

A CA where the cells can have different state sets is called a polygeneous CA. Such CAs have not received much attention except for the work of Holland [Burks 1970, Essay 15]. The case where the state sets of all cells are the same is the usual one. This set can have an algebraic structure. For linear CA, the state set is usually taken to be a field [Martin et al. 1984], though CA with state sets Z_m (the integers modulo m), for arbitrary m have also been studied [Ito et al. 1983]. In the VLSI context, this set is taken to be $\{0, 1\}$, the field of two elements.

A CA can be visualized as a collection of a set of finite automaton. Each cell of the CA is an individual finite automaton. Though it is possible to allow each cell to assume infinitely many states, such kinds of CA have not been studied. However, in Litow and Dumas [1993], CA is described for which the temporal sequence of a cell is an algebraic series, and hence the cell can store an arbitrary integer.

In the study of limit sets of CA evolution, it has been necessary to equip the state set with the discrete topology [Culik II et al. 1989].

2.2.2 *Geometry*. This can be a d-dimensional (possibly infinite) grid. Usually, the term CA is used for such structures. In case of finite grids, it is possible to define different boundary conditions. The grid is supposed to have a periodic boundary condition in some dimension if it is considered folded in that dimension. The dimension has a fixed boundary condition if the extreme cells are considered to be adjacent to cells in some prespecified state whose value does not change during the computation. In case this prespecified state is the quiescent state, the boundary condition is called a null boundary condition. For linear CA, the quiescent state is the state zero. Among the fixed boundary conditions, only the null boundary condition has been studied seriously. But see Martin et al. [1984] for a brief discussion of other possibilities. It is also possible to consider one end to have periodic boundary condition and the other end to have fixed boundary condition [Bardell 1990].

A more abstract way of defining the geometry is through group graphs. The following definition is from Harao and Noguchi [1978]. A group graph is a tuple $N = (G, h)$, where G is a group which defines the nodes for the cells and h defines a map from G to G^k by $h(g) = (h_1 \circ g, \ldots, h_k \circ g)$, where $h_i \in G$ and $\circ$ is the group operation. The map h provides the neighborhood for the cells. The concept of group graph is a convenient way to describe "uniform" geometry—a connection pattern which "looks the same" at all points. Nonuniform connections have also been studied, though the relation between uniform and nonuniform geometry has not been fully understood (see Jump and Kirtane [1974]).

So far we have considered what is called static CA—the node set and the interconnection pattern do not change with time. It is possible to consider node static CA where the node set does not change with time, but the interconnection pattern may change. Such a structure is still considered static and has not received much attention (see Varshavsky et al. [1970]). However, dynamic CA—both node set and connections may change—have been studied extensively due to its use in modeling of biological systems. For example, the work of Lindenmayer [1968], as described in Section 2, falls in this category.

Recently, there has been a recent interest in studying CA over Cayley graphs [Machi and Mignosi 1993; Roka 1995; 1994].

2.2.3 *Neighborhood*. In some cases such as group graphs, the geometry itself determines the neighborhood of a cell. However, if we are considering a d-dimensional grid it is possible to define different kinds of neighborhood. The von Neumann (orthogonal) neighborhood and the Moore (unit cube) neighborhood have already been mentioned in connection with the tradeoff results. It is possible to define input and output neighborhoods of a cell. A cell takes its input from its input neighborhood and its state is available to the cells of its output neighborhood. If the sizes of the input and output neighborhoods are equal, then the CA is balanced. For balanced but nonuniform neighborhoods, the connection to uniform neighborhood has been studied in Jump and Kirtane [1974]. A variant of CA where the local rule depends on the sum of the states of the neighboring cells is called totalistic CA, and was introduced by Wolfram. Computation universality of this kind of CA have been proved [Culik II 1990].

2.2.4 *Local Rule*. The local rule is usually assumed to be deterministic. This, however, is not necessary, and nondeterministic maps have been studied in connection with language theory [Smith III 1972; Seiferas 1974; Mahajan 1992] and reliable computation [Nishio and Kobuchi 1975]. A CA where each cell has its own local rule is called hybrid. Such structures have been studied in connection to VLSI applications [Serra et al. 1990; Chaudhuri et al. 1997; Sarkar and Barua 1998b]. It is possible for a cell to change its local rule at each time step. In the VLSI context, this is called a programmable CA [Nandi et al. 1994], and in theoretical studies on CA the structure has been called a tessellation automata.

Next, we discuss three variants of CA which have received more attention.

2.2.5 *Tessellation Automata*. This is a CA with an input line distributed to all cells. The setup can be visualized as each cell having a finite set of local rules and the input is used to choose the particular local rule to apply. See Yamada and Amoroso [1969; 1971] for a nice discussion on tessellation spaces. An interesting problem which is inherently tessellation-automata-theoretic is the completeness problem, and related to the Garden of Eden problem for CA. The problem is stated as follows. Starting from an initial configuration with only one nonquiescent state, is it possible to apply input to drive the automaton to any specified finite configuration? If the answer is yes for some subclass of automata, then the subclass is called complete. There are only partial answers to this question [Yamada and Amoroso 1970; Maruoka and Kimura 1974; 1977]. Tessellation automata have also been called time-varying CA and their formal language-theoretic properties have been studied [Mahajan 1992].

2.2.6 *Iterative Automata*. This is a CA where only one particular cell is given an input. Such structures have been considered in connection with language recognition studies [Kosaraju 1975; Seiferas 1974; Chang et al. 1988]. Different tradoff results (similar to CA) for this class have been considered [Cole 1969]. In Smith III [1972], it is shown that this class is an inherently slower device than the usual CA. (Note that in this case the input is provided one symbol at a time to a particular cell, whereas, for a CA, the input is the initial configuration.) Iterative automata languages contain the context-free languages [Kosaraju 1975]. A 1-d iterative automaton requires $O(n^2)$ steps to accept a string of a CFL of length n. The nondeterministic 2-d version of iterative automata can accept in linear time any language accepted in linear time by a nondeterministic multihead TM with a tape of arbitrary dimension [Seiferas 1974]. The paper also contains the result that the nondeterministic d-dimensional iterative spaces can accept in linear time any language accepted in time n^d by a nondeterministic multihead TM but with a 1-d tape.

An interesting application is a linear time multiplier designed by Atrubin. The binary representation of the multiplicands are fed to the first cell (least significant digit) first and the product is output from the first cell (again, least significant digit first) with no delay. See Knuth [1973] for a good exposition of the algorithm. Iterative linear arrays have also been used in VLSI applications [Kung 1988].

The concepts of tessellation and iterative automata can be generalized to tessellation and iterative graph automata by defining such structures on group graphs [Smith III 1976].

2.2.7 *One-Way CA*. A one-way CA allows only one-way communication, i.e., in a 1-d array each cell depends only on itself and its left neighbor. One can also consider dependence on the cell and its right neighbor. However, both-side dependence is not allowed. This lack of two-way flow of information can be considered to be a restriction on the power of the automaton. However, there are results which indicate otherwise. Morita [1992] has shown the computation universality of 1-d, one-way reversible CA. Language recognition properties of one-way CA have also been studied [Chang et al. 1988; Ibarra et al. 1985a]. However, Terrier [1996] provides an example of a language which is not recognizable in real time by one-way CA or iterative CA, but recognizable in real time by CA. One-way versions of the iterative automata have been defined and their properties carefully studied [Chang et al. 1988]. It turns out that they can accept PSPACE-complete languages and the languages accepted by a linear time-bounded alternating TM. The investigation in Chang et al. [1988] points out the connection of one-way iterative automata to complexity theory. (See Section 2.5 for formal language properties of this class of CA.) A related class of automata motivated by design of systolic systems and algorithms is the class of systolic trellis automata, which have been quite extensively studied by Choffrut and Culik II [1984]; and Culik II et al. [1984]. This class is equivalent to bounded space real-time one-way CA. Study of systolic arrays modeled as 1-d, 2-d, one-way CA, and iterative arrays were carried out by Ibarra and Kim [1984] and Ibarra et al. [1985b]. This work has resulted in the development of many easy-to-implement systolic algorithms. One-way CA on Cayley graphs have also been studied [Roka 1994].

2.3 Biological Connection

2.3.1 *L-systems*. CA were originally proposed by von Neumann to provide a formal framework for the study of "complicated" natural systems. Later work in this direction used a structure called dynamic CA for modeling of biological systems. One of the early attempts was by Lindenmayer [1968], who proposed a model of growth for filamentary organisms based on ideas of CA. The class of CA used is called dynamic CA where cells may appear or disappear with time. The key idea is to consider a sequence (1-d array) of cells of the organisms. Then cell division is modeled by allowing a cell to be replaced by more than one cell, each in some prespecified state. If a cell has a neighborhood consisting of its left neighbor, then after division the same neighborhood holds. The model just described is for nonbranching filamentary organisms. It is also possible to extend the theory to model branching organisms along with a neighborhood consisting of both left and right neighbors. For the model with both left and right neighbors, two output functions are defined, the left and the right output. The input to the left cell in the next step is the left output and similarly for the right cell. Thus we may consider the cell to consist of three components (a technique which has also been used very successfully in other areas of CA). For the branching organism, the local rule specifies the first cell of the branch to be created. So if the local neighborhood of a cell is conducive, a new branch is created, which is then considered attached to the basal cell. A cell may give rise to several branches, and in the model it is not possible to distinguish between the relative orientation of the branches. However, it is also possible for the branches to give rise to new branches, and so on. Use of L-systems in modeling plant life is discussed in detail [Prusinkiewicz and Lindenmayer 1990]. Later work on such systems was mainly formal language-theoretic (see Kari [1997]).

2.3.2 *Self-Reproduction and Artificial Life*. The first attempt at modeling artificial life with CA was von Neumann's self-reproducing automata. An implementation of this construction was done [Pesavento 1995]. Langton [1984] argued that computation universality is not a fundamental requirement for a self-reproducing automata.

