2010

To live and die in CA

Jane Frances Curnutt

TO LIVE AND DIE IN CA

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Jane Frances Curnutt

December 2010

TO LIVE AND DIE IN CA

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Jane Frances Curnutt

December 2010

Approved by:

[redacted]

Yasha Karant, Advisor, School of Computer Science and Engineering

7 Dec 2010
Date

[redacted]

Ernesto Gomez

[redacted]

Keith Evan Schubert

ABSTRACT

In the search for life on Mars and other extraterrestrial bodies, how do we recognize life or the remains of ancient life when we find it? We will need to recognize residual patterns left by life. One approach to recognizing these kinds of patterns is to look at patterns created and left by life in extreme environments here on Earth. In this thesis, we investigate the nature of elementary cellular automata to better understand their relationship of the models they support to the biological organisms that create the mats and soil crusts found in extreme environments here on earth. A better understanding of cellular automata will also make it possible for us to match the cellular automata more effectively to the biological area being modeled. This investigation has led us to develop a graphical grammar for simple cellular automata, using L-systems, a grammar system designed by a biologist, Aristid Lindenmayer, to describe growth in biological systems. We also discuss scaling algorithms, and the associated variable dependencies that create them. Future directions for research in both modeling with cellular automata, and cellular automata theory are discussed.

ACKNOWLEDGEMENTS

Many thanks to Penny Boston and Chris McKay for their continuing encouragement. Thanks also, to my hard working committee. Particular note to my advisor, whose contributions cannot be expressed. Thanks to Catharine Hoaglund, who kept me on track with encouragement and gentle chiding, when I was lost in the wilderness, and did yeoman's work proof reading the manuscript. Bless you, Cathy.

Special thanks to Dr. Richard Botting, who, although not part of my committee, became an integral part of this thesis by freely giving of his wisdom and time.

DEDICATION

To Norval Curnutt, who made it possible for me to soar, and always believed in me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Overview

A key aspect of planning a space mission is to set scientific mission objectives with the ability to adapt them based on observations and mission situations. The search for extraterrestrial life is a major scientific objective, but the exact nature of that life and how to confirm it is a major debate. A further problem in the search for life in space is how to select the areas that we want to investigate more intensively. Photo surveys, geology, and knowledge of biology here on Earth can target a likely area, but, unlike the Martian dust devils [29], there is no motion to point out the right direction from there.

We use life in extreme environments on Earth as analogs for the kinds of life that we could encounter in space. We propose using these analogs to create a series of templates which could be used to indicate areas that might be worth deeper investigation. In resource limited environments, organisms grow in patterns that are self-enforcing and exhibit hysteresis [1, 26, 57] which can be used to recognize them and their fossils at a distance. Particularly on Mars, as the environment became less hospitable, extremophiles similar to Earth's were likely the last to survive, and should be the easiest to find. It has recently been suggested that fossilized bacteria may be identified in the rock varnish on Mars [50, 15]. Among the techniques that have been

used to model these patterns are evolutionarily stable strategies in game theory and differential equations [24, 18, 26, 55, 57].

While good results have been generated using differential equations, they require tuning of the parameters and experience in mathematical and numerical techniques to obtain valid results. In previous research [12, 3, 52, 40, 53, 4]we have developed cellular automata that produce similar predictions to the differential equation models, while preserving the rapid modeling and hypothesis testing of cellular automata (CA). Similar models can also be applicable to group animal behavior [11, 20, 25, 31]. Our method for deriving rules for cellular automata from observed data in organism growth patterns accounts for soil nutrients, water, root growth patterns, and geology, allowing scientists to easily examine the effects of modifying conditions without damaging the environment. We apply this model to identify factors affecting patterning with respect to growth, die-out, and stabilization in extreme environments. We have compared the results of our model with better to say biological soil crusts or microbiotic soil crusts growth in Zzyzx, CA. These models could be used to rapidly check data from space missions to rate the potential of various locations for the possibility of life or fossils.

To further refine our modeling, this thesis seeks to investigate the nature of simple cellular automata. The results from this thesis will allow us to make better informed decisions when setting up parameters for modeling, and may make it possible, in the future, to do in situ modeling in locations with limited computer resources, such as Mars.

1.2 Background

In the 1940s, John von Neumann developed the first cellular automata, while working on the self-replicating systems problem. Von Neumann's original CAs contained 29 states in each cell. In 1970, John Conway developed his Game of Life, a two dimensional cellular automaton that exhibited aspects of both order and randomness. In 1983, Stephen Wolfram published the first of many papers on cellular automata. His research into this area of mathematics culminated in 2002 with the publication of his book, "A New Kind of Science."

Cellular automata have been used to study growth and patterns in forests, arid desert environments, predator-prey problems, and sea shells. It has also been used to study areas as diverse as epidemiology and linguistics. Cellular automata have been used as the core of computer games as well. In particular the computer game, Sim City, is driven by a cellular automaton [61].

1.3 Literature Survey

1.3.1 Complexity and Cellular Automata

One class of CAs are called totalistic cellular automata. These include Conway's game of life [17]. A totalistic CA is one in which the value of a particular cell in the next time step is based on the sum of the values of the cells immediately surrounding the target cell.

Alvy Ray Smith wrote his PhD thesis, Cellular Automata Theory [44], in 1969, at Stanford University. He wrote a number of papers on cellular automata between 1968

and 1993. Many of his papers focused on the application of formal language theory to cellular automata [47, 49, 45]. In his paper, Two-Dimensional Formal Languages and Pattern Recognition by Cellular Automata [47], Smith developed a measure of computing speed for two dimensional CAs based on the increase in the perimeter size with respect to the number of time steps.

Tommaso Toffoli did his PhD thesis, Cellular Automata Mechanics [56], in 1977, while part of The Logic of Computers Group, at the University of Michigan at Ann Arbor, Michigan. His primary areas of interest were reversible CA, and parallel computing with CA.

There is an entire area of research involving chaos theory, and complexity that looks into how simple systems can generate complex results (chaos theory) and how complex systems generate simple outcomes (complexity theory). Self-organization, self-similarity, fractal theory, emergent behavior and cellular automata are all aspects of research in these areas. A hallmark of this kind of research is a holistic approach to problems which exhibit behaviors that are more than the sum of their parts or very complex systems that deliver profoundly simple results. In this sense, as early as the 1969, a German physicist, Konrad Zuse proposed, in his book, "Rechnender Raum"(*Calculating Space*) [62], that the universe is a cellular automaton. Melanie Mitchell has written an excellent survey of the sciences of complexity in her book, "Complexity a Guided Tour" [27]. Mitchell's book, published in 2009, provides an up-to-date context for cellular automata within the sciences of complexity theory. Russ Abbott [2], in his paper, "Emergence Explained", discusses the area of complexity and emergence, and proposes that a Turing machine emulator can be implemented on a

cellular automata. Martin Nowak [30] uses CAs in what he calls spatial games to study the struggle between cooperators and defectors in his book, "Evolutionary Dynamics". Palash Sakar published an excellent survey of the history of cellular automata [36]. Stephen Wolfram [59] has written a number of papers cellular automata since the early 1980s. This work culminated in the publication, in 2002, of his magnum opus, "A New Kind of Science" [60], which is an exhaustive catalog of cellular automata .

In "Keith on Numerical Analysis" [39], Dr. Keith Schubert discusses using the three point formula to numerically approximate first and second derivatives by taking the point slope form, $y = mx + b$, over the two points $f(x)$ and $f(x+h)$, applying the same formula to the two points $f(x)$ and $f(x-h)$ and averaging the result to approximate the first derivative of $f(x)$. This can be extended to the numerical estimation of the second derivative by using the same technique with $f'(x)$ and $f'(x \pm \frac{h}{2})$.Brian Strader in his master's thesis [51] develops a set of guidelines for choosing parameters discretizing PDEs. He looks at optimizing these choices to provide stability and convergence. Using the three point formula technique, Strader [51], points out that that in the resulting approximation, that $f(x)$ depends on the values of two equidistantly spaced points at $f(x \pm h)$. He goes on to state that these numerical results point to a possible link between continuous partial differential equations and discretized cellular automata.

1.3.2 Biology and Self-Organization

The idea for Lindenmayer systems, or L-systems [33] began in 1968 with Aristid Lindenmayer, who was the director of the Theoretical Biology Group at the University

of Utrecht. The original idea was to provide a theoretical framework for studying plants. Lindenmayer later collaborated with the Computer Graphics Group at the University of Regina, under the supervision of Przzemyslaw Prusinkiewicz. This interdisciplinary collaboration lead to the development of a graphical grammar with roots in both biology and formal language theory. Since its inception, L-systems has gone on to be used in computer graphics for the nearly photorealistic computer visualization of plant structures.

A number of papers have been published that model plant growth patterns using partial differential equations. E. Meron, et al, have studied patchiness in the Negev Desert, modeling the vegetation patterns as a function of the rainfall gradient in that area, and proposing a hysteresis loop that is part of the desertification and recovery cycle [26, 57].

One of the most referenced papers in the area is, Self-organized Patchiness and Catastrophic Shifts in Ecosystems, by Reitkerk, et al [34]. This article studies the relationship between self-organized patchiness and catastrophic shifts in various ecosystems, using CAs to model the patchiness in an attempt to predict the catastrophic shifts the behavior of the patchiness both in situ and in the cellular automata model. P. Hogeweg has used CAs to model ecologies [21]. D. L. Dunkerly has used CAs to model banded vegetation with uniform rainfall [16]

William Schlesinger and his coauthors discuss the desertification of the Jornada Experimental Range in southern New Mexico. This was formerly grasslands [38]. They propose the change from grasslands to desert is a change from a homogeneous environment for grasslands to isolated discrete environments that occur when, for any

one of a number of reasons, the grasslands become deserts.

In our research with Penelope Boston, Director, Cave and Karst Institute, New Mexico Tech, [12, 3, 52, 40, 53, 4] we have used CAs to model the patterns of biovermiculation growth that appear inside caves, and on the structure walls at Palenque. Boston has authored or coauthored a number of books and articles on plant growth in caves [5], sulfuric acid caves, [7, 22, 23], and astrobiology [8, 5]. Boston and Michael Spilde were the first people to identify the correlation between the cellular automata models our research group was working on in 2006 and 2007, and the biovermiculations that they had identified in sulfuric acid caves in Mexico [6]. New Scientist [15] reported on a presentation by M.N. Spilde [50] stating that a form of desert varnish was identified in caves in Snowy River, NM, and speculating that, since the source of desert varnish here on Earth is bacterial interaction with dust and manganese, perhaps the first extraterrestrial life that we will identify on Mars will be in the rock varnish there.

1.4 Contributions

In this research we have discovered high level abstractions to model cellular automata that may be of use in the resource-constrained computing environments of a Mars lander. These rules provide us with a better understanding and the insight to be more effective in choosing the best variables for a particular CA model. In the process, we have discovered a connection to L-system grammar that opens many new lines of future research.

1.5 Document Organization

The rest of this thesis is organized into four major sections, plus a chapter that includes contributions and future research. The first major section is chapter 2, Initial Conditions, which contains a brief introduction to CA. It also contains a discussion of the types of CA we chose to work with, and the decisions that we made regarding the experimental setup and why those choices were made. There is a discussion of calculation methods and the advantages and disadvantages to each method. A survey of examples are shown to demonstrate the different development of different initial start patterns. Another comparison is made of the same start pattern, with the same number of time steps at six different radii. There is also a discussion and examples of an experiment with deliberate perturbation of patterns showing comparative distortion due to missing values and added values at a calculation radius equal to 1 (3×3 Moore neighborhood), and comparing the effect of exactly the same perturbation at a calculation radius equal to 3 (7×7 neighborhood). To summarize, this chapter establishes our experimental perspective on cellular automata and the associated methodology that we chose.

The next major section is chapter 3, Fractal Cellular Automata, which surveys growth behavior of elementary CA with the minimal variables of $birth = 1$, $radius = 1$, four primal start patterns. Chapter 3 also introduces the effect of the radius of calculation on the developing pattern. The effect of adding a death value to this set of variables is observed. The contribution each of these variables makes to fractal behavior of this very elementary 2-dimensional CA is discussed. Two visual analysis techniques are introduced. A brief foray is made into the effect of larger (and smaller)

calculation radii, when birth numbers are held constant. The observation is made that through out the material presented in chapter 3, the variables have been independent of each other.

The third major section is chapter 4, On Birth and Rules, that takes the observations that we made on growth behavior in chapter 3, isolates the variable responsible for it, and creates a context sensitive graphical grammar for simple CAs by adapting a grammar, L-systems, designed by Aristid Lindenmayer [33] (a biologist), to describe the growth behavior of plants.

The final major section is chapter 5, The Effect of Scaling on Developing Patterns, which uses a family of nearly identical CA patterns to investigate the effect of scaling start patterns, birth numbers, and calculation radii, together. We demonstrate that the variables that were previously independent of each other, are now each dependent on the others.

Finally, chapter 6, Conclusions, contains the sections, Contributions, and Future Research.

Appendix 6.2 with a full listing of the Scilab code used to do the research for this thesis.

2. INITIAL CONDITIONS

2.1 Introduction

What is a cellular automaton? A cellular automaton, is a discrete model that develops on an infinite grid or an $n \times n$ grid (like a piece of graph paper). Each cell (or square on the graph paper) has 1 or more values stored in it. The cellular automata used in this thesis have one of two values stored in each square (1 or 0). Each square is examined in every time step using a set of rules that determine what happens to the square in the next time step. These rules are based on the sum of the values stored in the eight squares surrounding the "target" square, (the square being examined). These eight squares are the target square's neighborhood. This type of neighborhood is a Moore neighborhood. (Another neighborhood configuration, a von Neumann neighborhood, uses only the four squares immediately adjacent to the sides of the the target square, and has the appearance of a plus sign.) The simplest rules are: something new can grow in an empty square (be born) if the sum (in the eight squares) $= 1$ (or some other single value ≤ 8, the maximum possible sum), and what is in the square will die (go to zero), if $sum > 3$ (notice that this value is a range between $3 < sum \leq 8$. For this example, there are values, 2 and 3, that cause neither birth, nor death. For sums of the eight surrounding squares with these values, the value currently stored in the target square stays the same. The new value for square is written to a copy

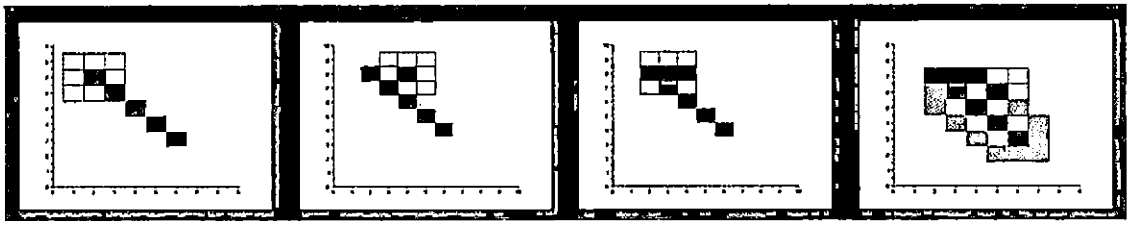

Fig. 2.1: Diagonal line with calculation neighborhoods

of the grid. Once the entire grid has been calculated, the copy and the original grid are swapped and the process begins again on the new grid. This calculation of the new grid and the subsequent swap of grids ensures that all of the cells in the grid are updated simultaneously. During the calculation process no cell is aware of the new state for any other cell. In practice, this is considerably simpler that it sounds. The key to this is the eight square neighborhood with the target square at the center. Think of a card with an aperture the size of the neighborhood cut into it. When you lay the card on the grid, all you can see is the neighborhood. Count the number of black squares in the aperture, around the target square, and apply the rules. Store the result in the copy of the grid.