An interesting biological connection was studied by Holland [1976]. He used CA as a model to study the spontaneous emergence of self-replicating systems. The CA is used as a model of the universe (called the α-universe) where each cell has two parts. The first part stores the state of the cell and the second part indicates the nature of the bond (strong or weak) the cell has with its left or right neighbors. Stochastic operators are used to manipulate the states in accordance with the bonds and in a conservative manner—elements are never created or destroyed, they are only moved about and rearranged by the operators. The operators are themselves encoded by the states of the cells. The crucial parameter studied is the expected time until the emergence of self-replicating systems, which is an arrangement of the universe which can replicate itself.

For other work on modeling of artificial life using CA, see Ikegami and Hashimoto [1995] and Adami [1994]. A great amount of work has been done using CA for modeling biological systems. One can see current issues of the *Journal of Theoretical Biology* for recent work in this area.

2.4 Fault-Tolerant Computing

The idea of fault-tolerant computing also originates from von Neumann [1963b], who showed how to build a reliable Boolean circuit out of unreliable components. For the case of CA, the unreliable components are taken to be the cells. Each cell can misoperate and assume an incorrect state, i.e., one not dictated by the local rule. Early work in this area assumed a fault model called k-separated misoperation [Nishio and Kobuchi 1975], i.e., there exists a finite set K of Z^d such that given a cell $x \in Z^d$ at most one cell in the set $x + K$ will misoperate (here d is the dimension of the grid, and Z is the set of integers). In Nishio and Kobuchi [1975], it is shown how to construct a CA which will correctly simulate an unreliable CA with k separated misoperation, step for step. The basic idea is to encode the initial configuration of the unreliable automaton suitably to form the initial configuration of the simulating automaton. The coding is carefully designed so that each cell in the coded configuration can use a majority voting rule to decide its state. The local rule of the simulating automaton is almost the same as the original one, except that at each step each cell of the simulating automaton corrects any error in its neighboring cells before applying the local rule. This leads to an increase in the neighborhood size. It has been shown that under the same fault model, unreliable CA over group graphs can also be simulated in an error-free way [Harao and Noguchi 1975].

Gacs [1986] has shown how to construct a 1-d CA which can reliably perform arbitrarily large computations, and where each cell can perform an error with a positive probability. The fault model so considered is important from an ergodic theory point of view, and Gac's result leads to the refutation of the "positive probability conjecture" in statistical physics, which states that any one dimensional infinite particle system with positive transition probabilities is ergodic. For recent work on reliable cellular automata, see Gacs [1997].

2.5 Language and Pattern Recognition

A finite CA can be thought of as a language acceptor by considering the initial configuration to be the input string and acceptance or rejection is determined by a specific cell (say the rightmost) going to an accept or reject state. For a 2-d CA, the problem is one of pattern recognition and the accept cell can be the northeast one in a rectangular grid or it could be the easternmost cell in the northernmost row for a general 2-d layout. It turns out that the linear, Dyck and bracketed context-free languages can be accepted by CA (also by one-way CA) in real time [Smith III 1972; Dyer 1980; Ibarra et al. 1985a]. In

Smith III [1972], it is shown that nondeterministic-bounded (the input is delimited and all other cells are in the quiescent state and remain so during the computation) CA can recognize the CFLs in real time. The deterministic case is open. If the number of steps of a computation is fixed (but language-dependent), then the set of languages accepted by nondeterministic 1-d CA is the set of regular languages [Sommerhalder and Westrhenen 1983]. Here acceptance is defined by all nonquiescent cells entering some final state. This notion of acceptance allows languages to be accepted is less than real time. In Ibarra et al. [1985a] it is shown that there are noncontext-free languages recognizable in $O(\log n)$ time, and that the languages accepted in $o(\log n)$ time are regular.

Certain language classes can be defined by both restricting and enhancing the power of CA. This is done by introducing the following four conditions:

(1) one-way communication giving rise to *oneway CA*;

(2) for an input of n symbols, the number of steps of computation required is exactly n; this is called *real-time* computation;

(3) for an input of n symbols, the number of steps of computation is proportional to n; called *linear-time* computation;

(4) the local rule is *nondeterministic*, giving rise to nondeterministic CA.

The symbols O, r, l, and N are used as prefixes to the word CA to denote a particular language class. As an example, rOCA denotes the class of languages accepted by real-time one-way CA. The CA is taken to be bounded, so that all computations take place within the n cells of the initial configuration of length n. The relationships among CA language classes, as well as their relationship to the classical language classes, were extensively studied. See Mahajan [1992] for a good survey of results and techniques in this area. Here we briefly mention several important results. The first (and easy) result is that the language class CA is equal to DSPACE(n). The class lCA is a subset of OCA [Chang 1988; Ibarra and Jiang 1987; Ibarra et al. 1985b]. This is obtained by considering the relationship of both OCA and lCA to sweeping automata [Chang et al. 1988]. It is also known that rOCA is a proper subset of rCA [Choffrut and Culik II 1984; Culik II et al. 1984], and rCA is equal to lOCA [Choffrut and Culik II 1984]. The PSPACE-complete language QBF (quantified Boolean formulas) belongs to OCA [Ibarra and Jiang 1987] and NSPACE($\sqrt{n}$) and ATIME(n) are subsets of OCA. The class OCA lies between NSPACE($\sqrt{n}$) and CA=DSPACE (n), and proper containment between OCA and CA would separate these two classes, improving Savitch's result. It is also conjectured that lCA is properly contained in OCA, since lCA is a subset of P and OCA contains QBF, any proof that lCA=OCA will imply that P=PSPACE, a rather unlikely result. For the nondeterministic language classes, it has been proven in Dyer [1980] that NOCA=NCA=NSPACE(n), the class of context-sensitive languages. Further, it is known that rNOCA contains an NP-complete problem [Ibarra and Kim 1984]. Open problems and additional examples of languages contained by rOCA, rCA, lCA, and OCA can be found in Mahajan [1992].

2.6 Invertibility, Surjectivity and Garden of Eden

A major focus of research in CA is related to questions of invertibility. A CA rule ρ is called invertible if there exists another rule ρ^{-1}, called the inverse rule, which drives the CA backward, i.e., if application of ρ to a configuration c produces a configuration d, then application of ρ^{-1} to d produces c. A CA is called invertible if its local rule is invertible. Richardson [1972] proved that a CA is invertible iff its global map is injective. The technique does not provide an inverse, as topological arguments are used to prove the result. For an automata-theoretic approach to the problem, see Culik II [1987]. Amoroso and Patt proved that there is an effective procedure to determine invertibility of 1-d CA, based on the local rule [Amoroso and Patt 1972]. Kari [1990; 1994] has shown that for a 2-d CA the question of determining invertibility from the local rule is undecidable. The reduction is from the tiling problem in conjunction with a special version of the tiling problem, called the directed tiling problem.

The surjectivity of the global map of a CA have also been studied. A configuration is called a "Garden of Eden" configuration if it is not "reachable," i.e., it can only occur as an initial configuration in any evolution. Existence of such a configuration shows that the global map is not surjective. Myhill [1963] proved that a global map is surjective iff its restriction to finite configurations is injective. The surjectivity of 1-d CA is decidable [Amoroso and Patt 1972]. Kari [1990; 1994] proves that the problem is undecidable for two dimensions by showing that the injectivity problem restricted to finite configurations is undecidable. To tackle finite configurations, Kari [1990; 1994] introduced a special class of tilings with the "finite tiling property".

For linear CA over Z_m, Ito et al. [1983] provide necessary and sufficient conditions for invertibility. Computation of the inverse of a CA, even when it is invertible, can be a difficult job. Sato [1994] provides a construction for a special class of CA, called the group-structured CA. Manzini and Margara [1998] provide an efficiently computable formula for the inverse of a d-dimensional linear CA over Z_m. There is a quadratic-time algorithm to determine reversibility and surjectivity of the global map of a linear CA [Sutner 1991]. The algorithm is based on the representation of a configuration of a linear CA by a finite graph (a De Bruijn graph) as used by Wolfram [1984a].

Given a 1-d CA, it is possible to construct an invertible 1-d CA which can simulate the original CA [Morita 1995]. It is even possible to simulate TM by invertible CA [Dubacq 1995]. Toffoli [1977] has shown how to simulate any k-d CA by an invertible $(k + 1)$-d CA. This proves the computation universality of invertible CA for dimensions higher than one; and from the result of Morita [1995], 1-d invertible CA is also capable of universal computation. However, the question of whether a k-d CA can be simulated by a k-d invertible CA is still open for $k > 1$. The invertibility question is of fundamental importance to physics, as it can be used for modeling microscopically reversible dynamical systems; see Toffoli and Margolus [1990] for a survey.

For a finite CA, an injective global map has to be bijective. Moreover, if the global map of a finite CA is injective, it does not necessarily mean that there is an inverse CA, in the sense that there is a inverse local rule that can be used to force a configuration to retrace the original evolution. So a finite CA is said to be invertible if the global map is a bijection. In this case, it is trivial to see that the nonexistence of Garden of Eden configuration is a necessary and sufficient condition for invertibility of the global map. It is, in general, difficult to determine invertibility of finite CA; see Harao and Noguchi [1978] for a discussion of the dynamics of finite CA. If the global map is a linear transformation, then the problem becomes more manageable. Extensive discussion on properties of linear or additive CA can be found in Martin et al. [1984]. In fact, for 1-d linear CA, the question is easy to answer [Martin et al. 1984; Sutner 1990b; Barua and Ramakrishnan 1996]. Extensions to 2-d CA are studied in Barua and Ramakrishnan [1996]; Sutner

[1996]; and Sarkar [1996] and multi-d CA in Sarkar and Barua [1998a].