In figure 3.8, the start pattern is shown in green, and the target square is in red. After we have seen the calculation neighborhood for a target square, that square will be dark blue in the following frames. The final frame shows the remaining squares that will grow in the next time step in light blue. The first frame shows a neighborhood with a target square and one square of the original diagonal line. The sum in this neighborhood is one. The birth rule for this cellular automaton is $birth = 1$, that

is to say, if there is one, and only one square in the neighborhood that is alive (has a one stored in it), then something can grow in the center or target square in the current neighborhood in the next time step. In the next frame, the neighborhood is in a different position, but the sum is the same, so this target square will grow in the next time step. In the third frame, the two previous target squares are shown in dark blue, and the neighborhood again has a sum equal to one, and will grow in the next time step. One more neighborhood and target square are shown in the final frame. It, too will grow in the next time step. The remaining squares that will grow (with neighborhoods that we didn't look at in figure 3.8, are shown in light blue.

Calculation neighborhoods come in different sizes or radii. The one shown in figure 3.8 has a radius equal to one. (In the rest of the thesis this will be referred to as $radius = 1$. This neighborhood is 3×3 squares in size, or 9 squares. A $radius = 2$ neighborhood is 5×5 squares in size, or 25 squares. A $radius = 3$ neighborhood will be seven squares wide. Most of the neighborhood used in this thesis will fall into one of those three sizes.

Death rules are applied the same way as the birth rules, by counting up the the number of occupied squares in the neighborhood and applying the rule. The one difference is that the number shown in the death rule is a lower bound on the range for the death value. If the sum in the neighborhood is such that neither the birth rule nor the death applies, the value in the target square stays the same for the next time step.

Conway's Game of Life was one of the earliest broadly available cellular automata. Its behavior has been exhaustively studied, to the point that there are several catalogs

of start patterns and their subsequent behavior. Some of these catalogs are available on the Web [9, 32, 54, 42]. Paul Callahan's Patterns, Programs, and Links for Conway's Game of Life was originally started in 1995, at Johns Hopkins. Some sites not only catalog Game of Life patterns and behavior, but also provide (sometimes exhaustive) lists of links to other Game of Life sites [9, 32]. Conway's Game of Life continues to be an active test bed for cellular automata, complexity and emergent behavior [2]. In the spirit of the exhaustive investigation of seed patterns and their behavior in Conway's Game of Life, we have adopted a similar approach.

2.2 Variables

Our initial research focused on finding the driving variables in the cellular automata models of biological patterns on cave walls and soil crusts. We determined, by experimentation [3, 4, 12, 40, 52, 53], that replicating biologically generated patterns could be satisfactorily achieved via cellular automata with just a few variables, three of which have a rough correspondence to the environmental factors that the life forms might encounter. The variables selected were:

1. initial pattern - roughly corresponds to the species of life form
2. birth rules - how difficult it is to germinate and grow
3. death rules - how difficult it is to survive
4. the radius over which these rules were applied,
5. the choice of weighted or unweighted calculations.

This set of 5 variables are used to form a tuple that uniquely identifies the initial conditions for an experimental run, and categorize the results. Other factors we looked at are the number of iterations executed, and whether or not the run had distortions, or randomness. the effects of the basic five factors are discussed in more detail in the rest of this section.

Weighted calculations had the advantage that the results from various radius calculations with the same birth and death rules could be compared to each other using the same set of rules for all. The disadvantage to using weighted calculations is that calculations in the neighborhood are based on weighted averages of the sums within the neighborhood, not the specific locations of live cells within the neighborhood, so the results, while interesting for some things, are generic, and not specific to the particular configuration being examined. Unweighted calculations consider only the specific experimental configuration currently being used. What do we gain from these choices? Weighted calculations simplify the rules for larger radius CAs, and provide an overview of sorts of what is possible with just the right configuration. (They also provide a smoothing effect on CAs with $radius > 1$). The unweighted calculations allow us to deconstruct the rules that lead to certain behaviors, i.e., to examine changes in pattern caused by small changes in single variables.

Initially, we chose a variety of initial seed patterns (5) to include: 1. a horizontal line 7 squares long; 2. a 4x4 hollow square; 3. a corner 4 wide x 4 long; 4. a diagonal checkerboard 3x3; 5. a 5x5 cross over a 3x3 square (4 rotated corner patterns together) to use with a weighted calculation. The sixth pattern, consisting of a filled square with adjustable dimensions was created to provide a a sufficiently saturated pattern

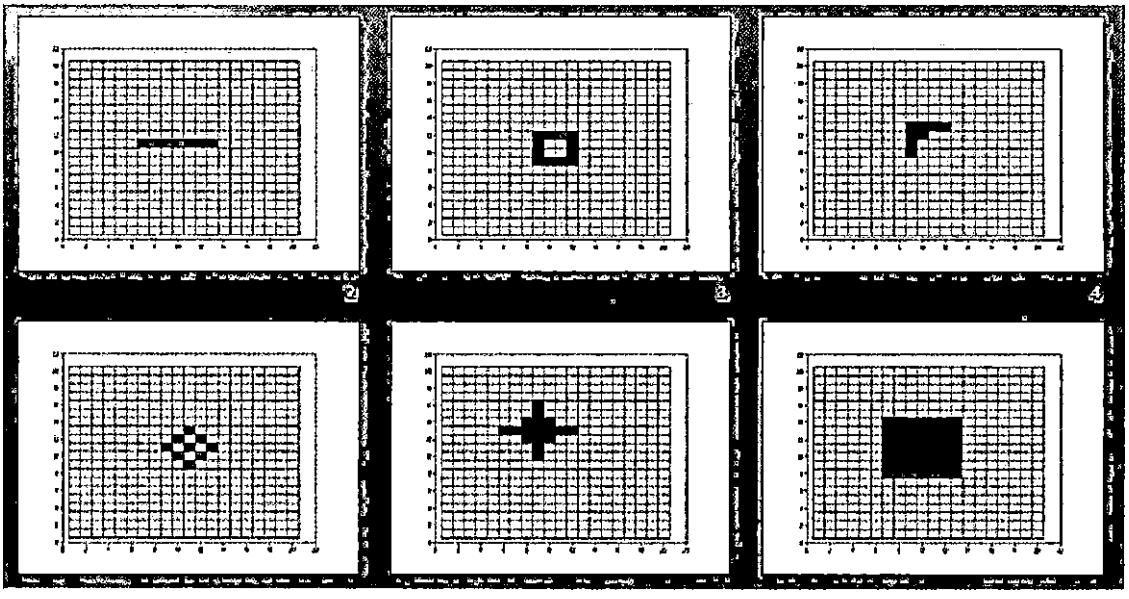

Fig. 2.2: 6 initial seed patterns: line, hollow 4×4 square, corner, 3×3 checkerboard, 7×7 cross and 3×3 square at center, 7×7 filled square

to start CAs with sparse rules. These patterns are shown in figure 2.2.

From a programming standpoint, we chose to use synchronous totalistic CAs, which are updated all at one time and based on the sum of the values in the squares around the target square. This decisions influenced the decision we made to use L-systems for the grammar that we present in chapter, On Birth and Rules. The examination and calculations for each square in the grid are done one at time and the updated values are placed on a temporary grid until the entire original grid is completed. No square, during the examination and updating process, is aware of any other square, outside its neighborhood, and no square is aware of any of the updates until they are all updated together. At this point, the two grids are swapped, and the process begins again. This continues until the predetermined number of time cycles

is completed. The complete Scilab code used for the research done in this thesis is in appendix, Scilab Code.

A decision had to be made about how the edges of the grid would be handled. The alternatives were: a cylindrical wrap, a toroid wrap, or no wrap at all. The patterns that we model, on cave walls, are bounded on all sides (by ceiling, floor, and rocky outcrops. With this in mind, for simplicity, we elected not to use wrap-around on grid calculations. To implement this in the program, a border (equal to the radius being calculated) was not examined in the calculation process. This eliminated unrepresentative artifacts in the pattern edges caused by the edge squares not having 8 surrounding squares.

The birth rules we chose to use with the weighted calculations were designed to work in radius=1 (which include the 8 adjacent squares around the target square). This is often referred to as the cell's neighborhood. (In particular, this is called a Moore Neighborhood. [35]) The values we chose were from $2-4$. The death rules we selected were a range, from death value, d, to 8. These were usually expressed as death on values $> d-1$. The values we selected were > 3, > 4, > 5, and > 6. We will discuss the effects of the birth in most of the rest of the thesis. Death rules are discussed in chapter, Fractal Cellular Automata.

We were the unsure of the importance of wider range calculations in our modeling, so we systematically tested our birth and death rules over radii from 1 to 6. This involved making a decision to vary either the rules to fit the number of additional squares in wider radius calculations, or weighting the sums of all the squares to provide values equivalent to radius 1, so that we could hold the original birth and death rules

constant.

Ultimately, after surveying the initial results we settled on an approach that allowed us to limit variables to a minimalist set that included only:

1. *radius* = 1 initially, and in the later part of chapter 3, and in chapter 4, expanding to include *radius* = 2, and *radius* = 3
2. *birth* = 1 initially, and later expanding to include *birth* = 2, and *birth* = 3
3. *death* == 0 initially, expanding in chapter 3 to include some fractal patterns with $death > 1, death > 2$, and $death > 3$.
4. *initial start pattern*

2.3 Methods of Calculation

Weighting was only necessary for radii greater than one. The method used to do the weighted calculations had an effect on the final patterns. In order to compare the results of the larger radii to the *radius* = 1 results, it was necessary to use a neutrally weighted calculation method. This method treated sums of all radii equally. This was done by dividing the final total of all the surrounding squares (at all radii being calculated) by the total number of squares inspected for that total, multiplying by 8 (to simulate the 8 squares in a radius 1 CA), and rounding to the nearest integer. Doing this provided us with a basis to compare results from various radius calculations with the same birth and death rules to each other.

Once the patterns were systematically calculated for each start pattern, by applying each of the birth rules as specified, over each of the death rules, we inspected the

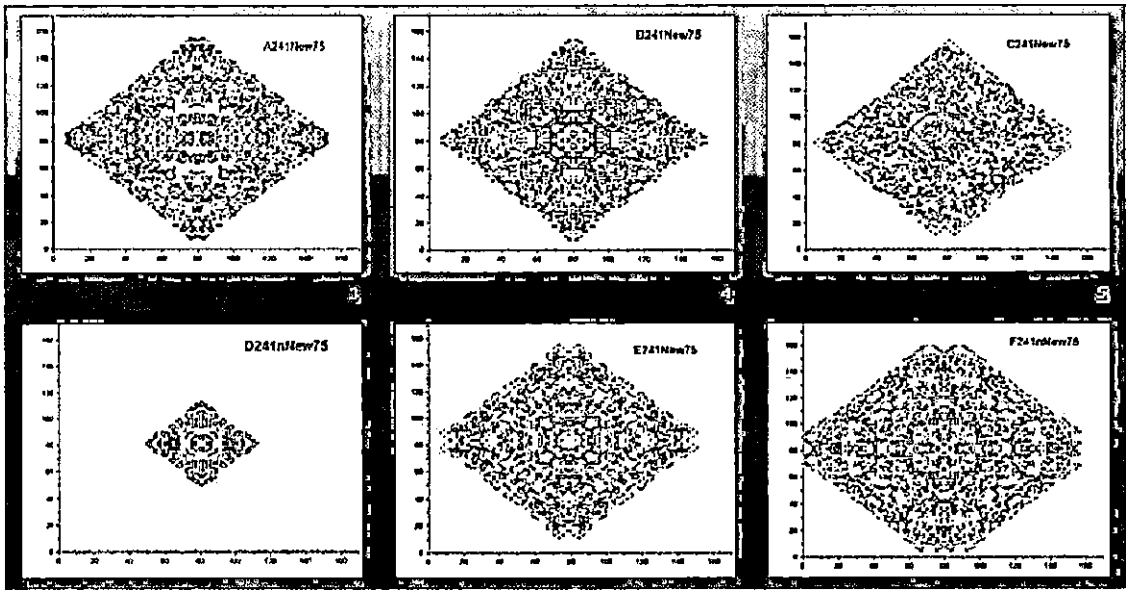

Fig. 2.3: The 6 different seed patterns: line, hollow 4×4 square, corner, checkerboard (3×3), 7×7 cross with a 3×3 square at the center, 7×7 filled square, all with $radius = 1$

results. For $radius = 1$, the general result was patterns that were strongly geometric and reflected the starting seed pattern. The square seed pattern was used to see how the self organizing patterns developed in an initial condition that didn't restrict resources.

The patterns shown in figure 2.3 were calculated with a birth number of 2, a death number of 4, and the calculation radius is 1. All of them were calculated with 75 iterations.

For radius 2, patterns continued to be geometric and reflect the starting seed pattern, but not as strongly as radius 1.

The patterns shown in figure 2.4 are all using the same filled square pattern. We are using the same birth and death values used in figure 2.3, birth = 2, death = 4,

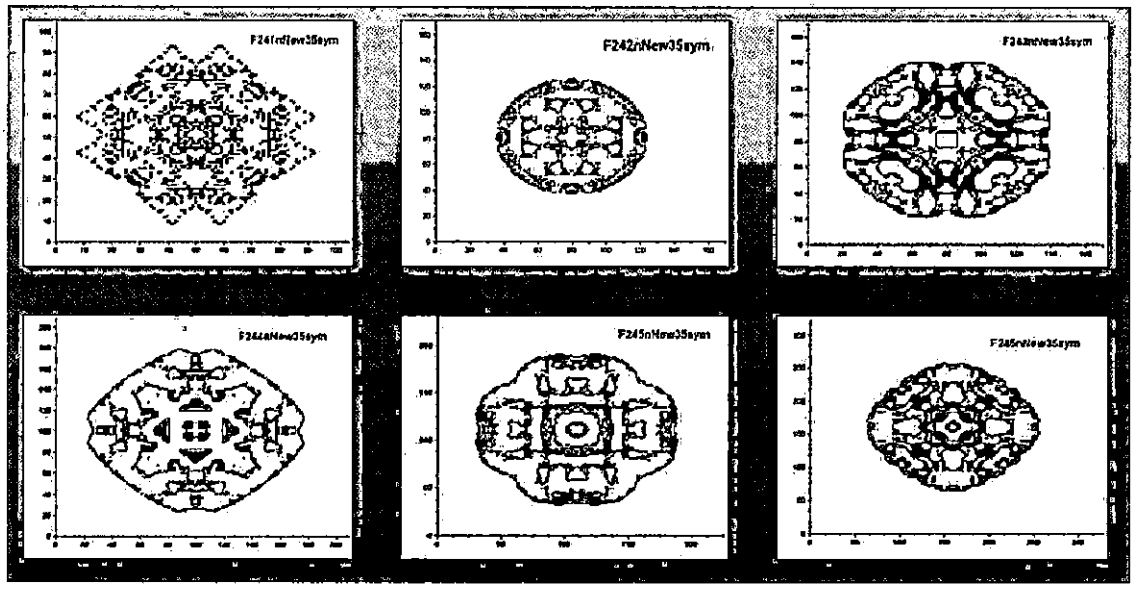

Fig. 2.4: Filled square seed patterns ($birth = 2$, $death = 4$), with $radius = 1$, on the upper left, increasing to $radius = 6$, on the lower right

but the variable here is the the radius. The calculation radius for this set of patterns increases incrementally from 1 to 6.

Radius = 3 patterns tend to chaotic behavior with an emerging self-organizing behavior that occurs with changes in death rules. It is worth pointing out that, in figure 2.4, the residual effect of the filled start square can be seen at the center of all of the patterns. Each of them has a square and/or a cross at the very center of the pattern, and has four axes of symmetry. At $radius = 4, 5, and 6$, the patterns in figure 2.4 had a familial resemblance to the *radius* = 3 pattern shown in figure 2.4 that appears as a progressively more rounded central cross and square motif.