3. CA GAMES

3.1 Firing Squad Problem

This is basically a synchronization problem, but can also be thought of as a game. The problem was first proposed by Minsky around 1957, and first appeared in print in Moore [1964]. The following is a simple description of the problem. Consider n soldiers (out of which one is a general) standing in a row. The soldiers (including the general) can communicate only with their immediate left and right neighbors. The general gives the command to fire. Ultimately, the soldiers and the general are required to fire simultaneously, and for the first time. In CA terms, the problem is to design a cell and a local rule such that starting from an initial configuration, where only one cell is on and the other $n - 1$ cells are off, there is an evolution such that all the cells enter a predesignated state all at once and for the first time. Note that the problem can also be considered on an infinite 1-d array, but then the other cells must all be in the quiescent state and remain so throughout. The basic problem is to design a cell which is independent of the number of soldiers, and hence will work for an array of arbitrary length. This means that none of the cells can count upto n. In case the general is one of the end cells, it is easy to see that the minimum time required for synchronization is $2n - 2$ steps. Waksman [1966] provides a solution in $2n - 2$ steps. The solution depends heavily on the idea of signals propagating through the array at different speeds. A signal is essentially a symbol which passes from one cell to its neighbor in a particular direction (left or right). A signal propagates at the "speed of light" if it moves one cell at each step. This is the fastest speed at which a signal can propagate through the array. It is possible for a cell to suppress a signal for a fixed number of time steps. Then the speed of the signal determines its geometry—the angle that it makes with the horizontal. A minimum state solution to the problem is provided in Mazoyer [1987].

For a solution to the problem where the general can be any cell, see Moore and Langdon [1968]. Culik II [1989] considered several other variation, and has used the results to disprove a conjecture of Ibarra and Jiang that real-time one-way CA cannot accept certain languages. The problem has also been generalized to higher dimensions [Nguyen and Hamacher 1974; Shinahr 1974] and node static and dynamic CA [Herman et al. 1974; Varshavsky et al. 1970]. A generalization to arbitrary graphs called the Firing Mob problem was introduced in Culik II and Dube [1991], where an efficient solution is also provided. The introduction to Culik II and Dube [1991] also contains a brief history of the Firing Squad problem and also the solutions attempted by various researchers. The central result that it is possible to design such a CA is called the Firing Squad theorem, and used in language and pattern-recognition studies of CA [Smith III 1972; Culik II 1989]. A related "desynchronization" problem is to design a CA such that all cells are initially in the same state, and ultimately only one cell goes to a predesignated state. It is called the "Queen Bee" problem [Smith III 1976].

3.2 Game of Life

This game was originally proposed by Conway and made popular through Martin Gardner's column in the *Scientific American* [Gardner 1970; 1971]. The original motivation was to design a simple set of rules to study the macroscopic behavior of a population. The criterion for choosing the rules was based on the principle that the growth or decay of the population should not be easily predictable. After a great deal of experimentation, Conway chose the following set up. The population is represented by a configuration of a 2-d infinite array of cells with Moore (unit square) neighborhood, where each cell can be in one of the states 1 or 0. The local rule is described as follows:

(1) *Survival*: If a cell is in state 1 (alive) and has 2 or 3 neighbors in state 1, then the cell survives, i.e., remains in state 1.

(2) *Birth*: If a cell is in state 0 and has exactly 3 neighbors in state 1, then in the next time step the cell goes to state 1.

(3) *Deaths*: A cell in state 1 dies (goes to state 0) of loneliness if it has 0 or 1 neighbors. Also, it dies because of overcrowding if it has 4 or more neighbors.

Each configuration is called a population, and the evolution of the population is studied. As with many CA evolutions, the "Game of Life" shows fantastic variation in the growth patterns of the initial population. A group at MIT has shown that there is a simple initial configuration that grows without limit. The configuration grows into a "glider gun" and, after 40 steps, fires the first "glider," and thereafter continues firing gliders after every 30 moves. It has been informally proved that the "Game of Life" is capable of universal computation. For a good account of the game and for some good pictures, see Gardner [1970; 1971] and Conway et al. [1992]. There are also several Internet pages dedicated to the "Game of Life."

3.3 $\sigma(\sigma^+)$ -Game

This game was first proposed by Sutner [1990a] and is based on the battery-operated toy MERLIN [Pelletier 1987]. It is a two-person game and is played on a 2-d finite grid, where each node has a bulb that can be either on or off. A move is made by choosing a node and, as a result, the states of all the bulbs in orthogonal neighborhood positions toggle. A configuration of the game is a state of the grid where some of the bulbs are on and others are off. Player A chooses two configurations, the initial and the target configurations. Player B has to make a sequence of moves starting from the initial configuration and reach the target configuration. It is easy to see that choosing a node twice is the same as not choosing it at all. The order of the choice of nodes is not important. Thus, any winning strategy (solution) for B can be viewed as a set rather than a sequence. This set of nodes can then be thought of as a configuration of the grid (the bulbs in the set are on, the others are off). Suppose the initial configuration is the all 0 configuration and the target configuration is X_t. If Z is a solution to this instance, then $\sigma(Z) = X_t$, where σ is the global rule of a finite 2-d CA whose local rule is the sum (modulo 2) of the four orthogonal neighbors. Again, Z is a solution for the pair (X_s, X_t) iff $\sigma(Z) = X_s + X_t$ and hence the number of solutions (if any exist) is 2^k, where k is the corank of the linear map σ. Thus, the study of the σ-game reduces to the study of linear 2-d CA [Barua and Ramakrishnan 1996; Sutner 1990b]. The corresponding game where the state of the chosen bulb also changes is called the σ^+-game. Both the σ and σ^+-games have been studied on 2-d and multidimensional grid; see Sarkar and Barua [1998a] for results on multidimensional CA, and with direct relevance to the multidimensional $\sigma(\sigma^+)$-game. The game has also been considered over arbitrary graphs [Sutner 1988b; 1989b], but results are more difficult to obtain in this setting.

4. MODERN RESEARCH

4.1 Empirical Study

The mid-1980s are an important period in the history of CA, largely due to the work carried out by Wolfram. The nature of his questions represent a paradigm shift in CA research. Wolfram carried out an extensive experimental analysis of the growth patterns of CA. An early paper by Wolfram [1983] discusses several statistical parameters of the space-time patterns of CA evolution. Later work extended and clarified much of the intuition in several directions. An excellent source of papers on this period of CA research is a book by Wolfram [1986].

The approach taken is to consider CA as models of complex systems, in the sense that very simple CA rules can give rise to extremely complicated patterns. The mathematical simplicity in CA description is thought to be a significant advantage for modeling, rather than using systems of differential equations. A related phenomenon in CA evolution is self-organization. Starting from random unordered configurations with maximum entropy, a CA will evolve to states of lesser entropy. This is contrary to the second law of thermodynamics, which states that reversible systems evolve to states of maximal entropy. The microscopic irreversibility of CA is the reason behind this self-organizing behavior.

The 1-d, 3 neighborhood, binary CA is the one extensively studied by Wolfram. A numbering system for the possible local rules of such a CA can be found in Wolfram [1986]. Rules 90 and 150 are important. Rule 90 is the sum modulo 2 of the states of the nearest two neighbors. Rule 150 is the sum modulo 2 of the states of the nearest two neighbors and the state of the cell itself. Note that both rules 90 and 150 are linear.

The approach taken by Wolfram [1984b] in studying the growth patterns of CA is to define several local and global statistical parameters and to study their behaviors. Some important local parameters are

—average density of nonzero sites, which is a "rough" measure of the growth of CA evolution;

—the average number of triangles or triangle density $T(n)$ of triangles of base length n, in the space time pattern (see Figure 3);

—sequence density $Q_i(n)$ is the density of sequences of exactly n adjacent sites with the same value i.

Both the triangle and the sequence density follow an exponential rule for evolution from an initial disordered state. For example, for large n, $T(n) \sim l^{-n}$ and the parameter l distinguish between linear ($l \approx 2$) and nonlinear ($l \approx 4/3$) rules. Another important feature of the space time evolution from an initial disordered state is that triangles of all sizes are obtained, and hence the structure is generated on all scales.

For a finite N-cell CA, one can consider the finite set of 2^N configurations to be an ensemble where each configuration has equal probability of occurrence. After evolution for a few time steps, an equilibrium is achieved where the configurations have different probabilities according to some distribution function. On taking the average over the ensemble, properties of configurations with higher probability dominate. This indicates the self-organizing character of CA evolution. Another measure of self-organization is entropy. For a finite CA, the entropy is defined as $\Sigma p_i \log p_i$, where p_i is the probability of configuration i. For irreversible CA, this entropy decreases from an initial maximum (for random initial configurations) to lesser values. A corresponding entropy called "block" or "Renyi" entropy can be defined for infinite 1-d CA, and shows a similar phenomenon. For second-order (next state depends on present and previous states of neighbors) reversible infinite CA, the entropy almost always increases with time.

Another interesting approach to characterize CA evolution comes from formal language theory. It was shown in Wolfram [1984a] that the set of configurations that can appear after t time steps forms a regular language. The size of the minimal DFA after t steps provides an indication of the complexity of the set of configurations after t steps.

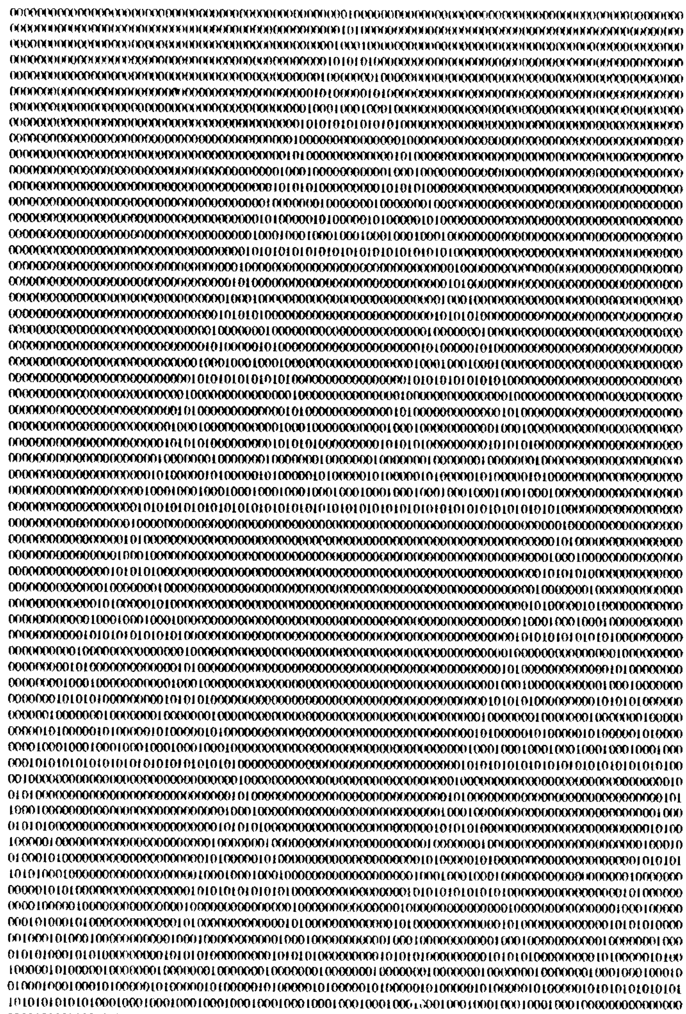

Figure 3. Triangles in the space time pattern of a CA. The pattern shows a self-similar structure.

For many CA rules, the minimal DFA becomes more complicated at each step and the sequence of DFAs does not appear to exhibit any overall structure. Again, for some CA rules, the infinite limit set of configurations (the set of configurations reachable at arbitrarily large time steps) is also a regular language; but there are others whose regular language complexity grows with time, and hence seem to generate nonregular language in the limit. In fact, Hurd [1987] provided examples of CA with strictly nonregular, noncontext-free and non-r.e. limit sets. In Green [1987], a CA is described whose limit set is NP-hard.

A modification of this approach associates a weight, corresponding to the probability P_i that each node is visited, to each node of the minimal DFA. One then computes the entropy measure $\Sigma P_i \log P_i$ and uses it to study the growth pattern of the configurations for details, see the Appendix (Table 11) of Wolfram [1986].

4.2 Classification of CA

A major problem stemming from Wolfram's work is classifying CA rules according to their behavior. The initial empirical classification was proposed by Wolfram himself [Wolfram 1984b]. His classification is based on entropy measures and identifies the following four classes.

(1) Evolution leads to a homogeneous state.

(2) Evolution leads to a set of separated simple stable or periodic structures.

(3) Evolution leads to a chaotic pattern.

(4) Evolution leads to complex localized structures which are sometimes long-lived. It is believed that this class is capable of universal computation.

Later work concentrated on formalizing the intuitive classifications by Wolfram et al. Culik II and Yu [1988] proposed the following classification. Let ρ be the local rule for a CA. Then

(1) Rule ρ is in class one iff every finite configuration, i.e. configurations in which only a finite number of cells are in nonquiescent states, evolves to a stable configuration in finitely many steps.

(2) Rule ρ is in class two iff every finite configuration evolves to a periodic configuration in finite number of steps.

(3) Rule ρ is in class three iff it is decidable whether a configuration occurs in the orbit of another.

(4) Class four comprises all local rules.

They show that the problems of deciding membership of a rule ρ in classes one and two are Π_1^0-hard. Similarly, class three is Σ_1^0-hard. Sutner [1989c] has shown that classes one and two are Π_2^0-complete and class three is Σ_3^0-complete. The arguments are based on encoding TM instantaneous descriptions by natural numbers and the simulation of TM by CA. It is important to note that the above classification considers only finite configurations. Infinite configurations in general cannot be finitely described, and hence cannot be tackled by conventional computability theory. A classification of periodic boundary condition CA (whose configurations can be thought of as spatially periodic configurations of an infinite CA) have also been proposed [Sutner 1990a]. Using a nonstandard simulation of a TM by a CA, it is shown that the problem of deciding membership in the hierarchy is undecidable.

In a recent study, Braga et al. [1995] provided a classification of CA based on pattern growth. The pattern growth properties are shown to be dependent on the truth table of the local rule of the corresponding CA. This provides an algorithm for classifying CA rules, and hence defines an effective hierarchy of CA rules, in sharp contrast to the undecidability results discussed above. The essential technique is the fact that certain shift-like dynamics in the evolution can be discovered by looking at the truth table of the local rule. Then a proper grouping of rules exhibiting similar dynamics yields a classification which is close to that of Wolfram; see Culik II et al. [1990] for other approaches in classifying CA.

A preliminary study of 2-d CA [Packard and Wolfram 1985] shows that it is possible to classify 2-d CA along the same lines as 1-d CA. This suggests that the global behavior of 2-d CA is similar to 1-d CA. However, 1-d and 2-d CA show marked differences with respect to other properties. Golze [1976] has shown that for 1-d CA every recursive configuration (a configuration where each cell value can be calculated effectively) has a recursive predecessor; but in the 2-d case, even a finite configuration may fail to have a recursive predecessor. Again, invertibility of 1-d CA is decidable, while it fails to be so for 2-d (and higher dimension) CA.

4.3 Limit Sets and Fractal Properties

One important direction of CA research in the modern era is the study of the limit sets of CA space-time patterns. Early work in this area was done by Willson [1978; 1981], and the topic received an impetus from Wolfram [1983; 1984b]. However, the notion of a limiting set of configurations obtained by evolving a CA was introduced by Podkolzin in 1976 (see Culik II et al. [1990]). Later, we will mention some of the work in this area.

4.3.1 *Fractal Dimension of Space-Time Patterns*. The space time-pattern observed during simulation shows several kinds of interesting characteristics (see Appendix of Wolfram [1986]). One of the important features is a scale invariance and self-similarity on different scales. This immediately suggests computing the fractal dimension of such patterns. Wolfram's empirical investigation [Wolfram 1983] outlines two natural ways to do this. In the first approach, a parameter $T(n)$ is defined that measures the density of triangles of base length n. A geometrical construction shows that for rule 90, $T(n) \sim n^{-1.59}$ and for rule 150, $T(n) \sim n^{-1.69}$. The invariants 1.59 and 1.69 then give the limiting fractional dimension of the patterns. In the second approach, the space-time configurations are scaled to fit the same perimeter, and one considers the set of all limit points. This gives rise to a fractal dimension which is a "geometric" dimension, and is also called the Kolmogoroff dimension. Willson [1984b] investigates theoretically why the two approaches to compute dimension should coincide and provides examples where the Kolmogoroff dimension differs from the more usual Hausdorff-Besicovitch dimension.

Theoretical study of the limit sets of CA evolution via geometric invariants were performed by Willson [1984a]. The basic object of study is the sequence

$$\omega, F\omega, F^2\omega, \ldots, F^p\omega, \ldots,$$

where ω is a configuration of an n-dimensional CA, and F is the global rule of some CA. If we fix a state q, then we can think of the set of cells (in space-time configuration) having value q as a set of points where each point is given by an $(n + 1)$-dimensional vector. Let X_p be the above set corresponding to $F^p\omega$. Consider the set $X_{p/p}$, where the vectors of $X_{p/p}$ are obtained by dividing each vector of X_p by p. This scaling ensures that the space-time configurations fit the same perimeter at each time step. Let $Lim(\omega, q)$ be the set of points in the limit $p \to \infty$. This limit is taken as an approximation of X_p and properties of the limit indicate the nature of the growth pattern in space-time configurations. For example, if $Lim(\omega, 1)$ is a tetrahedron, then we expect the configurations to grow into a tetrahedral form. When the CA rule is linear (mod 2), it has been shown that the limit set is a compact subspace of Euclidean space and can have fractional Hausdorff dimension. For linear CA, this provides a formal proof of Wolfram's basic intuition. Space-time patterns of arbitrary linear CA have also been studied [Takahashi 1992]. The corresponding limit sets are generally fractals. The self-similar structure is characterized by a transition matrix, whose maximum eigenvalue determines its Hausdorff dimension.

4.3.2 *Limit Sets of CA Evolution*. Limit sets were also studied from a different direction using formal language-theoretic methods [Hurd 1987; Culik II et al. 1989]. In this approach, the set of configurations rather than the space-time patterns are considered. For a d-dimensional infinite CA having S as the set of cell states, the set of configurations is S^{Z^d}. When S is endowed with the discrete topology, then S^{Z^d} with the product topology is compact by Tychonoff's theorem, and the global map G of the CA is a continuous function. Letting $S^{Z^d} = \Omega_0$ and $\Omega_i = G(\Omega_{i-1})$ for $i \geq 1$, each Ω_i is a compact subspace of S^{Z^d} and $\Omega = \cap_{i\geq 0}\Omega_i$ is the limit set for the CA. This Ω is the object of study. It was shown in Culik II [1989] that, for $d \geq 2$, it is undecidable whether Ω contains a finite configuration. Using the notion of a limit set of a CA, it is possible to define a limit language as follows. Consider a 1-d CA, then every configuration is a bi-infinite word over S. For a configuration c, define

$$L[c] = \{w \in S^* : w \text{ is a finite subword of } c\}$$

and let $L[C] = \cup_{c\in C} L[c]$ for a set of configurations C, then $L[\Omega]$ is the limit language. The membership problem for such a limit language is undecidable [Culik II et al. 1990]. For a survey of results regarding this limit language, see Culik II et al. [1990]. Given a CA, the complement of the limit language is r.e. [Culik II et al. 1990]. Also, for any language whose complement is r.e., one can construct a CA whose limit language yields the chosen language after intersection with a regular language and a ϵ-limited homomorphism. This can be used to show that there exists a CA whose limit language is not r.e. Similar properties have been obtained for Π, the closure of the points periodic under the global CA map; see Culik II et al. [1990] for details.