One of the things that we observed was that the closer the death value was to the birth value, the more time cycles the center of the developing pattern oscillates before stability (in the center) is achieved. Once the center of the patterns reach a stable state, most of growth is on or around the perimeter of the pattern.

2.4 Pattern Distortion

We investigated how much it took to distort the patterns by selecting one square in radius 1 (the upper right hand corner - $(i + 3, j + 3)$), and subtracting it from the neighborhood total, and for comparison, adding it into the neighborhood total twice. One square was enough to distort the results completely within 25 iterations. The interesting result was that all of the resulting patterns looked very much alike.

The diagonal line start was not right/left symmetrical in the perturbed patterns, unlike the line and square starts, which were right/left symmetrical. The diagonal start, itself, does not have right/left mirror symmetry. Interestingly, the diagonal line

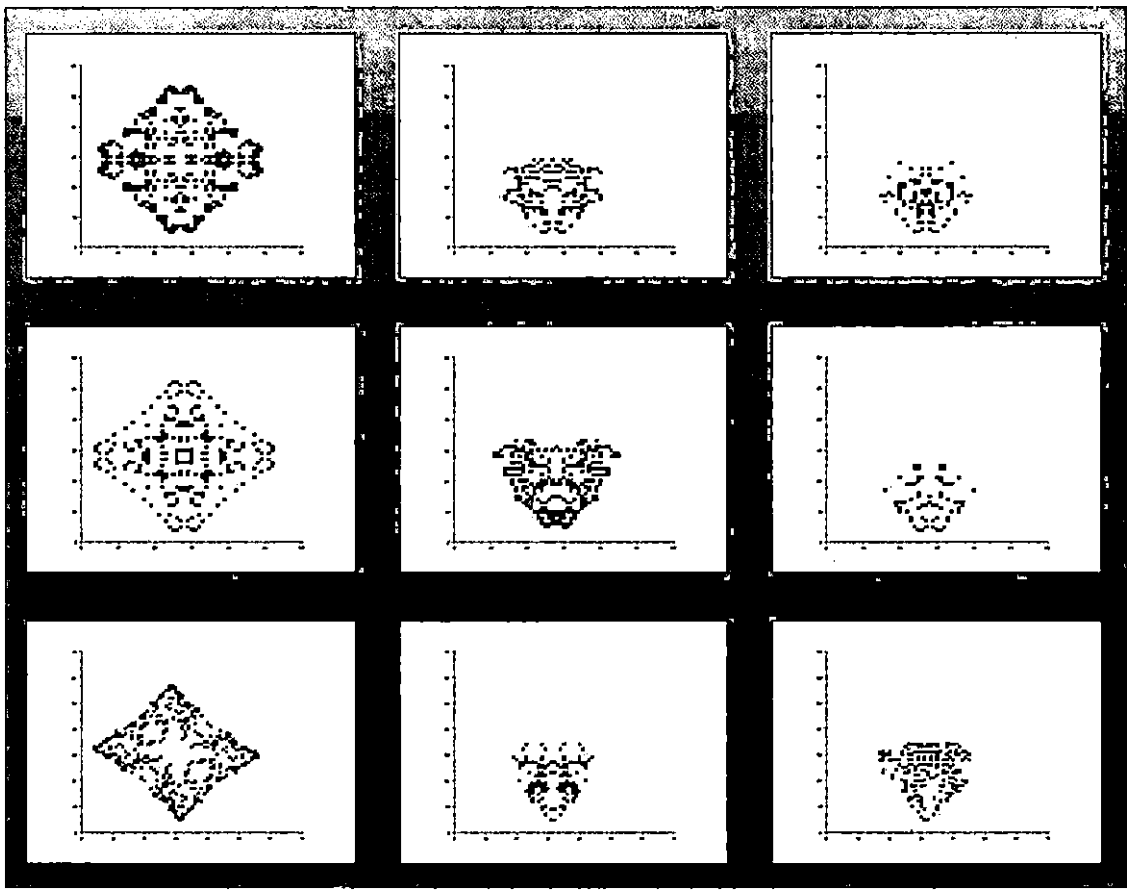

Fig. 2.5: Radius = 1 patterns: row 1:line, row 2:square, row 3:diagonal line; column 1: unperturbed, column 2: upper RH square added twice, column 3: upper RH square subtracted, t=25

start, unperturbed, has upper left/lower right, and lower left/upper right symmetry, but the distortion caused by adding an additional upper right corner, or subtracting it, distorts the symmetry altogether.

The result of perturbing the *radius* = 1 patterns raised the question, "Would a larger radius be more perturbed, or less perturbed (due to a larger number of squares in a larger radius neighborhood)?" We chose to use the same patterns in *radius* = 3, which has a 7×7 neighborhood, to compare to the results we observed in the *radius* = 1, 3×3 neighborhood. The *radius* = 1 has $9-1 = 8$ squares (we are not using the value in the center square to calculate the sum for the rules). By comparison the *radius* = 3 neighborhood has $49-1 = 48$ squares. The perturbation as a function of the calculation sum for *radius* = 1 is $1 \div 7 = 0.1486$ for the *radius* = 1 patterns perturbed by a missing square. For the the *radius* = 1 patterns perturbed by adding a square, the perturbation would be $1 \div 9 = 0.1111$. In *radius* = 3 patterns, the perturbation caused by a missing square, would be $1 \div 47 = 0.0213$, and that caused by adding an extra square would be $1 \div 49 = 0.0205$.

The result of this experiment (see figure 2.6) shows considerable distortion in *radius* = 3 patterns, but because the overall patterns are larger, the apparent visual distortion could be, partially, a result of the larger patterns, which are, in turn due to the larger radius of calculation. The *radius* = 3 perturbed patterns also look very much alike, and the same symmetry issues we observed in the *radius* = 1 patterns exist here, as well. The bottoms of the patterns are less disturbed than the tops because we were adding or subtracting an upper corner, and that corner of the calculation neighborhood was never down at the bottom of the pattern. The

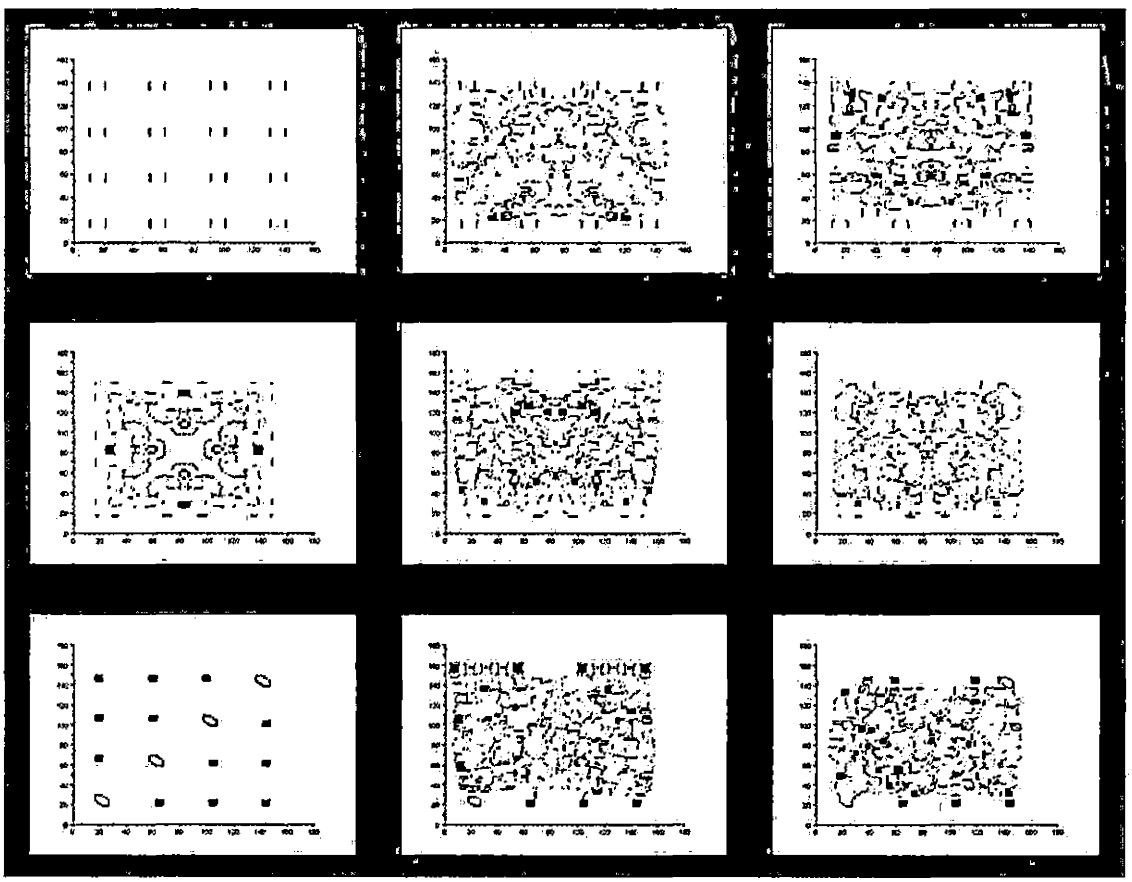

Fig. 2.6: Radius = 3 patterns: row 1:line, row 2:square, and row 3:diagonal line; column 1: unperturbed, column 2: upper RH square added twice, column 3: upper RH square subtracted, t=25

distortion seen at the bottom of the pattern is the result of a trickle-down effect of the having the upper pattern distorted and calculating repeatedly over the distortion in course of twenty five time steps.

The most interesting results from this are:

1. There is a strong resemblance among the $radius = 1$ patterns with different starting patterns. This leads us to believe that the effect of the radius on the overall developing pattern, at $radius = 1$ at least, is very strong.

2. At $radius = 3$ and above, with the same set of birth and death rules or values, the resulting patterns have a familial resemblance. The implications of this, for modeling biological growth, are that we only need to model for $radius = 1$, $radius = 2$, or $radius = 3$ spaces.

3. The distorted patterns more closely resemble what we have seen in the images taken in caves. This may be because perfect symmetry in seed patterns in biomats is unlikely in the extreme, and these model patterns are easily distorted. They also may serve in models as a method for including geological features that channel extra nutrients to some areas, or restrict the flow of nutrients to particular area. Another potential use, outside of the caves, is modeling asymmetrical root growth due to any number of causes, which could include rocky soil, roots due to animal predation, lack of water to part of the root ball, etc.

In the next three chapters we examine the results of using an unweighted calculation and initially holding the radius constant at 1 (a 3×3 neighborhood grid), and eliminating death from the calculations. This essentially leaves the birth value

as the only variable. We have also chosen to target the first 1 to 15 time steps as our primary area of interest for these CAs. As our understanding grew, we added or changed variables one at a time in an effort to identify the variable (or variables) that were responsible for each behavior in the resulting CA.

3. FRACTAL CELLULAR AUTOMATA

3.1 Introduction

Cellular automata can be as delicate and stunningly beautiful (and complex) as those shown in figure 3.1. The top row and the bottom are the same automata. The bottom row shows residual growth in green, new growth in blue, and death in this time step in red. What drives the beauty and complexity that derive from such simple rules? In this chapter we are going to take a minimalist's view of cellular automata to try and determine some of the mechanism that creates patterns like the one in figure 3.1 . The only way we can understand the underlying framework of two dimensional CAs is to look at them in their most elementary form. What are the fewest number of variables? How do they behave? What happens when we throw away crowding, throw away death, and start with birth= 1, radius= 1, and build on that?

We are going to start by covering the behavior of three prototypical seed starts. Using this approach we will begin with birth= 1, no death (minimal case), and discuss their habits of growth.

With a solid view of what happens with birth= 1, radius= 1 and no death, we will begin adding other variables. We will continue holding birth and radius equal to 1, and add death to the mixture, discussing the results. Then we will step back to radius= 1, birth= 1, no death, and introduce the effects of larger radii, which will

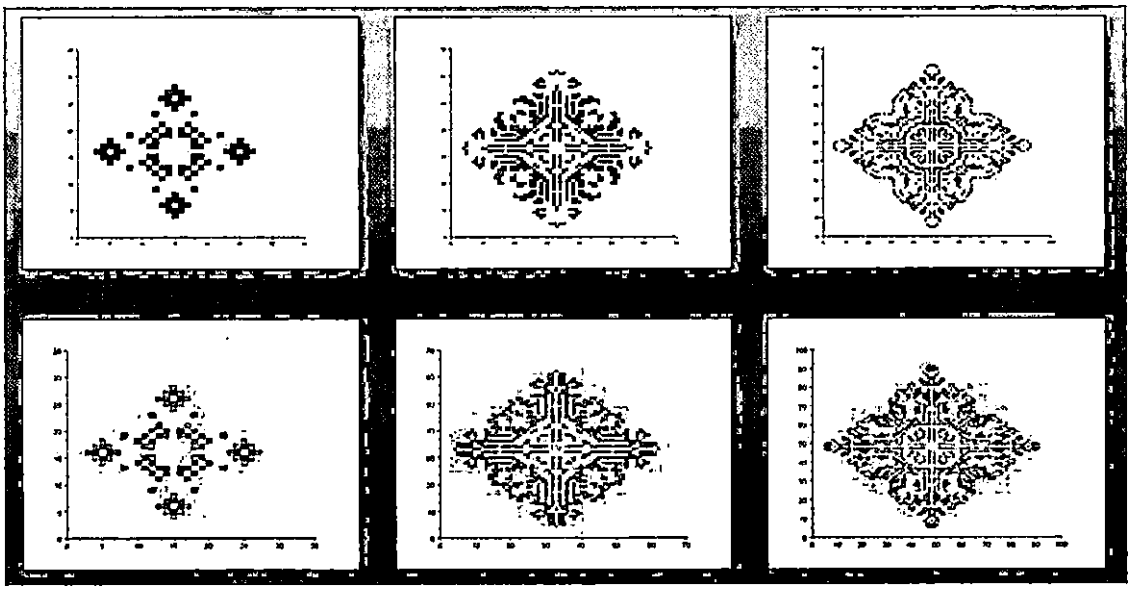

Fig. 3.1: *Birth* = 3, *death* = 5, *radius* = 1, upper row and lower row the same images - top row: black and white, bottom row: green = growth from previous time step, blue = new growth in this time step, red = death in this time step, columns (L - R): $t = 24, 68, 100$

introduce the scaling seed patterns to the neighborhood or radius size as variable.

This whole journey is to try and determine which initial conditions in a simple CA are key variables, and how they affect the developing pattern. The next chapter on the effect of birth rules will attempt to codify what we observe here into a series of grammatical statements.

In this chapter, the variables are independent of each other, but in the chapter on the effect of scaling on developing patterns, we will discuss what happens when a group of variables become interdependent, and produce a family of twenty nearly identical patterns across a range of seven different radii, and an equal number of birth numbers. We will look into how these variables work together to produce this particular pattern, and what the effects of variables are that make this possible.

3.2 *Radius*= 1, *Birth*= 1, *No Death*

The simplest case for the cellular automata that we have studied is radius= 1, birth on 1, no death. The seed patterns that we chose to study are: a horizontal line 7 squares long; a 7×7 filled square; a single square; and a diagonal line. These patterns are of additional interest because the line has 2 axes of symmetry, the square and the single square have a 4-way symmetry, and the diagonal line, has 2-way symmetry (like the line, but rotated 45 degrees). (Other patterns, like a triangle, have only one axis of symmetry.) The patterns that develop from these starts preserve the original symmetry of their seed patterns.