4.4 Dynamics of CA

4.4.1 *State Transition Diagram*. One can define a State Transition Diagram (STD) for an infinite CA by considering an infinite directed graph whose vertices are the configurations of the CA and whose edges represent one-step evolution of the CA. This was done by Podkolzin (see Culik II et al. [1990]), where it is shown that the STD either has a single connected component or has uncountably many connected components. If a CA has only one single connected component, it is called a nilpotent. It has been proved (see Culik II et al. [1990]) that, for two or more dimensions, the problem of CA nilpotency is undecidable. The same result was proved by Kari [1992] for one dimension. Podkolzin has also shown that, for any CA, either the limit set is a singleton and the CA is nilpotent, or the limit set contains an infinite number of elements; see Culik II et al. [1990] for further discussion on limit sets.

4.4.2 *Symbolic Dynamics*. Another interesting approach to the study of dynamic properties of CA is to consider the CA as a computational device acting on bi-infinite strings, on one hand, and as a continuous function on a compact metric space on the other. This gives rise to considerations of symbolic dynamics on bi-infinite strings. If S is the state set for a cell of a 1-d CA and Z is the set of integers, then S^Z is the set of all configurations of the CA. It should be noted that if G is a global CA map, then it is a shift-invariant continuous map from S^Z to S^Z. The converse that any shift-invariant continuous map from S^Z to S^Z arises as a CA map was proved by Hedlund [1969]. A topologically closed subset of S^Z is called a subshift if it is invariant under the shift map. A subshift is said to be of finite type if no bi-infinite word in it contains any block from an excluded finite set. A sofic system is the image of a shift-invariant continuous map acting on a subshift of finite type. It has been shown that each sofic system is a $\omega\omega$-regular set, and for each $i \geq 0$, $G^i(S^Z)$ is a $\omega\omega$-regular set [Culik II and Yu 1991] where G is the global map of a CA; see Culik II and Yu [1991] and Culik II et al. [1990] for a more detailed discussion.

4.4.3 *Topological Properties*. A topic closely related to limit sets, which has received a lot of attention in recent times, is the topological properties of CA. These properties arise when a CA is considered to be a discrete time dynamical system. As mentioned before, the set of d-dimensional CA configurations with global map F can be considered equipped with the product topology when the state set of a cell is given a discrete topology. An element of a subbasis for this topology is a set of configurations such that a particular cell of all the configurations in the set are in a fixed state. It is possible to define several distance measures on the set of configurations, all of which induce the product topology (see Finelli et al. [1998] for an example).

One can define several dynamical properties [Manzini and Margara 1999; Finelli et al. 1998; Hurd et al. 1992] such as

(1) *Topological transitivity*: For each pair of nonempty open subsets U, $V \subseteq X$, there exists $n \geq 0$ such that $F^n(U) \cap V \neq \phi$.

(2) *Sensitivity to an initial condition*: If there exists $\delta > 0$ such that for all configurations x and for all $\epsilon > 0$ there exists a configuration y and an $n \geq 0$ such that $d(x, y) < \epsilon$ and $d(F^n(x), F^n(y)) > \delta$.

(3) *Attractor*: A nonempty subset Z of configurations is an attractor for F iff there exists an open set U of configurations such that $F(\bar{U} \subseteq U)$ and $Z = \cap_{j \geq 0} F^j(U)$.

(4) *Expansivity*: If there exists $\delta > 0$ such that for every pair of distinct configurations x, y there exists an integer n such that $d(F^n(x), F^n(y)) > \delta$. Since n can vary over the set of integers, this definition makes sense only if the associated CA is invertible. For noninvertible CA, this definition can be modified by restricting n to be nonnegative. In this case, the CA is said to possess positive expansivity.

(5) *Topological entropy*: Informally this measures the uncertainty of the forward evolution of any dynamical system in the presence of an incomplete description of an initial configuration. A definition tailored to 1-d CA is provided in Hurd et al. [1992].

(6) *Lyapunov exponents*: This is usually defined over differentiable spaces. An adaptation of this concept for the topology on CA configurations is provided in Shereshevsky [1992].

Many interesting results have been obtained for these and other topological properties. Hurd et al. [1992] have shown that the topological entropy of CA is uncomputable. However, for linear and positively expansive CA, this can be computed as shown in Michele et al. [1998]. Attractors of CA are studied in Blanchard et al. [1997] and in Kurka [1997]; and linear CA in Manzini and Margara [1999]. Complete characterizations of most topological parameters for linear CA have also been done (see Manzini and Margara [1999] for a list of such properties). The relationship of Lyapunov exponents to expansivity and sensitivity were studied in Finelli et al. [1998]. A classification of CA into five disjoint classes based on the structure of their attractors was made by Kurka [1997].

4.5 Computational Complexity

An early task in the study of CA's computational complexity is learning the minimum number of steps required to perform certain computations. Serious attempts at studying complexity-theoretic questions regarding CA is a later development. Wolfram [1984a] shows how to construct a graph to represent configurations reachable after one time step of a 1-d CA. All possible infinite paths through the graph represent all possible configurations. The notion can be generalized to a finite number of time steps and also to limit sets. The graph can be regarded as the state transition graph of a finite automaton that may be nondeterministic. The equivalent minimum state DFA can be constructed, and the number of states in such a DFA provides a measure of the complexity of the corresponding configuration set. For some interesting properties of this measure, see Wolfram [1986]. A consequence of Wolfram's result is that the predecessor existence problem (i.e., given configuration X, does there exist a configuration Y such that Y evolves to X in one time step?) for 1-d CA is decidable.

This leads to a more formal study of the computational complexity of CA. In particular, it was important to find NP-complete problems for CA. First results appeared in Green [1987], where a CA is constructed for which the following problems are NP-complete:

—determining if a given subconfiguration s can be generated after $|s|$ time steps;

—determining if a given subconfiguration s will recur after $|s|$ time steps;

—determining if a given temporal sequence (values of a particular cell taken over time) of states s can be generated in $|s|$ time steps.

The particular CA described is quite complicated, since an arbitrary structure of the 3-SAT problem has to be encoded in the essentially local communication mechanism of a CA. For an infinite CA, certain problems [Sutner 1989a] such as configuration reachability (CREP, source configuration X; target configuration Y; is Y reachable from X?), and predecessor existence (PEP) are undecidable. Undecidability of CREP is easy to see, since a CA can simulate a TM, and configurations of the CA encode instantaneous descriptions of TM. Hence the halting problem for TM can be translated to CREP by asking whether a halting configuration is reachable from the initial configuration. In fact, CREP is Σ_1^0-complete for infinite CA of any dimension. However, for PEP there is a marked difference for the 1-d and higher dimensional CA. From Wolfram's characterization of 1-d CA using regular grammars [Wolfram 1984a], it follows that PEP is decidable. On the other hand, Yaku [1973] has shown that for 2-d CA restricted to finite configurations, PEP is equivalent to the problem of whether a TM halts on the empty tape, and hence is Σ_1^0-complete.

Similar results for finite CA are studied in Sutner [1995]. For 1-d CA, PEP is NLOG-complete, and is NP-complete for all dimensions higher than one. In Sutner [1995], examples of local rules are constructed such that CREP is PSPACE-complete/NP-complete for 1-d CA. For 1-d CA, if one restricts attention to a polynomially bounded version of CREP (i.e., the number of steps is less than or equal to some polynomial in the number of cells), it is possible to construct a local rule such that CREP is P-complete (w.r.t. log space reductions). For 2-d CA, an example of rule r is provided such that CREP is NP-complete. A classification of CA rules similar to that of Culik and Yu (for infinite CA) is connected to several deep problems in complexity theory.

Durand [1994; 1995] provides complexity results for CA with a different flavor. The injectivity problem for 2-d CA restricted to finite configurations and von Neumann neighborhoods is co-NP complete [Durand 1994]. This result is about arbitrary CA and is different from the above results where examples of CA are provided for which a problem is complete for some complexity class. Hence this kind of result may be called uniform complexity results. Durand [1995] also proves that the reversibility problem for 2-d CA restricted to certain types of finite configurations is complete for the class RNP introduced by Levin.

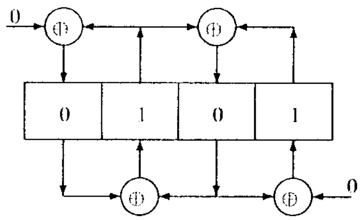

Figure 4. A 4-cell null boundary CA.

4.6 Finite CA and its Applications

A finite CA has a finite number of cells. Figure 4 shows a 4-cell CA whose local rule is the sum modulo 2 of the states of the left and right neighbors. For finite CA, the dynamical properties are completely captured by the State Transition Diagram (STD), which is a directed graph whose nodes are configurations of the CA, and there is a edge from node i to node j iff configuration i leads to configuration j in one time step. Since a finite CA is an autonomous deterministic machine, it is easy to see that the STD will consist of components, with each component having a unique cycle and trees of height ≥ 0 rooted on the cycle vertices. The cycles capture the steady state behavior of the system and are sometimes called attractors, while a branch in a tree captures the initial transient behavior. One can ask several important questions regarding the dynamical parameters of the system; the number of cycles, length of the cycles, height of the trees, branching degree of each node, etc. For an arbitrary CA such questions are very difficult to answer. For CA with a periodic boundary condition, some results for reversibility and maximal cycle length are presented in Harao and Noguchi [1978]; see Lee and Kawahara [1996] for recursive formulas describing STD of finite CA. However, a complete characterization is not known and generalization to higher dimensions is difficult.

4.6.1 *Linear CA*. For finite linear CA, much more information can be obtained using algebraic methods. The STD in this case shows a more uniform behavior [Martin et al. 1984]; the trees rooted on any cycle vertex are isomorphic to the tree rooted on the null configuration, the indegrees of all the nodes are equal, and equal to the dimension of the kernel of the linear map, etc. For a wealth of results on the STD of 1-d periodic boundary CA, see Martin [1984]. Additional results can be found in Guan and He [1986]. For 2-d CA, Kawahara et al. [1995] investigate when the configuration reachable in one time step from the all ones configurations lies on a cycle. The dimension of kernel of 2-d linear CA were studied by several authors [Barua and Ramakrishnan 1996; Sutner 1988b; 1990b; Sarkar 1996], and is related to the σ-game mentioned before. For multidimensional CA, it is difficult to obtain a characterization of the dimension of the kernel, but a characterization of reversibility is presented in Sarkar and Barua [1998a].

An important problem in the algebraic analysis of linear CA is the representation of the linear global map. Martin et al. [1984] use dipolynomials to represent the configuration of a periodic boundary CA. The next configuration is obtained by multiplying the present configuration with a fixed polynomial (which represents the local rule) modulo $X^N - 1$, for an N cell CA. The algebra of dipolynomials is then used in the algebraic analysis of the map. The extension of this method to multidimensional CA is possible, but requires working with multivariate dipolynomials, which is difficult (see Martin et al. [1984] for details). However, the technique of dipolynomials cannot be used directly for the null boundary condition. Another way to use dipolynomials (or polynomials) to handle null boundary conditions arises from a nice technique introduced in Martin et al. [1984], whereby an N-cell null boundary 1-d CA can be embedded in a $(2N + 2)$-cell periodic boundary 1-d CA. Kawahara et al. [1995] extended this approach to study

2-d null boundary CA. However, the polynomial method fails for hybrid CA. A different approach to the problem, and one that is extensively used in VLSI applications, is to represent the global rule of a CA by a matrix. For an uniform periodic boundary 1-d CA, the matrix is circulant, and for nearest neighborhood null boundary 1-d CA, the matrix is tridiagonal. The characteristic and minimal polynomial for this matrix encodes all information about the STD of the CA; for details, see Barua and Ramakrishnan [1996] and Sutner [1996]. A generalization to multidimensional CA results in the linear operator being represented by a sum of Kronecker products of certain special matrices [Sarkar and Barua 1998a]. Another approach to multidimensional linear CA can be found in Le Bruyn and Van den Bergh [1991], where each cell state is considered a vector.

All of the above discussion is for CA on grids. However, linear CA on arbitrary graphs was studied by Sutner [1989b; 1988a]. In Sutner [1989b], it is shown that the all-ones configuration is not a Garden of Eden for a linear binary CA on any finite graph. For a CA on a finite undirected graph with addition carried out in some finite Abelian monoid, the predecessor existence problem is studied in Sutner [1988a]. It is shown that the problem is polynomial time solvable if the underlying monoid is a group and is NP-complete for an arbitrary monoid. Further, a linear time algorithm is presented to decide reversibility over a special class of graphs.

A more abstract treatment of linear CA, where the cell space is an Abelian group and the state space is a finite commutative ring, can be found in Aso and Honda [1985]. An interesting decomposition of a CA with state space Z_m, into a set of CA with state space power of a prime which divides m is also presented in Aso and Honda [1985]. In yet another approach to the study of linear CA, the generating function for the temporal sequence of a cell is studied and is shown to be an algebraic series [Litow and Dumas 1993]. Additional results on linear CA can be found in the work of Jen [1988]; for fractal properties of infinite linear CA, see Section 4.

4.6.2 *VLSI Applications*. One important area of application for finite CA is in VLSI design; see Chaudhuri et al. [1997] for details of applications of additive cellular automata to VLSI. The local communication structure of CA and the homogeneous nature of each cell are provided as strong arguments in favor of using CA for VLSI. In its use as a VLSI structure, it is often offered as a replacement for the Linear Feedback Shift Register (LFSR). Perhaps the most successful area of applying VLSI for CA is generation of pseudorandom sequences, and their use in built-in self-test (BIST). The successive configurations of a CA are taken as a random sequence. Other areas of VLSI where CA is used are in error-correcting codes, private key cryptosystem, design of associative memory, aliasing, and testing the finite state machine.

In the VLSI context, the 1-d binary CA is most common, though use of a 2-d structure has been reported. Since nonlinear CA cannot be analyzed satisfactorily, they are not used in applications. Most applications are based on CA where the global map is a linear or affine map. Another important feature of CA used in VLSI applications is the null boundary condition, since periodic boundary conditions require "long distance" communication between the end cells. Also, the CA structure is usually a hybrid one, where each cell has its own local rule. For theoretical questions regarding hybrid 1-d CA, see Bardell [1990]; Nandi and Chaudhuri [1996]; Tezuka and Fushimi [1994]; Serra and Slater [1990]; and Sarkar and Barua [1998b]. An important problem in VLSI applications is to design a null boundary 90/150 CA given an irreducible or primitive polynomial, which is the characteristic polynomial for the CA. The problem was first mentioned in Bardell [1990], and a solution appears in Serra and Slater [1990] using a version of the Lanczos tridiagonalization algorithm over $GF(2)$. However, a much simpler and elegant algorithm appears in Tezuka and Fushimi [1994]. For periodic boundary CA, the characteristic polynomial can always be factored. This is suggested in Bardell [1990] and tackled in Nandi and Chaudhuri [1996]. Several CA-based cryptographic primitives such as stream ciphers, private key cryptosystems, public key cryptosystems, and hash functions have been proposed; see Niemi [1997] for details.

5. CONCLUSION

CAs have been studied from several different angles other than the ones mentioned here (they are important, but are not included here mainly because they are either new or have an extensive literature requiring a separate survey). A (perhaps incomplete) list of these topics includes modeling in physics [Wolfram 1986] (see also *Physica D* vols. 10 (1984) and 45 (1990)); asynchronous CA [Pighizzini 1994]; cellular neural networks [Chua and Yang 1988]; quantum CA [Richter and Werner 1996; Watrous 1996]; relation to polyomino tilings [Aigrain and Beauquier 1995]; and the interesting work done at the Santa Fe Institute on evolving a CA with genetic algorithms [Mitchell et al. 1994].

ACKNOWLEDGMENTS

The author is greatly indebted to Rana Barua for reading the manuscript and, more importantly, for all the discussions on CA that made this work possible. Comments and suggestions from anonymous referees helped in improving the presentation and treatment of several topics.

REFERENCES

ADAMI, C. 1994. On modeling life. *Artif. Life 1*, 4 (Summer 1994), 429–438.

AIGRAIN, P. AND BEAUQUIER, D. 1995. Polyomino tilings, cellular automata and codicity. *Theor. Comput. Sci. 147*, 1-2 (Aug. 7, 1995), 165–180.

ALADYEV, V. 1974. Survey of research in the theory of homogeneous structures and their applications. *Math. Biosci. 15*, 121–154.

AMOROSO, S. AND COOPER, G. 1971. Tessellation structures for reproduction of arbitrary patterns. *J. Comput. Syst. Sci. 5*, 455–464.

AMOROSO, S. AND PATT, Y. 1972. Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *J. Comput. Syst. Sci. 6*, 448–464.

ARBIB, M. 1966. Simple self-reproducing universal automata. *Inf. Control 9*, 177–189.

ASO, H. AND HONDA, N. 1985. Dynamical characteristics of linear cellular automata. *J. Comput. Syst. Sci. 30*, 291–317.

BANKS, E. 1970. Cellular automata. AI Memo 198. MIT Artificial Intelligence Laboratory, Cambridge, MA.

BARDELL, P. 1990. Analysis of cellular automata used as pseudorandom pattern generators. In *Proceedings of the on IEEE International Test Conference*, IEEE Press, Piscataway, NJ, 762–768.

BARUA, R. AND RAMAKRISHNAN, S. 1996. σ -Game, σ^+-game and two-dimensional additive cellular automata. *Theor. Comput. Sci. 154*, 2, 349–366.

BLANCHARD, F., KURKA, P., AND MAASS, A. 1997. Topological and measure-theoretic properties of one-dimensional cellular automata. *Physica D 103*, 1-4, 86–99.

BRAGA, G., CATTANEO, G., FLOCCHINI, P., AND VOGLIOTTI, C. Q. 1995. Pattern growth in elementary cellular automata. *Theor. Comput. Sci. 145*, 1-2 (July 10, 1995), 1–26.

BURKS, A., Ed. 1970. *Essays on Cellular Automata*. University of Illinois Press, Champaign, IL.

BUTLER, J. T. 1974. A note on cellular automata simulations. *Inf. Control 3*, 286–295.

CHANG, J. H., IBARRA, O. H., AND VERGIS, A. 1988. On the power of one-way communication. *J. ACM 35*, 3 (July 1988), 697–726.

CHAUDHURI, P. E. AL. 1997. *Additive Cellular Automata Theory and Applications. Vol. 1*. IEEE Press advances in circuits and systems series. IEEE Press, Piscataway, NJ.

CHOFFRUT, C AND CULIK, K 1984. On real-time cellular automata and trellis automata. *Acta Inf. 21*, 4 (Nov. 1984), 393–407.

CHUA, L. AND YANG, L. 1988. Cellular neural networks: Theory and applications. *IEEE Trans. Circ. Syst.*, 1257–1290.

CODD, E. 1968. *Cellular Automata*. Academic Press, Inc., New York, NY.

COLE, S. 1969. Real-time computation by n-dimensional iterative arrays of finite state machine. *IEEE Trans. Comput. 18*, 349–365.

CONWAY, J., GUY, R., AND BERLEKAMP, E. 1992. *Winning Ways: For Your Mathematical Plays, vol. 2*.

CULIK II, K. 1987. On invertible cellular automata. *Complex Syst. 1*, 1035–1044.

CULIK II, K. 1989. Variations of the firing squad problem and applications. *Inf. Process. Lett. 30*, 3 (Feb. 1989), 153–157.

CULIK II, K. AND DUBE, S. 1991. An efficient solution of the firing mob problem. *Theor. Comput. Sci. 91*, 1 (Dec. 9, 1991), 57–69.

CULIK II, K, GRUSKA, J, AND SALOMAA, A 1986. Systolic trellis automata: Stability, decidability and complexity. *Inf. Control 71*, 3 (Dec. 1986), 218–230.