3.2.1 Observations

Horizontal Line

In the case of the line, with a *birth* = 1 (shown in figure 3.2), there is only one way that the pattern can grow. It must grow at the end points of the original line, and because we are calculating in radius= 1, the lines can grow to a maximum of three squares long, as there are only three positions of neighborhood that include the endpoint, and do not include any other part of the line (since birth only takes place with one neighbor only). At the end of time step 1 we have the seven unit line, with two 3-unit lines– one attached vertically to each end of the original horizontal line. There are no more endpoints on the original line, but there are 2 endpoints on each of the 2 new lines. On the next time step (t= 2), the pattern grows four 3-unit, horizontal lines, one at each endpoint of the 2 vertical lines we grew in the last time step. Again, there are no more endpoints to grow from this time, but there are eight end points to grow from on the next time step. In the next time step (t= 3), the pattern grows four 3-unit vertical lines on the outer ends of the 8 end points we created last time. We begin to see some crowding in this time step (t= 3) which causes the inside lines to only be 2 units long. There is a growth pattern developing here:

1. The number of growth points is doubling with every time step, and
2. Growth is alternating between vertical and horizontal lines on successive time steps. In point of fact, on odd numbered time steps, the pattern only grows vertical lines. On even numbered time steps, the pattern only grew horizontal

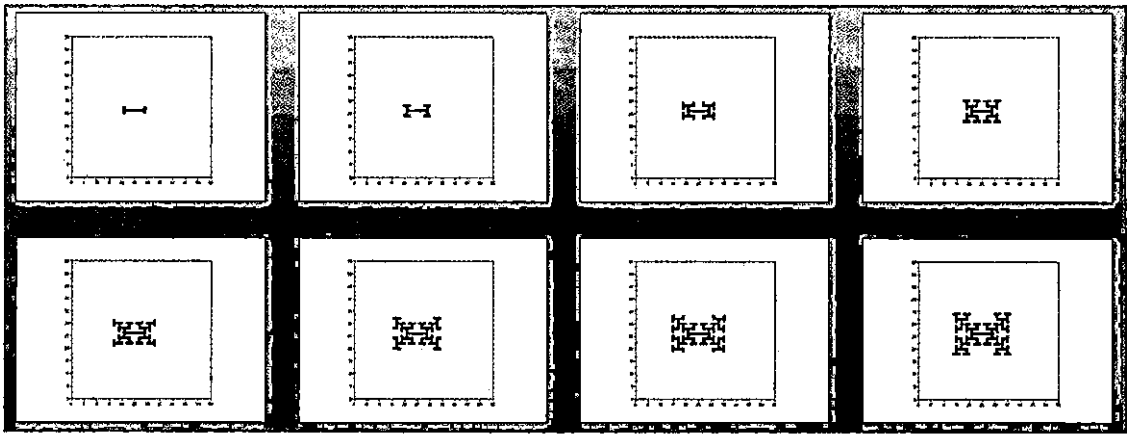

Fig. 3.2: Fractal 7-unit line growth, *birth* = 1, no death, *radius* = 1, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

lines.

On the fourth time step, we develop a filled square with four horizontal lines on top and four on the bottom. On the next time step, there are only four places it can grow (on the outside of the pattern). Figure 3.3 shows that this is a recurring cycle. The pattern develops in this binary fashion, first vertically, then horizontally until the only end points that can grow are on the corners. this cycle increases in a predictable way producing filled patterns on the 4th, 12th, 28th, 60th, 124th (and so on...) time steps, with 4, 8, 16, 32, 64 lines top and bottom respectively. At each filled pattern it takes twice as many time cycles to reach the next filled pattern.

Note that the 2-way symmetry we began with in the 7 unit line persists in the developing pattern. It is also worth noting that the measure of computation speed developed by A. R. Smith [47] could be used here - the change in perimeter with respect to the number of time steps.

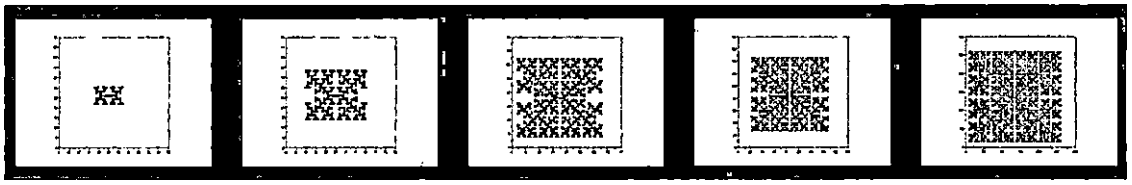

Fig. 3.3: Fractal 7-unit line growth complete Squares, $birth = 1$, no death, $radius = 1$, $t = 4, 12, 28, 60, 124$

Filled Square / Single Square

In figure 3.4, with the 7×7 filled square as a seed pattern, growth can't begin on endpoints. There is another way that birth with only one neighbor can grow in a 9 square Moore Neighborhood. Since we are looking at birth= 1, with no death, and a calculation radius= 1, the filled center of the square will remain. Without endpoints, the only place these rules will allow growth is a single point diagonally out from the corners of the square. In time step (t= 2), this develops into new corners at each corner of the original square. In the next time step these corners each provide 2 end points that can grow, in addition to the new points diagonally out from these corners, themselves. In the fourth time step the diagonal points from the previous time step develop into corners, again, and each of the lines that grew at the end points of the previous corners now grows a 3-unit line. By the end of the this time step the pattern has become a 4×4 filled pattern with no end points to grow from (a larger version of the original pattern), leaving only the corners as growth points, in time step 5.

The filled squares in this sequence(see figure 3.5) also occur in the same recursive cycle as the pattern that began with a line. Additionally, due to the 4-way symmetry, it is easy to see that the size of the filled square is the same as the number of time

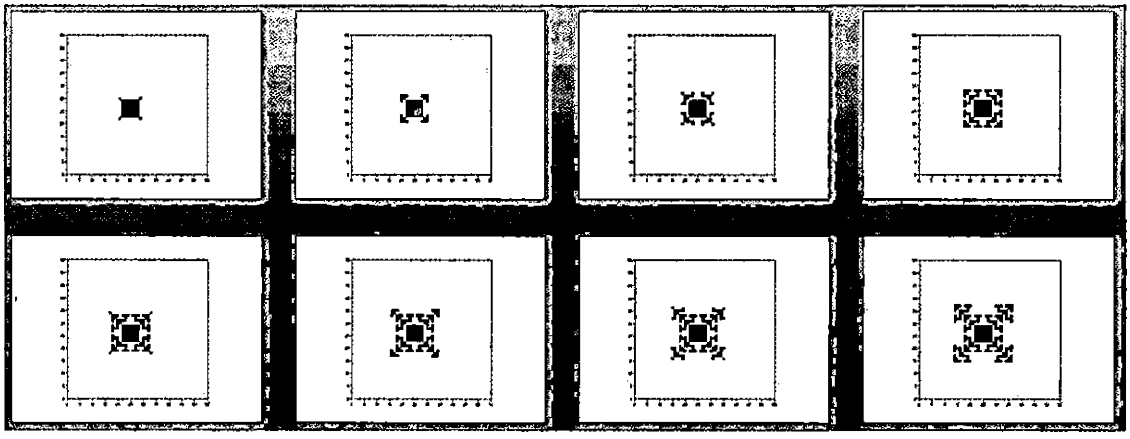

Fig. 3.4: Fractal 7×7 square growth, *birth* $= 1$, no death, *radius* $= 1$, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

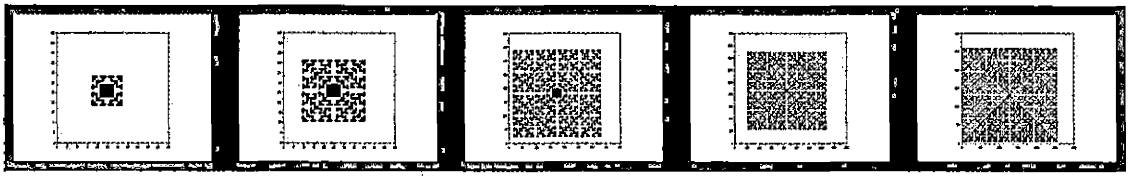

Fig. 3.5: Fractal 7×7 square growth complete squares, *birth* $= 1$, no death, *radius* $= 1$, $t = 4, 12, 28, 60, 124$

steps in the cycle that it completes. For t= 4, the filled square is 4×4. In t= 12, which took 8 $(12 - 4)$ cycles to complete the resulting filled square is 8×8. The filled square at t= 28 is 16×16, and doing the math $(28 - 12 = 16)$, it took 16 time steps to develop. Each successive cycle takes twice as long to complete.

Another interesting observation is that where the pattern that develops from the line grows in alternating vertical and horizontal growth in odd and even time steps, this pattern alternates lines perpendicular to the sides of the original square, with lines parallel to its sides of the original square. Additionally, we observe that the 4-

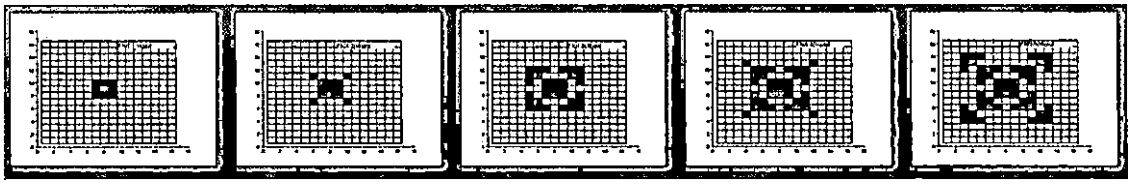

Fig. 3.6: Single square growth, *birth* = 1, no death, *radius* = 1, $t = 1 - 5$

way symmetry in the initial seed pattern is preserved as a result of the corner growth from the filled patterns.

Single square starts in radius= 1, birth= 1, and no death, grow to a 3×3 square in the next time step, and proceed to grow following the patterns above described for the filled square. Figure 3.6 is shown in color to clearly show the single square start in the first frame (in green). Note that the 3×3 square is the size of the Moore neighborhood at radius= 1. This is a function of radius of calculation being used. Think of a card with a square aperture the exact size of the radius 1 neighborhood. Then think of placing the card in various positions that include the single square start. For each of these positions the square in the center of the card would change from empty (0) to 1. There are eight of these surrounding the single square, which causes the square to grow from a single square at t=0, to a 3×3 square at t= 1. This can only take place if there are no other squares appearing in the 3×3 aperture that we use to examine the pattern.

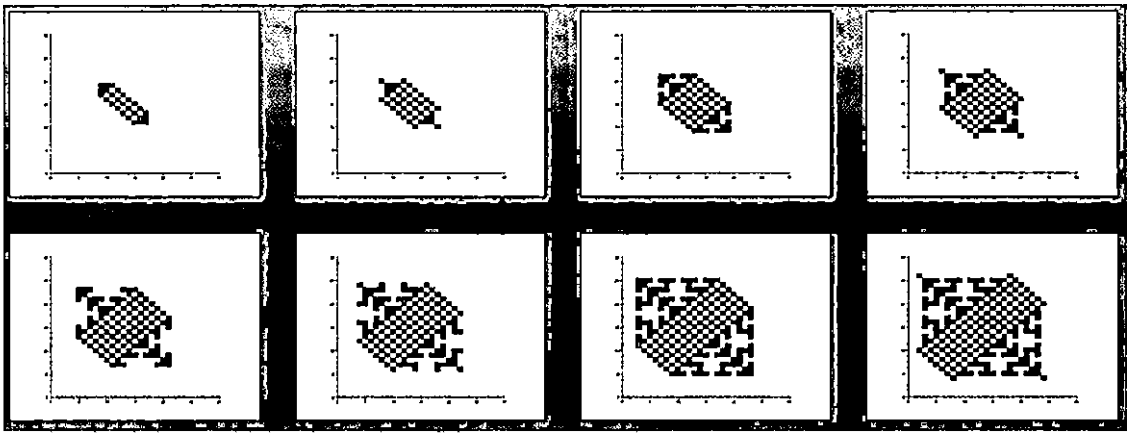

Fig. 3.7: 7-unit diagonal line growth, *birth* = 1, no death, *radius* = 1, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

Diagonal Lines

Another pattern of growth in radius= 1, birth= 1, no death, is diagonal lines. We have observed that a diagonal line grows parallel to itself and continues on in the same direction, with diagonal lines of the same width. It generates one diagonal line at each time step. Figure 3.7 shows the first eight time steps of pattern development that begins with a 7-unit diagonal line. The first corner forms off of the endpoints of the line in the very first time step. The sequence shows that the end corners alternate between new corners and a single diagonal square off the corner formed in the previous (odd numbered) time step. By time step 6, the sequence of corners has gotten far enough away from the band of parallel diagonal lines that there is enough open space around the latest corner's endpoints to permit lines to grow there for the first time.

3.3 Conclusions for Radius= 1, Birth= 1

3.3.1 Habits of Growth

There are only 4 ways that radius= 1, birth= 1 patterns can grow:

1. They can grow off the endpoints of lines, and if not crowded, the lines will be 3 squares long (the length of one side of the *radius* = 1 Moore neighborhood);
2. They can grow diagonally, with a single square, at each of the corners of the square, and in the next time step, they will form another corner;
3. They can grow from a single square to a 3×3 square (the size of the calculation neighborhood);
4. They can grow from a diagonal line, parallel and equal in length to the original line.

Can we reconcile the alternating vertical and then horizontal growth of line starts with the square start's single square diagonally out from the corners of the starting square, and alternating growth that is parallel to the sides of the square and then perpendicular to the sides of the square? Is there a simple rule that will cover all of these?

Line

The line growth appears to alternate between horizontal and vertical growth. But, this is really perpendicular growth at the endpoints of the lines grown in the previous time step.

Filled Square / Single Square

The single square start is really a subset of the filled starting square above. We can break it down into 2 sets of squares, the diagonal squares we discussed above, and those squares that are orthogonal to the sides of the original single square (like a von Neumann neighborhood). Clearly the orthogonal squares are perpendicular to the sides of the original square, and clearly the diagonal corners are perpendicular to both sides of the original square. Once we get past time step 1, the growth proceeds exactly like the starting square above, with a 3×3 square.

The growth in the 7×7 starting square appears to be a bit more complicated, but if we look at it carefully, the single square at each corner is the only point that is perpendicular to both sides that form the corner. The next time step has two 3-unit lines joining to form a corner, parallel to the original corner of the start-square, and one row beyond the single square. If the single square is perpendicular to both sides of the start-square that forms the corner, then the corner that is formed parallel to the original starting corner, must be perpendicular to the single square that formed in the previous time step. So this growth is also perpendicular to growth in the previous time step.

Diagonal Line

The diagonal line's growth is parallel to the original line with one row for each time step. If the line is a single line, not part of a larger pattern, two things will occur. The growth will occur on both sides of the diagonal line, like the growth at the endpoints of the line start, and the end points of the diagonal line will begin to develop as corners.