CULIK, K., HURD, L. P., AND YU, S. 1990. Computation theoretic aspects of cellular automata. *Physica D 45*, 1-3 (Sep. 1990), 357–378.

CULIK, K., PACHL, J., AND YU, S. 1989. On the limit sets of cellular automata. *SIAM J. Comput. 18*, 4 (Aug. 1989), 831–842.

CULIK II, K. AND YU, S. 1988. Undecidability of CA classification schemes. *Complex Syst. 2*, 2 (Apr., 1988), 177–190.

CULIK, K. AND YU, S. 1991. Cellular automata, $\omega\omega$-regular sets, and sofic systems. *Discrete Appl. Math. 32*, 2 (July 1991), 85–101.

DUBACQ, J. 1995. How to simulate Turing machines by invertible one-dimensional cellular automata. *Int. J. Foundations Comput. Sci. 6*, 4, 395–402.

DURAND, B. 1994. Inversion of 2D cellular automata: some complexity results. *Theor. Comput. Sci. 134*, 2 (Nov. 21, 1994), 387–401.

DURAND, B. 1995. A random NP-complete problem for inversion of 2D cellular automata. *Theor. Comput. Sci. 148*, 1 (Aug. 21, 1995), 19–32.

DYER, C. 1980. One-way bounded cellular automata. *Inf. Control 44*, 261–281.

FINELLI, M., MANZINI, G., AND MARGARA, L. 1998. Lyapunov exponents versus expansivity and sensitivity in cellular automata. *J. Complexity 14*, 2, 210–233.

GÁCS, P. 1986. Reliable computation with cellular automata. *J. Comput. Syst. Sci. 32*, 1 (Feb. 1986), 15–78.

GACS, P. 1997. Reliable cellular automata with self-organization. In *Proceedings of the Conference on Foundations of Computer Science*,

GARDNER, M. 1970. The fantastic combinations of John Conway's new solitaire game "Life". *Sci. Am. 223*, 120–123.

GARDNER, M. 1971. On cellular automata, self-reproduction, the Garden of Eden and the game of "Life". *Sci. Am. 224*, 112–117.

GARZON, M. 1995. *Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks*. EATCS monographs on theoretical computer science. Springer-Verlag, Berlin, Germany.

GOLZE, U. 1976. Differences between 1- and 2-dimensional cell spaces. In *Automata, Languages, Development*. North-Holland Publishing Co., Amsterdam, The Netherlands, 369–384.

GREEN, F. 1987. NP-complete problems in cellular automata. *Complex Syst. 1*, 453–474.

GUAN, P. AND HE, Y. 1986. Exact results for deterministic cellular automata with additive rules. *J. Stat. Phys. 43*, 463–478.

HAMACHER, V. 1971. Machine complexity versus interconnection complexity in iterative arrays. *IEEE Trans. Comput. C-20* (Apr.), 321–323.

HARAO, M. AND NOGUCHI, S. 1975. Fault tolerant cellular automata. *J. Comput. Syst. Sci. 11*, 171–185.

HARAO, M. AND NOGUCHI, S. 1978. On some dynamical properties of finite cellular automata. *IEEE Trans. Comput. 27*, 1.

HEDLUND, G. 1969. Endomorphisms and automorphisms of the shift dynamical systems. *Math. Syst. Theory 4*, 3, 320–375.

HERMAN, G. 1974. Synchronization of growing cellular arrays. *Inf. Control 25*, 2, 103–122.

HOLLAND, J. 1976. Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammers. In *Automata, Languages, Development*. North-Holland Publishing Co., Amsterdam, The Netherlands, 385–404.

HURD, L. 1987. Formal language characterizations of cellular automata limit sets. *Complex Syst. 1*, 69–80.

HURD, L., KARI, J., AND CULIK II, K. 1992. The topological entropy of cellular automata is uncomputable. *Ergodic Theor. Dynamic. Syst. 12*, 255–265.

IBARRA, O. H. AND JIANG, T. 1987. On one-way cellular arrays. *SIAM J. Comput. 16*, 6 (Dec. 1, 1987), 1135–1154.

IBARRA, O. AND KIM, S. 1984. Characterizations and computational complexity of systolic trellis automata. *Theor. Comput. Sci. 29*, 123–153.

IBARRA, O. H., PALIS, M. A., AND KIM, S. M. 1985. Fast parallel language recognition by cellular automata. *Theor. Comput. Sci. 41*, 2,3 (Mar. 1985), 231–246.

IBARRA, O., PALIS, M., AND KIM, S. 1985. Some results concerning linear iterative (systolic) arrays. *J. Parallel Distrib. Comput. 2*, 182–218.

IKEGAMI, T. AND HASHIMOTO, T. 1995. Active mutation in self-reproducing networks of machines and tapes. *Artif. Life 2*, 3, 305–318.

ITO, M., OSATO, N., AND NASU, M. 1983. Linear cellular automata over Z_m. *J. Comput. Syst. Sci. 27*, 125–140.

JEN, M. 1988. Linear cellular automata and recurring sequences in finite fields. *Commun. Math. Phys. 119*, 13–28.

JUMP, J. AND KIRTANE, J. 1974. On the interconnection structure of cellular networks. *Inf. Control 24*, 74–91.

KARI, J. 1990. Reversibility of 2D cellular automata is undecidable. In *Cellular Automata: Theory and Experiment*, H. Gutowitz, Ed. MIT Press, Cambridge, MA, 379–385.

KARI, J. 1992. The nilpotency problem of one-dimensional cellular automata. *SIAM J. Comput. 21*, 3 (June 1992), 571–586.

KARI, J. 1994. Reversibility and surjectivity problems of cellular automata. *J. Comput. Syst. Sci. 48*, 1 (Feb. 1994), 149–182.

KARI, L., ROZENBERG, G., AND SALOMAA, A. 1997. L systems. In *Handbook of Formal Languages, vol. 1: Word, Language, Grammar*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, New York, NY, 253–328.

KAWAHARA, Y. ET AL. 1995. Period lengths of cellular automata on square lattices with rule 90. *J. Math. Phys. 36*, 3, 1435–1456.

KNUTH, D. E. 1997. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. 3rd ed. Addison-Wesley Longman Publ. Co., Inc., Reading, MA.

KOSARAJU, S. R. 1975. Speed of recognition of context-free languages by array automata. *SIAM J. Comput. 4*, 3 (Sept.), 331–340.

KUNG, S. Y. 1987. *VLSI Array Processors*. Prentice-Hall information and system sciences series. Prentice-Hall, Inc., Upper Saddle River, NJ.

KURKA, P. 1997. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theor. Dynamic. Syst. 17*, 229–254.

LANGTON, C. 1984. Self-reproduction in cellular automata. *Physica D 10*, 134–144.

LE BRUYN, L. AND VAN DEN BERGH, M. 1991. Algebraic properties of linear cellular automata. *Linear Alg. Appl. 157*, 217–234.

LEE, H. AND KAWAHARA, Y. 1996. Transition diagrams of finite cellular automata. *Bull. Inf. Cybern. 28*, 1.

LINDENMAYER, A. 1968. Mathematical models for cellular interactions in development. parts I and II. *J. Theor. Biol. 18*, 280–315.

LITOW, B. AND DUMAS, PH. 1993. Additive cellular automata and algebraic series. *Theor. Comput. Sci. 119*, 2 (Oct. 25, 1993), 345–354.

MACHÌ, A. AND MIGNOSI, F. 1993. Garden of Eden configurations for cellular automata on Cayley graphs of groups. *SIAM J. Discrete Math. 6*, 1 (Feb. 1993), 44–56.

MAHAJAN, M. 1992. Studies in language classes defined by different types of time-varying cellular automata. Ph.D. Dissertation.

MANZINI, G. AND MARGARA, L. 1999. Attractors of linear cellular automata. *J. Comput. Syst. Sci.*.

MANZINI, G. AND MARGARA, L. 1998. Invertible linear cellular automata over Z_m : Algorithmic and dynamical aspects. *J. Comput. Syst. Sci. 56*, 1, 60–67.

MARTIN, B. 1994. A universal cellular automata in quasi-linear time and its S-m-n form. *Theor. Comput. Sci. 123*, 2 (Jan. 31, 1994), 199–237.

MARTIN, O., ODLYZKO, A., AND WOLFRAM, S. 1984. Algebraic properties of cellular automata. *Commun. Math. Phys. 93*, 219–258.

MARUOKA, A. AND KIMURA, M. 1974. Completeness problem of one-dimensional binary scope-3 tessellation automata. *J. Comput. Syst. Sci. 9*, 1, 31–47.

MARUOKA, A. AND KIMURA, M. 1977. Completeness problem of multi-dimensional tessellation automata. *Inf. Control 35*, 1, 52–86.

MAZOYER, J. 1987. A six-state minimal time solution to the firing squad synchronization problem. *Theor. Comput. Sci. 50*, 2 (Sept. 1987), 183–238.

MAZOYER, J. AND REIMEN, N. 1992. A linear speed-up theorem for cellular automata. *Theor. Comput. Sci. 101*, 1 (July 13, 1992), 59–98.

MICHELE, D., MANZINI, G., AND MARGARA, L. 1988. On computing the entropy of cellular automata. In *Proceedings of the 13th International Colloquium on Automata, Languages and Programming* (Rennes, France, July), L. Kott, Ed. Elsevier Sci. Pub. B. V., Amsterdam, The Netherlands.

MITCHELL, M., CRUTCHFIELD, J. P., AND HRABER, P. T. 1994. Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D 75*, 1-3 (Aug. 1, 1994), 361–391.

MOORE, E., Ed. 1964. *Sequential Machines. Selected Papers*. Addison-Wesley Publishing Co., Inc., Redwood City, CA.

MOORE, E. AND LANGDON, G. 1968. A generalized firing squad problem. *Inf. Control 12*, 212–220.