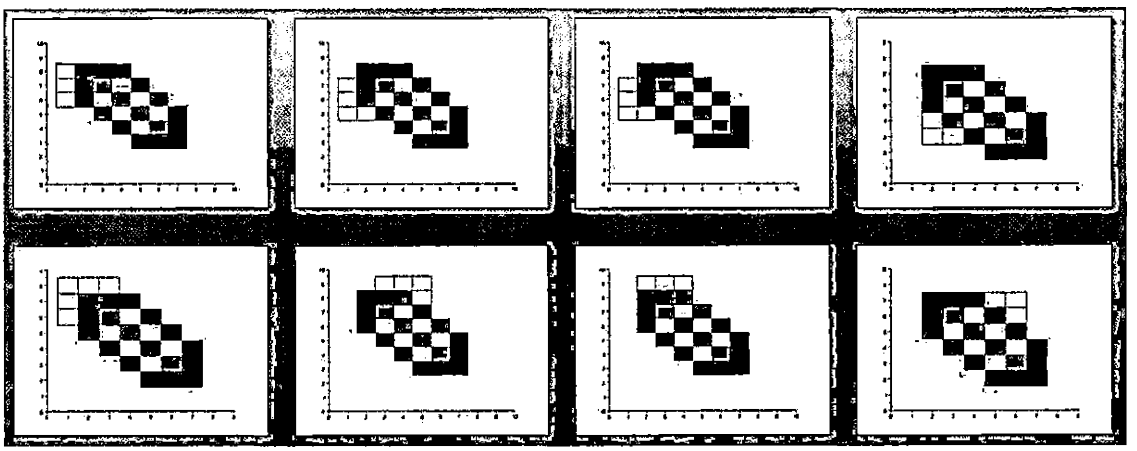

Fig. 3.8: Parallel growth in diagonal lines, *birth* = 1, no death, *radius* = 1, neighborhood shown, with target square in red; previous time step in green, current time step in blue

As we mentioned earlier, the symmetry of the original pattern will be preserved.

What makes the line, and the square patterns all grow in the same way while the diagonal line grows in a different manner? All of our patterns are being developed on an n×m rectangular grid, where the x and y axes are perpendicular to each other. The diagonal line start is the only pattern that grows parallel to the original pattern, and it is not along the x and y axes like the others.

If we return to the idea of the card with an aperture exactly 3 squares × 3 squares (the size of our calculation neighborhood at radius= 1), we will find that the center of the neighborhood that sees only one square is up one row and over one column from the square, on the start line, that the neighborhood sees. Up one row, and over one column, for each square in the entire diagonal line, forces the resulting line to be both parallel and of equal length. On the backside of the diagonal start the the location is down one square and over one square. (These directions presume that the original

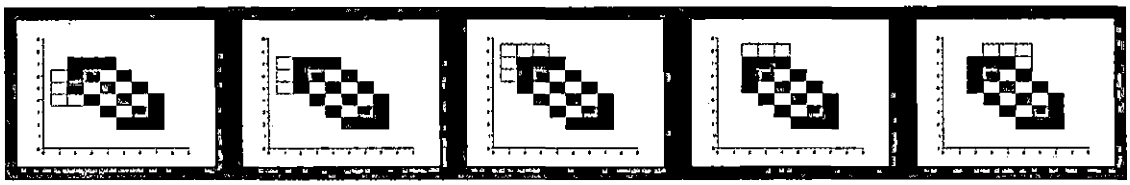

Fig. 3.9: Corner formation in diagonal lines, *birth* = 1, no death, *radius* = 1, with neighborhoods shown (the outside two frames were shown in 3.8)

line is a backward leaning diagonal– forward leaning diagonals will have similar, but not identical patterns of development for their parallel lines.)

In figure 3.8, the green squares are the pattern the neighborhood is using to calculate the next time step and the blue squares represent the results of that calculation. Figure 3.8 shows that the parallel lines that form in the next time step are in fact, the only way that the parallel line start can grow. Figure 3.9 show how the neighborhood forces the growth of a corner off the endpoints of the diagonal line.

These patterns are both recursive and self-similar by the nature of how birth with one neighbor forces them to grow. In the case of the line, the pattern can only grow from end points, growing by powers of 2, where the start line has two endpoints to attach to, and the two lines that attach have four endpoints, and so forth.

A completely filled square occurs at t= 4, t= 12, t= 28... Each cycle takes twice as long to complete, and has twice the perimeter of the previous filled square (remember that perimeter growth as a function of time steps is a measure of calculations speed [47]). Using the perimeter measure vs. number of time steps, we can see that doubling the perimeter takes double the number of time steps. These CAs are growing at constant speed. Once a CA fills in the square in each cycle, only the 4 end

points at the corners are open for growth in the next time step.

In the second case, the filled square, and by extension, the single square, start a pattern that is repeated at the end of each cycle, leaving only the corners available for growth in the next time step. These patterns also double their perimeter in double the number of time steps - so growth is at a constant rate.

This pattern also completes at t= 4, t= 12, t= 28, and so forth, at radius= 1, birth= 1.

The parallel diagonal line have a slightly different sequence. The pattern completes at the very first time step (with the corner that forms from the endpoint of the original line). Thereafter, new, larger patterns are completed at t= 3, t= 7, t= 15, and so forth, leaving only the exposed diagonal lines at two opposite corners of the square pattern, and completed corners at the other two corners of the square. This slight displacement of completed patterns is due to the first time step being complete. The doubling of the perimeter and time steps begins from there. This CA pattern is also growing in constant rate.

All of these variations of the birth with one neighbor, no death, patterns calculated over a single 9 square Moore neighborhood, have fixed geometric growth rules similar to the Sierpinski Triangle and Koch Snowflake, which are both classic, recursive fractal functions. Since each of these patterns returns to a state similar to the start pattern (in the case of the filled square) or the very first time step in the case of the line and the diagonal line) at the end of each of these cycles, the patterns repeat themselves with larger and larger squares that grow in longer and longer cycles all based on powers of 2. These patterns are not just cellular automata, they also meet the recursive,

geometrically driven growth rules that are characteristic of fractal patterns.

3.4 A Little Death Goes a Long Way... or The Effect of Death on Birth= 1, Radius= 1, Patterns

3.4.1 Introduction

We are going to start this section with a little discussion about the figures in this section. The addition of death to the $radius = 1$, $birth = 1$ rules makes the resulting patterns much more difficult to understand. With this in mind, we will be using a three colored palette for many of the figures for clarity. In the colored figures in this section, green represents residual growth left over from the previous time step, blue represents new growth from the current time step, and red represents cells that have died in the current time step. This is important to keep in mind, because *the red squares are not really there.*

Some of the patterns are so dense that they are difficult to view in color, and frequently the denseness is the result of a strong stability in the developing pattern that minimizes the effect of the death rule. Sometimes this stability isolates the effect of the death rule to the outside margins of the pattern, leaving the interior very dense. In cases like this, we will show the patterns in black and white, to best illustrate the effect of the developing pattern.

3.4.2 Line Starts

Radius= 1 line starts with death are fractal, and both simple (birth= 1, death> 1), and very complex. The line start with birth= 1, and death> 2 was the impetus for

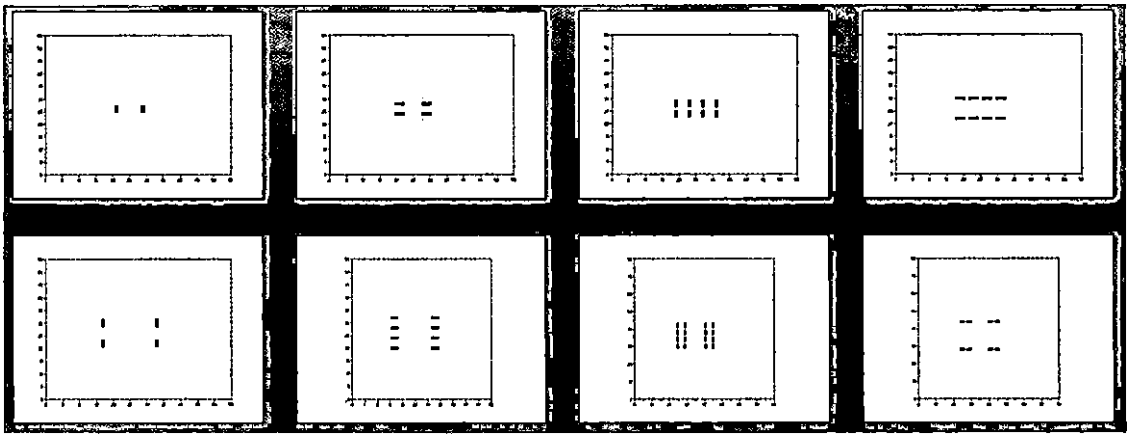

Fig. 3.10: Line start, $birth = 1$, $death > 1$, $radius = 1$, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

the redesigned program (see the appendix for the full SciLab code) that showed the cells that died at each time step in red. The pattern with death> 3 is so dense that it is easier to see in black and white. We will discuss each of them.

The pattern with death> 1, shown in figure 3.10, is very much the line equivalent of the square start with the same birth and death values. It begins with a 3-unit horizontal line which in the first time step has two 3-unit vertical lines attached at the endpoints, and death removes the original line. In time step 2, the 2 vertical lines from the previous step become 4 horizontal lines. The pattern continues to develop in this fashion. This is, in a linear way, the equivalent of the square starts for birth= 1, death> 1 (or > 2, or > 3).

We mentioned earlier that using a 3 color pattern showing growth from the last time step in green, new growth in this time step in blue, and death in this time step in red helped to clarify how some of these patterns develop. Figure 3.11 shows the

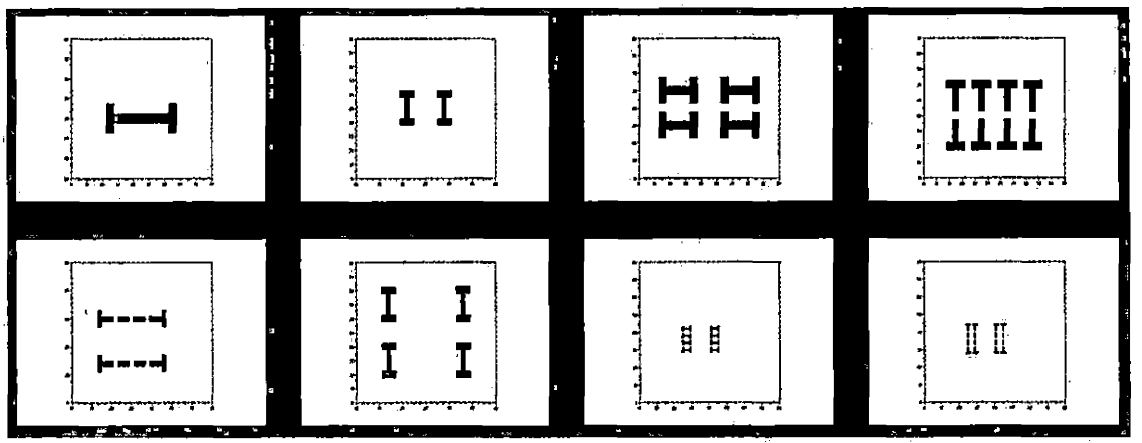

Fig. 3.11: Line start, *birth* = 1, *death* > 1, *radius* = 1, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$; pattern the same as figure 3.10 in color - blue is new growth, green is residual growth from previous time step, red is death in this time step

patterns from figure 3.10 using this technique– just remember *the red squares aren't really there.*

In figure 3.12 the line start pattern that develops from birth= 1, death> 2, at radius= 1 is surprisingly complex from the very beginning.

This pattern was the impetus for the color technique showing death in red. It was difficult to see how the pattern made the transitions from one time step to the next. As an example, compare both the black and white CA (figure 3.12), and the three color version in figure 3.13. It is a cyclic fractal, repeating, with changes in scale.

The pattern that we see with death> 3 is also fractal, cyclic, with changes in scale. It appears at first glance to be deceptively simple (see figure 3.14), but over a greater number of time steps (> 16, at least) it becomes clear that there is a subtle pattern of triangles and squares embedded in the larger pattern (see figure 3.15). This subtle pattern is the result of long (neighborhood+1) and short (neighborhood−1)

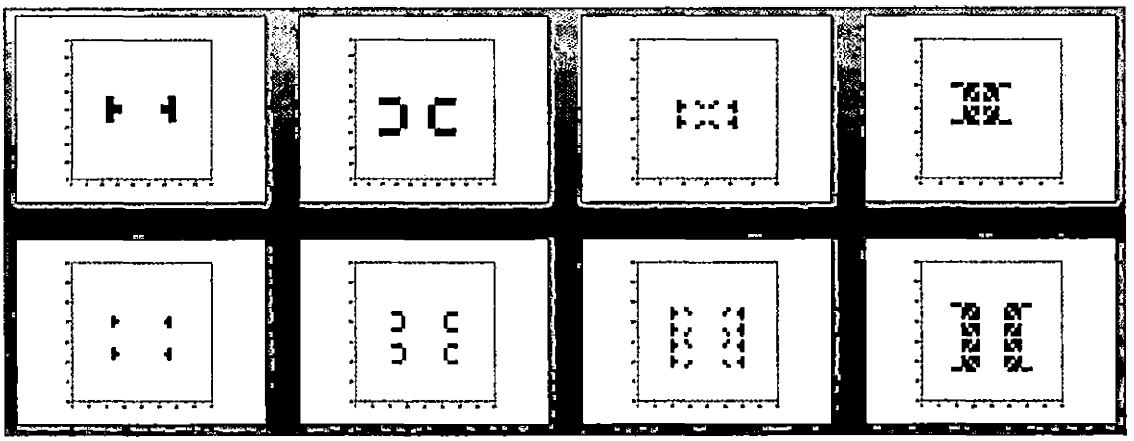

Fig. 3.12: Line start, *birth* = 1, *death* > 2, *radius* = 1, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

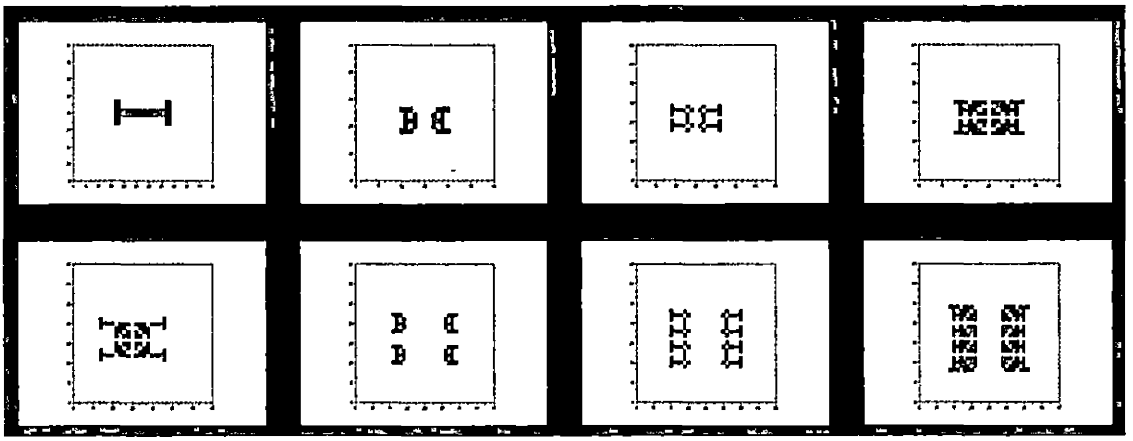

Fig. 3.13: Line start, *birth* = 1, *death* > 2, *radius* = 1, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$ in color - pattern the same as figure 3.10 in color - blue is new growth, green is residual growth from previous time step, red is death in this time step

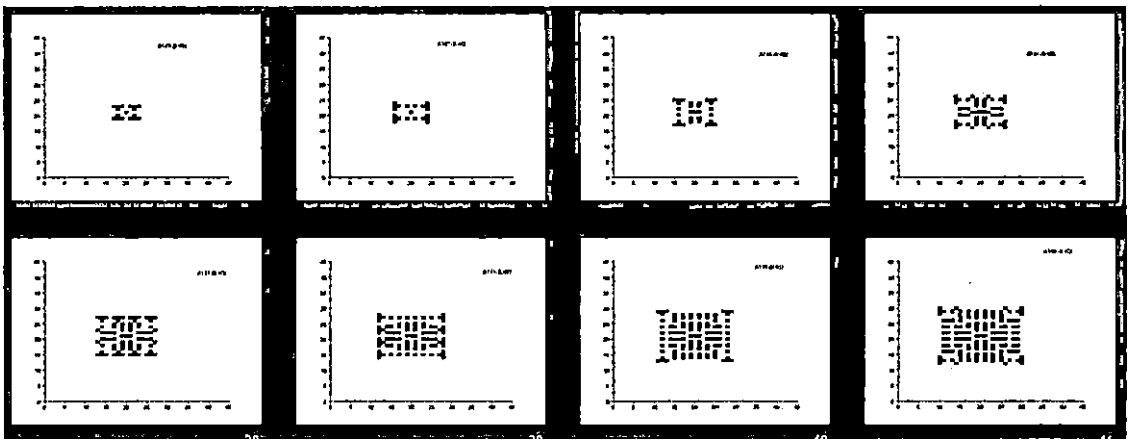

Fig. 3.14: Line start, $birth = 1$, $death > 3$, $radius = 1$, top row: $t = 1 - 4$, bottom row: $t = 5 - 8$

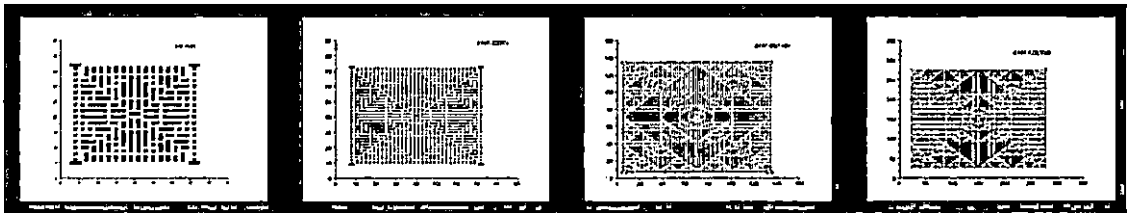

Fig. 3.15: Line start, $birth = 1$, $death > 3$, $radius = 1$, $t = 16, 32, 65, and 127$, shown

lines alternating with single squares to fill in the spaces. We are going to present these images in black and white. Because the pattern is so dense, it is much easier to see the developing pattern in black and white.