MORITA, K. 1992. Computation-universality of one-dimensional one-way reversible cellular automata. *Inf. Process. Lett. 42*, 6 (July 24, 1992), 325–329.

MORITA, K. 1995. Reversible simulation of one-dimensional irreversible cellular automata. *Theor. Comput. Sci. 148*, 1 (Aug. 21, 1995), 157–163.

MYHILL, J. 1963. The converse of Moore's Garden-of-Eden theorem. *Proc. Am. Math. Soc. 14*, 685–686.

NANDI, S. AND CHAUDHURI, P. P. 1996. Analysis of periodic and intermediate boundary 90/150 cellular automata. *IEEE Trans. Comput. 45*, 1, 1–12.

NANDI, S., KAR, B., AND PAL CHAUDHARI, P. 1994. Theory and applications of cellular automata in cryptography. *IEEE Trans. Comput. 43*, 12 (Dec. 1994).

NGUYEN, H. AND HAMACHER, V. 1974. Pattern synchronization in two-dimensional cellular spaces. *Inf. Control 26*, 12–23.

NIEMI, V. 1997. Cryptology: language-theoretic aspects. In *Handbook of Formal Languages, vol. 2: Linear Modeling: Background and Application*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, New York, NY, 507–524.

NISHIO, H. AND KOBUCHI, Y. 1975. Fault tolerant cellular spaces. *J. Comput. Syst. Sci. 11*, 150–170.

OSTRAND, T. 1971. Pattern recognition in tessellation automata of arbitrary dimensions. *J. Comput. Syst. Sci. 5*, 623–628.

PACKARD, N. 1985. Two-dimensional cellular automata. *J. Stat. Phys. 30*, 901–942.

PELLETIER, D. H. 1987. Merlin's magic square. *Am. Math. Monthly 94*, 2 (Feb. 1987), 143–150.

PESAVENTO, U. 1995. An implementation of von Neumann's self-reproducing machine. *Artif. Life 2*, 4, 337–354.

PIGHIZZINI, G. 1994. Asynchronous automata versus asynchronous cellular automata. *Theor. Comput. Sci. 132*, 1-2 (Sept. 26, 1994), 179–207.

PRUSINKIEWICZ, P. AND LINDENMAYER, A. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, NY.

RICHARDSON, D. 1972. Tessellations with local transformations. *J. Comput. Syst. Sci. 6*, 373–388.

RICHTER, S. AND WERNER, R. 1996. Ergodicity of quantum cellular automata. *J. Stat. Phys. 82*, 963–998.

RÓKA, Z. 1994. One-way cellular automata on Cayley graphs. *Theor. Comput. Sci. 132*, 1-2 (Sept. 26, 1994), 259–290.

ROKA, Z. 1995. The firing squad synchronization problem on Cayley graphs. In *Proceedings of the Conference on MFCS*, Lecture Notes in Computer Science Springer-Verlag, New York, 402–411.

SARKAR, P. 1996. σ^+-automata on square grids. *Complex Syst. 10*, 121–141.

SARKAR, P. AND BARUA, R. 1998a. Multidimensional σ-automata, π-polynomials and generalised S-matrices. *Theor. Comput. Sci. 197*, 1-2, 111–138.

SARKAR, P. AND BARUA, R. 1998b. The set of reversible 90/150 cellular automata is regular. *Discrete Appl. Math. 84*, 1-3, 199–213.

SATO, T. 1994. Group structured linear cellular automata over *Zm*. *J. Comput. Syst. Sci. 49*, 1 (Aug. 1994), 18–23.

SEIFERAS, J. 1982. Observations on nondeterministic multidimensional iterative arrays. In *Proceedings of the ACM Symposium on the Theory of Computing*, ACM Press, New York, NY, 276–289.

SERRA, M. ET AL. 1990. The analysis of one-dimensional cellular automata and their aliasing properties. *IEEE Trans. CAD/ICAS 9*, 7, 767–778.

SERRA, M. AND SLATER, T. 1990. A Lanczos algorithm in a finite field and its applications. *J. Comb. Math. Comb. Comp.*.

SHERESHEVSKY, M. 1992. Lyapunov exponents for one-dimensional cellular automata. *J. Nonlinear Sci. 2*, 1–8.

SHINAHR, I. 1974. Two- and three-dimensional firing-squad synchronization problems. *Inf. Control 24*, 163–180.

SMITH III, A. 1971. Cellular automata complexity trade-offs. *Inf. Control 18*, 466–482.

SMITH III, A. 1972. Real-time language recognition by one-dimensional cellular automata. *J. Comput. Syst. Sci. 6*, 233–253.

SMITH III, A. 1976. Introduction to and survey of polyautomata theory. In *Automata, Languages, Development.* North-Holland Publishing Co., Amsterdam, The Netherlands.

SOMMERHALDER, R. AND WESTRHENEN, S. V. 1983. Parallel language recognition in constant time by cellular automata. *Acta Inf. 19*, 397–407.

SUTNER, K. 1988a. Additive automata on graphs. *Complex Syst. 2*, 6 (Dec. 1988), 649–661.

SUTNER, K. 1988b. On σ-automata. *Complex Syst. 2*, 1 (Feb, 1988), 1–28.

SUTNER, K. 1989a. The computation complexity of cellular automata. In *Fundamentals of Computation Theory* Lecture Notes in Computer Science. Springer-Verlag, New York, 451–459.

SUTNER, K. 1989b. Linear cellular automata and the Garden-of-Eden. *Math. Intell. 11*, 2, 49–53.

SUTNER, K. 1989c. A note on the Culik-Yu classes. *Complex Syst. 3*, 1, 107–115.

SUTNER, K. 1990a. Classifying circular cellular automata. *Physica D 45*, 1-3 (Sep. 1990), 386–395.

SUTNER, K. 1990b. The σ-game and cellular automata. *Am. Math. Monthly 97*, 1 (Jan. 1990), 24–34.

SUTNER, K. 1991. De Bruijn graphs and linear cellular automata. *Complex Syst. 5*, 19–30.

SUTNER, K. 1995. On the computational complexity of finite cellular automata. *J. Comput. Syst. Sci. 50*, 1 (Feb. 1995), 87–97.

SUTNER, K. 1999. σ-automata and Chebyshev-polynomials. *Theor. Comput. Sci.*.

TAKAHASHI, S. 1992. Self-similarity of linear cellular automata. *J. Comput. Syst. Sci. 44*, 1 (Feb. 1992), 114–140.

TERRIER, V. 1996. Language not recognizable in real time by one-way cellular automata. *Theor. Comput. Sci. 156*, 1-2, 281–287.

TEZUKA, S. AND FUSHIMI, M. 1994. A method of designing cellular automata as pseudorandom number generators for built-in self-test for VLSI. In *Finite Fields: Theory, Applications, and Algorithms* Contemporary Mathematics: Cont. Math.. AMS, Providence, RI, 363–367.

TOFFOLI, T. 1977. Computation and construction universality of reversible cellular automata. *J. Comput. Syst. Sci. 15*, 213–231.

TOFFOLI, T. AND MARGOLUS, N. 1990. Invertible cellular automata: a review. *Physica D 45*, 1-3 (Sep. 1990), 229–253.

VARSHAVSKY, V., MARAKHOVSKY, V., AND PECHANSKY, V. 1969. Synchronization of interacting automata. *Math. Syst. Theory 4*, 3, 212–230.

VITANYI, P. 1973. Sexually reproducing cellular automata. *Math. Biosci. 18*, 23–54.

VOLLMAR, T. 1977. Cellular spaces and parallel algorithms, an introductory survey. In *Parallel Computation-Parallel Mathematics*, M. Feilmeier, Ed. North-Holland Publishing Co., Amsterdam, The Netherlands, 49–58.

VON NEUMANN, J. 1963a. The general and logical theory of automata. In *J. von Neumann Collected Works*, A. Taub, Ed.

VON NEUMANN, J. 1963b. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In *J. von Neumann Collected Works*, A. Taub, Ed.

VON NEUMANN, J. AND BURKS, A. W., Eds, 1966. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL.

WAKSMAN, A. 1966. An optimum solution to the firing squad synchronization problem. *Inf. Control 9*, 67–78.

WATROUS, J. 1996. On one-dimensional quantum cellular automata. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science* (FOCS '96), IEEE Computer Society Press, Los Alamitos, CA, 528–537.

WILLSON, S. 1978. On convergence of configurations. *Discrete Math. 23*, 279–300.

WILLSON, S. 1981. Growth patterns of ordered cellular automata. *J. Comput. Syst. Sci. 22*, 29–41.

WILLSON, S. 1984a. Cellular automata can generate fractals. *Discrete Appl. Math. 8*, 91–99.

WILLSON, S. 1984b. Growth rates and fractional dimensions in cellular automata. *Physica D 10*, 69–74.

WOLFRAM, S. 1983. Statistical mechanics of cellular automata. *Rev. Modern Phys. 55*, 601–644.

WOLFRAM, S. 1984a. Computation theory of cellular automata. *Commun. Math. Phys. 96*, 15–57.

WOLFRAM, S. 1984b. Universality and complexity in cellular automata. *Physica D 10*, 1–35.

WOLFRAM, S. 1986. *Theory and Applications of Cellular Automata: Including Selected Papers 1983-1986*. World Scientific Publishing Co., Inc., River Edge, NJ.

YAKU, T. 1973. The constructibility of a configuration in a cellular automata. *J. Comput. Syst. Sci. 7*, 4, 481–496.

YAMADA, H. AND AMOROSO, S. 1969. Tessellation automata. *Inf. Control 14*, 299–317.

YAMADA, H. AND AMOROSO, S. 1970. A completeness problem for pattern generation in tessellation automata. *J. Comput. Syst. Sci. 4*, 137–176.

YAMADA, H. AND AMOROSO, S. 1971. Structural and behavioural equivalences of tessellation automata. *Inf. Control 18*, 1–31.

Received: June 1998; revised: January 1999; accepted: February 1999