3.4.3 Square Starts

Looking at figure 3.16, observe that an interesting thing happens with these square starts - the same pattern develops for death > 1, > 2, and > 3. (Notice, the birth

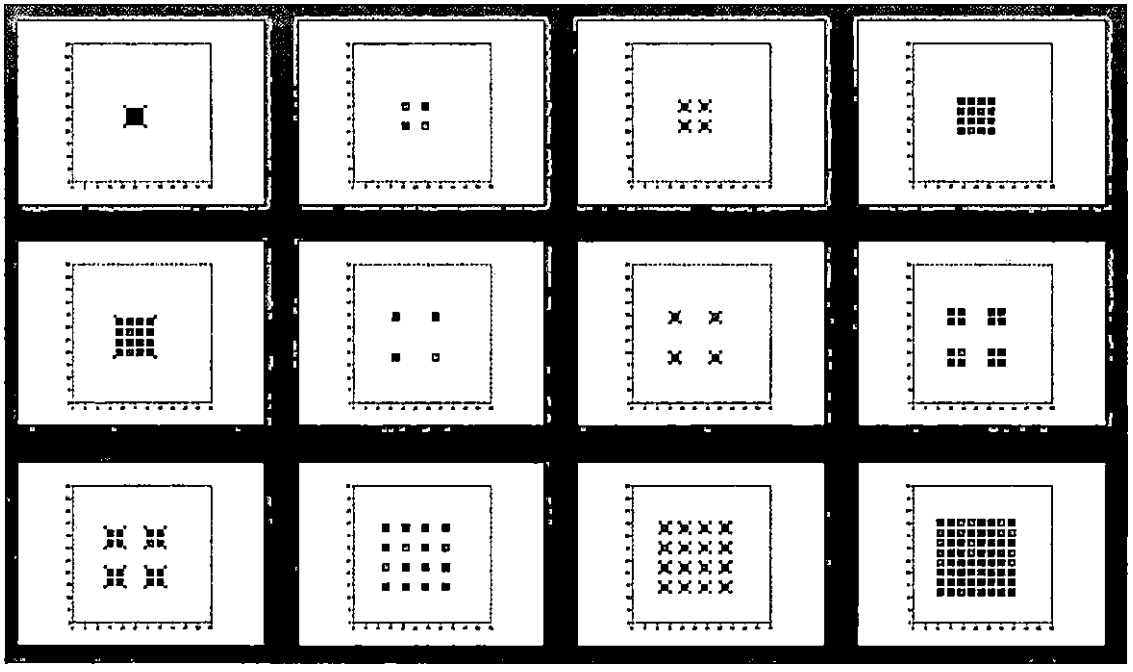

Fig. 3.16: Square start fractal patterns with *death* $> 1, > 2, > 3$, top row: $t = 1 - 4$, middle row: $t = 5 - 8$, bottom row: $t = 9 - 12$

numbers are single values and, for death, we are using a single range of values.) This unexpected, identical result for 3 different, but overlapping ranges of death leads us to the observation that in this case, the single values of death= 2, and death= 3, are irrelevant in this pattern.

Single Square Starts

For single square starts, in the first time cycle, the seed goes from the single square to a square the size of the neighborhood, and the pattern we saw with the 3×3 square seed above develops from there. This has the net result of offsetting the time cycle by one and developing the completed squares that we saw at above at t= 4, 12, 28,

etc., now appear at t= 5, 13, 29, and so forth.

3.4.4 Diagonal Line Starts

Figure 3.17 shows three different death rules sided-by-side so we can compare them. The first column is patterns formed by a *death* > 1 rule. The second column is a *death* > 2 rule and the third is a *death* > 3 rule. Notice that they also form the same pattern with minor differences in the odd numbered time steps. In death > 1, the original line dies off completely at $t = 1$. In death > 2, the original line dies off, except for the endpoints, at t= 1, and in death > 3, the original line does not die off at all in t= 1. By t= 2, however, all of the patterns are identical again. The same thing happens in t= 3, and t= 4. This family of patterns continues to repeat this cycle of divergence at odd time steps, and return to identical patterns on the even time steps.

3.4.5 Conclusions

In these simple starts, the addition of death to the rules seems to emphasize the fractal behavior that we observed before. There are very similar behaviors between the line start with *death* > 1(see figure 3.10 and figure 3.11) and the large square start for *death* $> 1, > 2$, and > 3 (see figure 3.16). In each of these patterns, as they grew new lines, or single squares, the death rule eliminated the original pattern that provided the basis for the new growth. There are corollaries between the diagonal line, with *death* $> 1, > 2$, and > 3 (see figure 3.17), and the square start, with *death* $> 1, > 2$, and > 3 (see figure 3.16) in that the death range did not really matter for either of

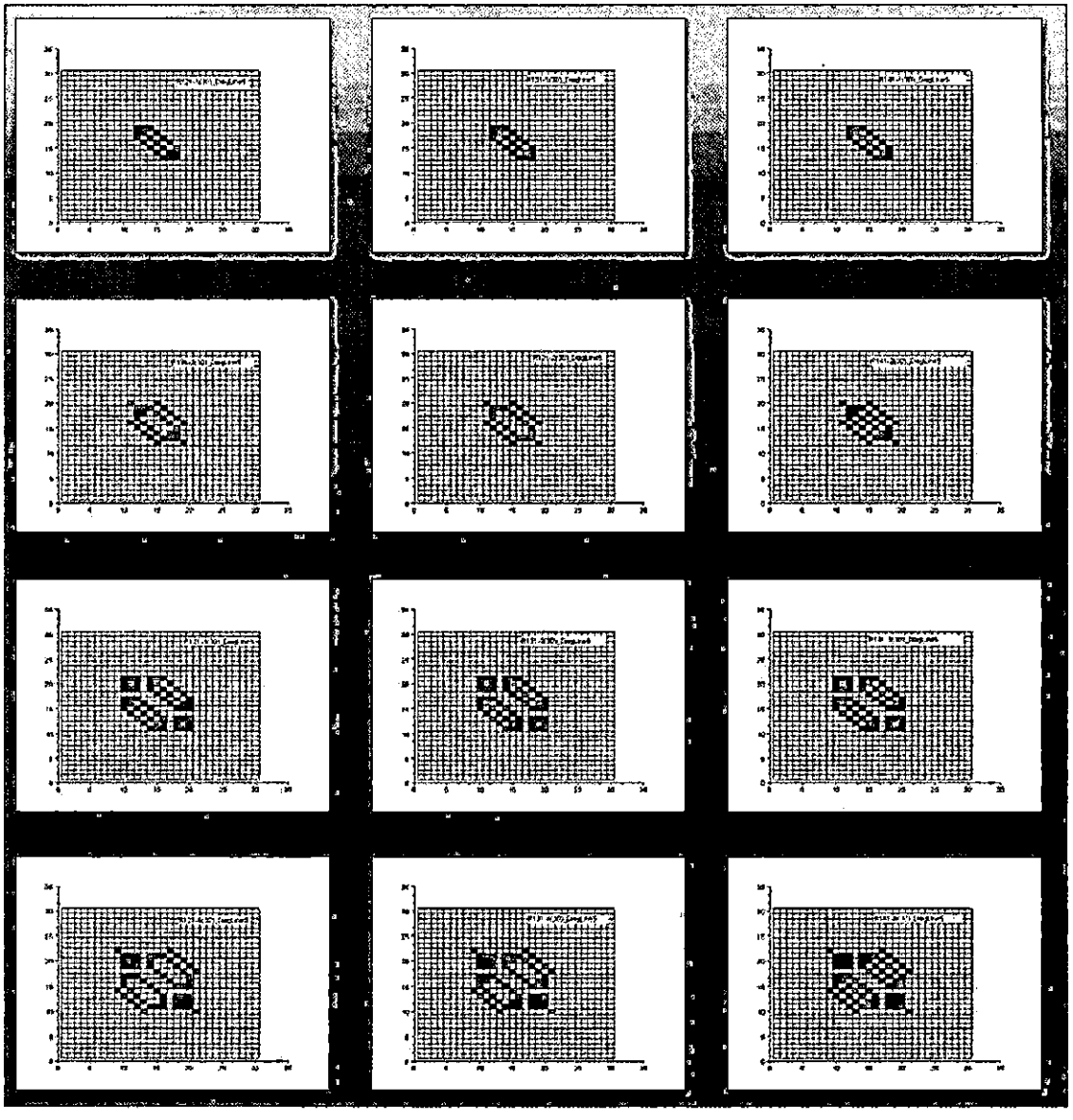

Fig. 3.17: Fractal diagonal line patterns, column 1: $death > 1$, column 2: $death > 2$, column 3: $death > 3$, row 1: $t = 1$, row 2: $t = 2$, row 3: $t = 3$, row 4: $t = 4$

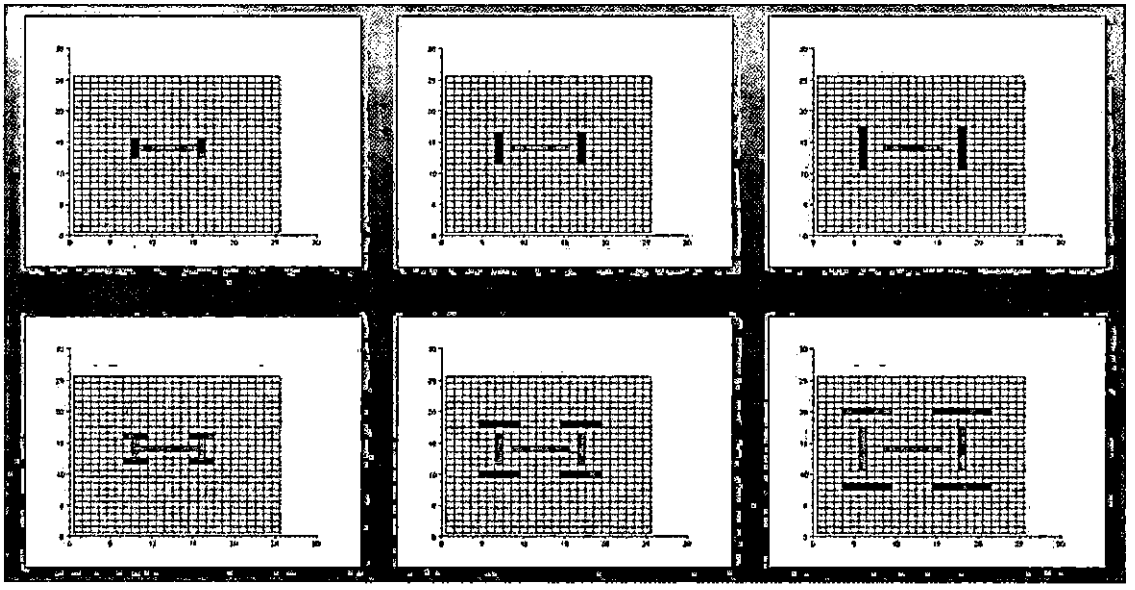

Fig. 3.18: Birth = 1, no death, 7-unit line start, column 1: *radius* = 1, column 2: *radius* = 2, column 3: *radius* = 3; row 1: $t = 1$, row 2: $t = 2$

them. The square pattern was identical for all three death ranges, and the diagonal pattern had minor differences in how the associated death range affected the pattern development in the odd numbered time steps, but by the very next time step (the even ones), the three patterns were identical. Despite the visual differences between these patterns, the similarities in growth behavior among them stands out very strongly.

Tab. 3.1: Birth = 1, line start scaling

radius	new line length	space between lines
1	3	0
2	5	1
3	7	2

3.5 Taking the Larger View... or The Effect of Larger Radii on Birth= 1 Patterns Without Death

3.5.1 Line Starts

Figure 3.18 shows a 7-unit line start with *birth* = 1. The left two diagrams show *radius* = 1, the middle two show *radius* = 2, and the right two show *radius* = 3. Looking at the top row (time step 1), at larger radii, the patterns that developed from the line-start, appear to scale. With birth= 1, and no death, at radius 2, the new lines that grow at the endpoints of the seed line (if not crowded), are now 5 units long (the size of a radius= 2 neighborhood). They no longer attach to the endpoints of the seed line. There is a single square sized space (the neighborhood radius −1) between them. At radius 3 the lines are 7 units long, and the spaces are 2 units. In the bottom row (time step 2) this scaling continues.

We conclude from this that patterns that begin as lines scale according to the neighborhood size with the birth number held constant at 1. This works for line starts because they are dealing with a single point of attachment at all radii.

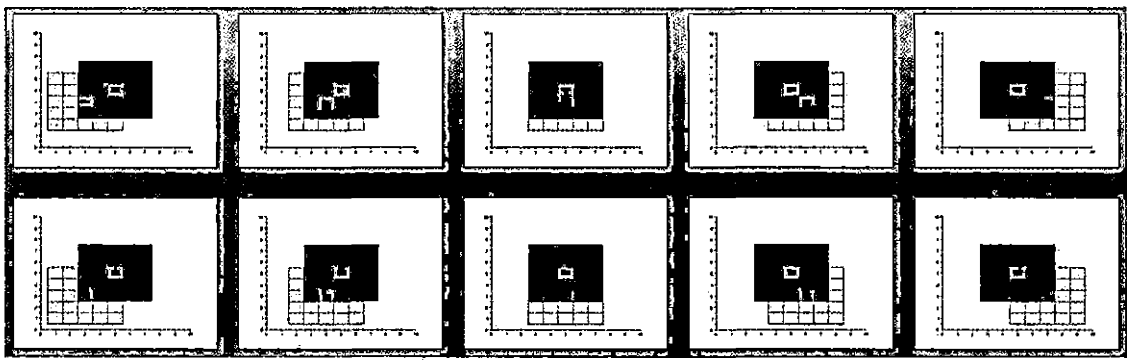

Fig. 3.19: Birth = 1, *radius* = 2, single square start with neighborhood grid, start square in green, target square in red, result of calculation (new growth) in blue

3.5.2 Square Starts

Single Square Starts

Single square starts continue to grow into filled squares at larger radii. However, the filled squares are the size of the radius of calculation. For example, in radius= 2 calculations, the resulting square is 5×5, the size of the current neighborhood. In figure 3.19, the target square is shown in red for clarity. This figure shows the sequence of neighborhoods that generate the bottom two rows of the resulting square. The outside two rows on the other three sides are generated in the same way.

This is a function of the size of the neighborhood. Using the idea of the card with a neighborhood sized aperture, as the calculation radius gets larger, there are more positions of the card (neighborhoods) that can see that single square, and consequently, more empty squares that are changed to values of 1 in the next time step.

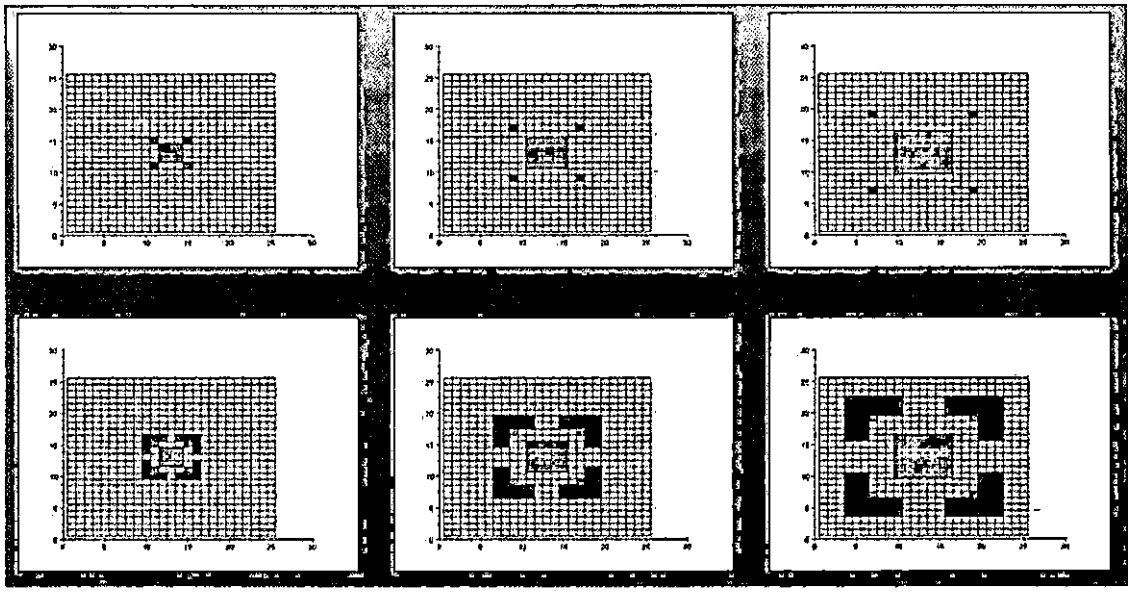

Fig. 3.20: Birth = 1, no death, filled square start, column 1: *radius* = 1, column 2: *radius* = 2, column 3: *radius* = 3; row 1: $t = 1$, row 2: $t = 2$

Filled Square Starts

The patterns for square starts the size of the calculation neighborhood at radius= 2, and radius= 3 are the same. They are scaled versions of the radius= 1 pattern, which include a scaling in the line thickness due to the larger sized calculation neighborhoods being used.

In radius= 2, the lines are 2 rows (or columns) wide. The single square at each of the corners remains, but it is no longer touching the corners. In this pattern it is offset by 1 square.

In radius= 3, the lines have been scaled to 3 rows (or columns) wide, and the offset at the corners is 3 squares.

There are a couple of differences to note:

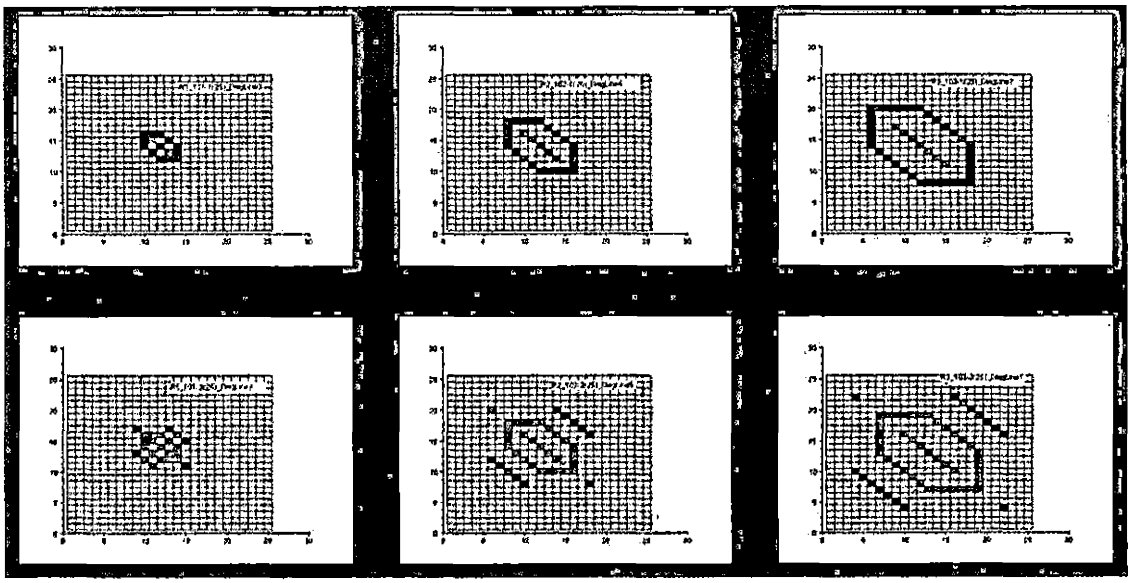

Fig. 3.21: Diagonal line starts, $birth = 1$, column 1: $radius = 1$, column : $radius = 2$, column 3: $radius = 3$; row 1: $t = 1$, row 2: $t = 2$

1. There is an implied offset at the corners of the radius= 1 pattern, but because it is equal to the radius= $1 - 1$, the net offset $= 0$, and thus is not visible.

2. In the radius= 1 pattern, at time step 5 the ends of the corners are treated as endpoints because the line width is 1. In radius $= 2$, the CA see the ends of the corners as corners, themselves, because the lines are 2 rows (or columns) wide. The same thing happens in the radius= 3 pattern, because the lines there are 3 rows (columns) wide.

3.5.3 Diagonal Line Starts

The patterns at larger radii, with birth= 1, and no death, are similar, but scaled by the calculation radius. Figure 3.21 shows the $birth = 1$ diagonal line start for time

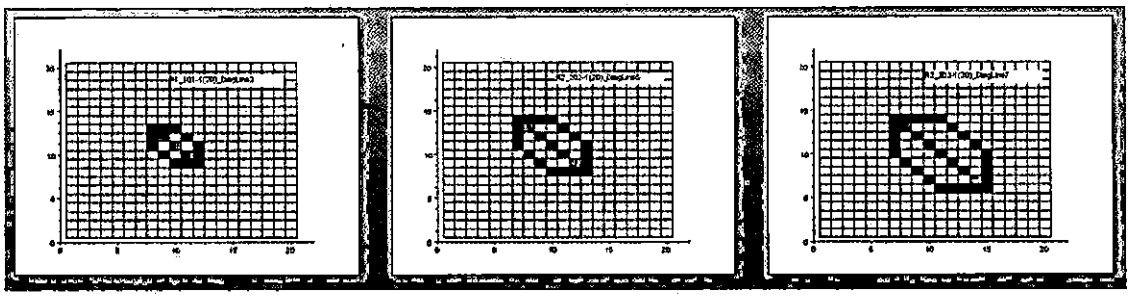

Fig. 3.22: The first time step for diagonal line starts, frame 1: $birth, radius = 1$, frame 2: $birth, radius = 2$, $birth, radius = 3$

steps one (top row) and two (bottom row). The left hand column is $radius = 1$, the center column is $radius = 2$, and the right hand column is with $radius = 3$. The diagonal start line continues to generate parallel, equal length diagonal lines. Now, however, there is an empty diagonal space between the parallel lines equal to the radius−1, and spaces between the endpoints of the diagonal line and the corner pattern equal to radius−1.

In figure 3.22 it is clear that if we make the birth number and the calculation radius equal and vary them together, the pattern continues to develop with the same general look as radius= 1, but the diagonal lines are offset by a horizontal distance equal to the radius, and there is no longer a gap between the endpoints and corners that form there.

3.5.4 Conclusion

This examination of the effect of larger radii on the simple patterns we are using leads to an insight that there is an additional variable that we have not previously

considered: the effect of scaling. The scaling to the size of the calculation neighborhood is so clearly demonstrated in these larger calculation radii, that it becomes clear that scaling in the seed patterns will also affect the developing pattern. If a line, for example, is too short (shorter than the calculation radius) then the developing pattern will collide, and the pattern will change. If that same line is too long, lines that would have intersected in the original pattern may not intersect, leaving gaps too large, for the calculation radius that we are using, to fill. We also saw this in the larger radius square starts. In radius= 1, t= 6, we saw perpendicular lines at the endpoints of the corners. In the larger radii, at the same time step (t= 6) the places that were endpoints in radius= 1, became corners (see figure 3.20 above) due to width of the corner lines that were scaled there. The width of the lines became scaled due to the effect of larger calculation radius.

4. ON BIRTH AND RULES

4.1 Introduction

The grid that a cellular automata grows upon has often been described as the analog of a chessboard [44, 46, 43]. Continuing the analogy, the pieces on the chessboard have only a limited set of moves that are geared to the layout of the chessboard, itself. In this chapter we discuss basic moves (methods of growth) available to a simple cellular automaton with $birth = 1$, $radius = 1$, and no death. We ave already examined the ways that this cellular automata can grow in the previous chapter. In this chapter we will codify what we learned there into a grammatical subset of cellular automata rules. We will extend the what we have observed to characterize the growth behavior for $birth = 2$, $radius = 2$ and $birth = 3$, $radius = 3$.

4.2 Birth = 1 - The Prototypical Seed Patterns

The three prototypical seed patterns in birth= 1 CAs that we discussed previously were line, dot/square, and diagonal. What makes them prototypical? In the initial setup for this investigation we have chosen to use a square grid. Other 2-dimensional grids available. Any single polygon that will tile the plane could make a grid, but hexagonal, diamond, and triangular are some of the commonly used ones. Because we have also chosen to use a Moore neighborhood on this square grid, certain CA

Fig. 4.1: A line can only grow at the endpoints

Fig. 4.2: A single square will grow to the size of the Moore neighborhood

behaviors follow from that, i.e., there are only a limited number of ways that a CA, on a square grid, using a Moore Neighborhood, can grow (if birth= 1): lines have one dimension, and therefore, it can only grow perpendicular to its end points (shown as a diagram in figure 4.1).

A single dot (square) has no dimension, but it has nothing, or everything (depends on your point of view) from which to draw a perpendicular (diagram in figure 4.2).

A square has two dimensions, no endpoints, and the square grid forces growth to be perpendicular to the previous time step (or the seed start at step 1), so that the only way for it to grow is a single square located diagonally out from each corner (which is perpendicular to both of the sides that form the corner). This is shown in the diagram in figure 4.3.

A diagonal line, which, unlike the other patterns, is not orthogonal to the square grid, propagates with identical lines parallel to itself. If the line is part of a larger

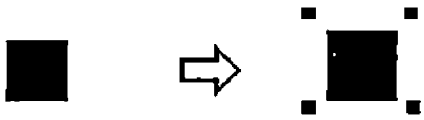

Fig. 4.3: A square will grow to a square with dots sprouting from the corners

Fig. 4.4: A diagonal line will grow parallel to itself

pattern it will grow one line. If the line is by itself, it will grow a new line on each side. In a square grid, each dot of the new diagonal line is perpendicular to its corresponding dot in the original line, as it is offset by the same amount both vertically and horizontally. The diagram in figure 4.4 shows the most general case of a single offset parallel line. Another related thing happens to the endpoints of the diagonal line. Cellular automata begins a corner formation as if they were the single diagonally offset squares that occur in time step one of a square start. This corner formation continues recursively as the parallel lines are propagating.

We have, in the previous chapter identified grammar-like behavior rules (types of growth, perpendicular growth, etc.), but we have not yet isolated which of our two variables, *radius* or *birth*, is the driving factor. It is possible that it takes both. To isolate *birth* number from *radius* value, we will look at these starts (line, filled square, single square, diagonal line) with $birth = 1$ at $radius = 2$.

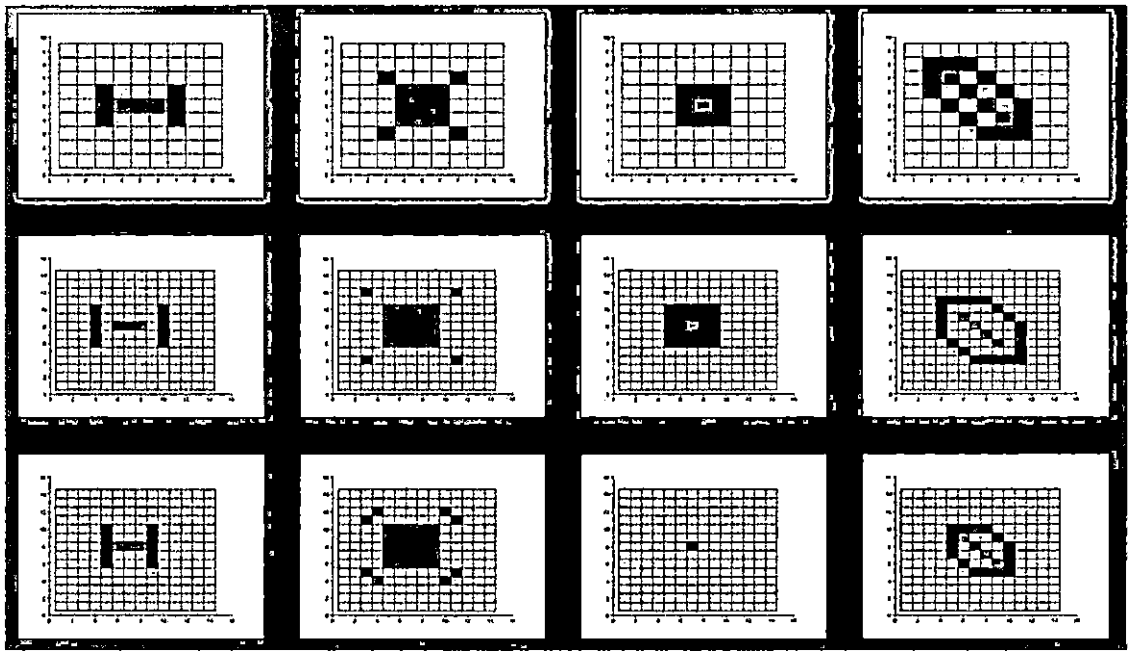

Fig. 4.5: The top row is $birth = 1$, $radius = 1$, the middle row is $birth = 1$, $radius = 2$, the bottom row is $birth = 2$, $radius = 2$

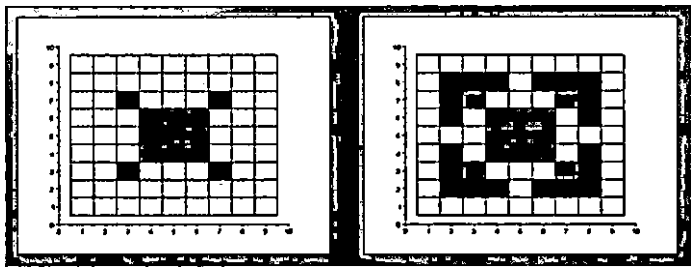

Fig. 4.6: Full square sequence to a new corner for $birth = 1$, $radius = 1$

One thing to note about these images: the start is in green (because this is $timestep = 1$, and there is no death). Comparing the top row of figure 4.5 to the middle row, we see the same basic growth patterns with minor differences in spacing and size of lines and squares that are due to the larger calculation radius at $radius = 2$. The square has a sequence of three time steps as shown in figure 4.6. As we discussed in the previous chapter, the single square start follows the same sequence after $t = 1$ when the larger square is formed. Once a new corner is formed the process starts all over again.

When we look at the bottom row, the first thing we notice is that the single square failed to grow. This is not a surprising result, as the rule that we used for the bottom row specified $birth = 2$, so there was not enough resources there for it to grow. This is something we need to keep in mind, and compensate for, when going to larger *birth* numbers. The second thing that is noticeable is that the spaces we saw at $birth = 1$, $radius = 2$ are gone. There appears to be a relationship or scaling effect between *birth* number and *radius* that is almost like native resolution on a liquid crystal display screen. *Birth* number appears to display without spaces in the same *radius*

number. Comparing the bottom row with the middle row (both are *radius* = 2), the line start and the diagonal are nearly the same, but the filled square, and the single square are not. The filled square is similar, and the single square start needs at least 2 squares to grow. We conclude that the grammar-like behavior is the product of the *birth* rules and not the *radius*. We also conclude that the behavior that we observed at *birth* = 1, *radius* = 1 appears to be a subset of a larger set of rules of growth that encompass other *birth* numbers as well.

To see if this behavior is consistent with other birth numbers, we will examine *birth* = 2 at *radius* = 2, *radius* = 1, and *radius* = 3. (We were unable to look at a smaller *radius* for *birth* = 1.) First, we need to investigate if we can go from two single squares to a larger square.

From figure 4.7 we can see that it is possible to get a filled square from 2 single squares, but there are limitations on the start configuration of the two squares. In order for the two start squares to be included inside the square or rectangle that develops, the squares must be adjacent to each other or be separated by no more than one empty square (*radius* − 1). Out of nine configurations the only ones that worked were 2 diagonals that fit the limitations, and the start with two side by side squares. In those 3 configurations, the two squares were no farther than the *radius* + 1 = 3 from the most distant side of the 4 × 4 resulting square.

4.3 *The Rules for Birth = 2 and Birth = 3*

What do we get if we put our theory to the test, and try *birth* = 2 on a *radius* = 1 grid or a *radius* = 3 grid? So far we haven't been able to try a smaller grid, but if

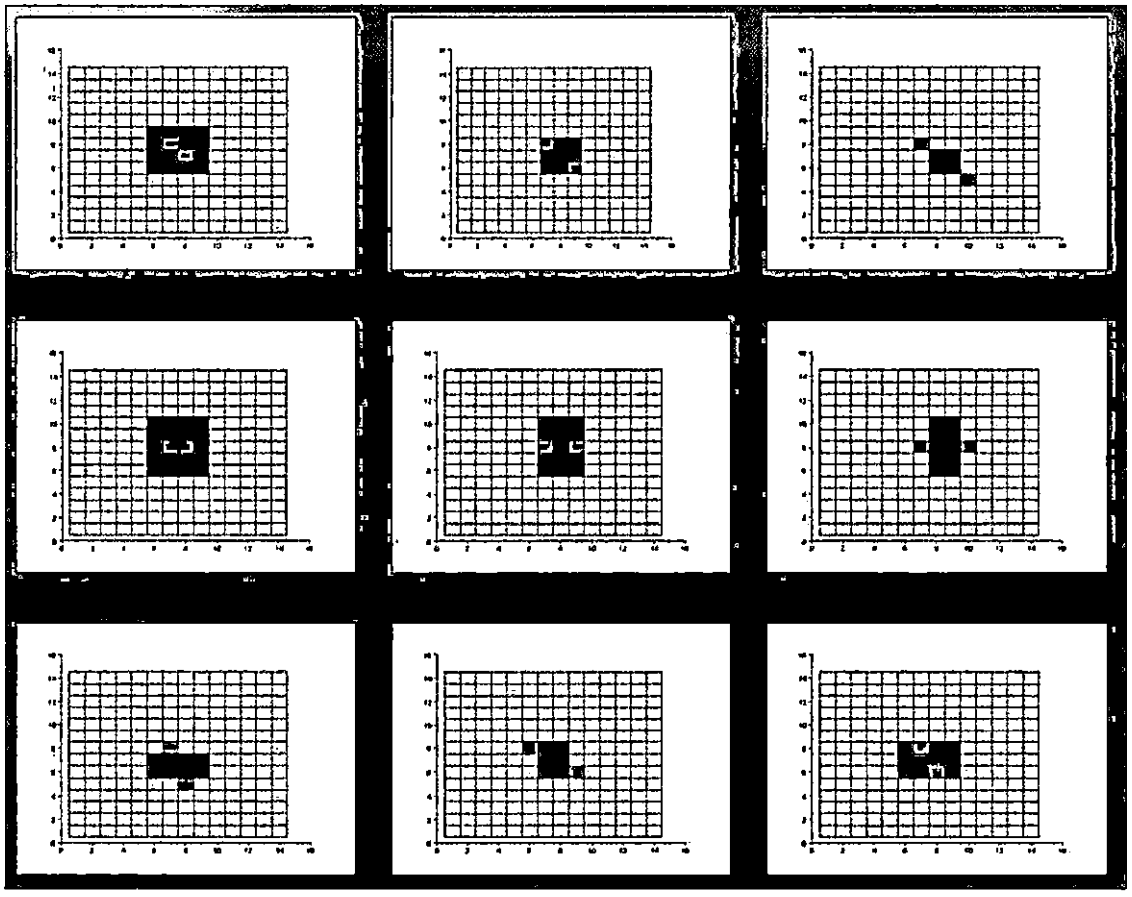

Fig. 4.7: Two square starts in various configurations, top row diagonal offset, the middle row horizontal offset, bottom row asymmetrical offset; column 1: line start, column 2: square start, column 3: single square start, column 4: diagonal line start

we are correct, we will get the same patterns at *radius* = 3, with longer lines and a space of some sort between the endpoint and lines.

Some of the results shown in figure 4.8 were as we expected. Comparing the *birth* = 2 at *radius* = 2 patterns, in the second row, with the *birth* = 2 at *radius* = 3 patterns, in the third row, the results show the same longer lines, larger squares and and the single extra space that we saw in figure 4.5 when we generated *birth* = 1 patterns with *radius* = 2. The full sequence to a corner for *birth* = 2, *radius* = 2, as shown in figure 4.9, still takes two time steps.

Comparing the top row (*birth* = 2 at *radius* = 1) we see a compression in the pattern that is due to the smaller calculation radius which shifts the target square in the neighborhood to the left one square, positioning it on top of the endpoint(s) of the start pattern. This compression of the pattern at smaller *radius* is exactly the opposite of the effect of placing a pattern on a larger *radius*, but the essential pattern is the same. There is something consistent happening here with the compression and expansion of the patterns. When we get compression, we are displaying a *birth* = 2 pattern with a *radius* = 1 neighborhood, which compresses the pattern, or moves the developing vertical lines in a negative direction. The *birth* number is constant, but we are changing the *radius* of the calculation neighborhood. At *radius* = 2, *birth* = 2, there is no offset, so it appears to be a difference between the value of the *birth* number and the value of the *radius*. At *radius* = 3, *birth* = 2, there is a positive offset that results in a single space between the endpoints on the start line and the vertical lines. That would imply that the relationship is $radius - birthnumber$. For *radius* = 3, that would be $3 - 2 = 1$. Does this explain the compression? For

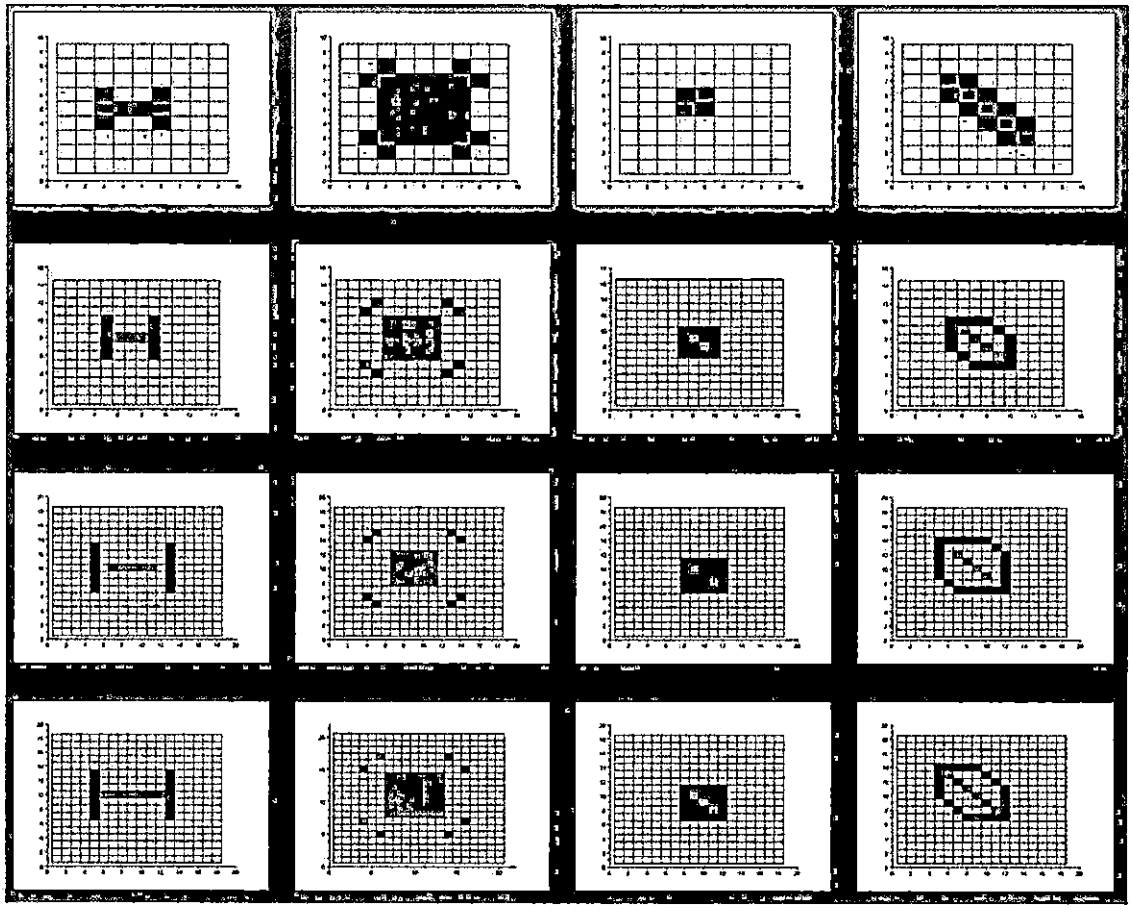

Fig. 4.8: The top row is *birth* = 2 at *radius* = 1, second row is *birth* = 2*at*radius=2, the third row is *birth* = 2 at *radius* = 3, the bottom row is *birth* = 3 at *radius* = 3; column 1: line start, column 2: square start, column 3: single square start, column 4: diagonal line start

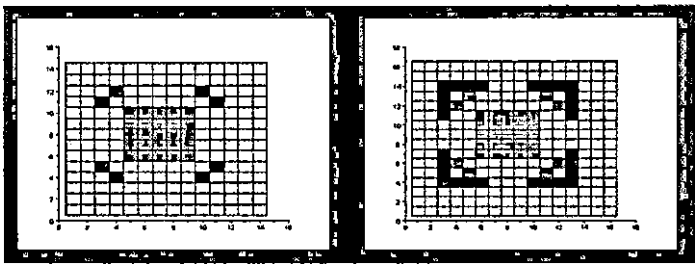

Fig. 4.9: Full square sequence to a new corner for *birth* = 2, *radius* = 2

Tab. 4.1: Space size in patterns at different radii, *radius – birth number = space size*

radius	birth number	space size
1	1	0
2	1	+1
1	2	-1
2	2	0
3	2	+1
3	3	0

radius = 1, then, it would be $1 - 2 = -1$. The size of the space(s) or compression in a pattern is, thus, shown to be equal to *radius-birth* number.

Table 4.1 shows the results of our analysis of figure 4.5 and figure 4.8 to determine what relationship between *radius* and *birth* number was causing the compression and expansion of the pattern when different calculation radii were used.

Notice that the bottom row in figure 4.8 which is *birth* = 3 at *radius* = 3 is the same for the line at larger scale (caused by the larger *radius*). The filled square has a slightly different configuration, , as shown in figure 4.10, (which takes 3 time steps to complete to a new corner instead of the the 2 time steps it takes for *birth* = 1, and

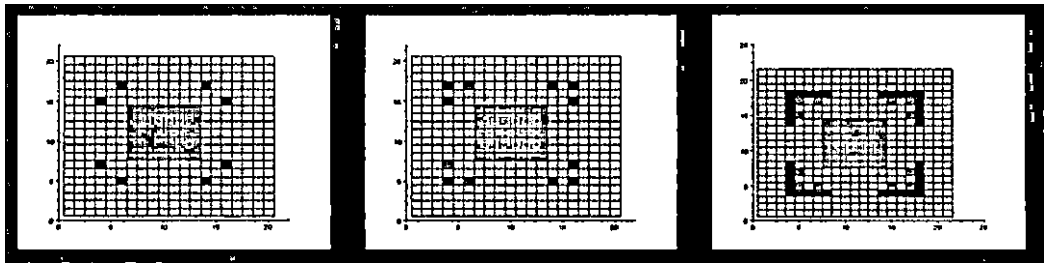

Fig. 4.10: Full square sequence to a new corner for $birth = 3$, $radius = 3$

$birth = 2$).

Three squares to a large square is a bit different, but consistent with what we saw at $birth = 2$. The squares from the start are still not more than $radius + 1$ distance from the farthest side of the square. The diagonal in $birth = 3$, $radius = 3$, forms the diagonal pattern in the second row, with scaling that spaces the parallel lines farther apart, and causes the corners that form over the end points to encompass two squares at each end of the original diagonal line. From this we can conclude that the driving variable in this grammar-like growth is the birth number alone. In each of the larger *birth* numbers we see the same growth pattern or some analog of the growth patterns that we saw in $birth = 1$ patterns. This consistency across multiple *birth* values leads us to the idea that the growth patterns observed in each of these *birth* numbers is a subset of a much larger context sensitive or context dependent graphical grammar.

4.4 L-systems

"Organic form itself is found, mathematically speaking, to be a function of time.... We might call the form of an organism an event in space-time, and not merely a configuration in space." –D'Arcy Thompson [14]

The kind of growth that we have observed in these cellular automata, with $birth = 1$, $birth = 2$, or $birth = 3$, and no death, is growth along a wavefront that begins at the initial seed pattern and propagates out, in all directions, with each time step, from there. This is important to keep in mind because new growth at the edges of the pattern is perpendicular to the wavefront propagation in the previous time step. The context that generates the next time step pattern appears to be only behind the pattern that is generated in the current time step. This observation ignores the fact that there must be sufficient empty space ahead of the wavefront to preclude crowding, if the pattern is to form completely. To simplify this investigation of a grammar, we are going to stipulate that both the CA and the associated grammar will be 2-dimensional, and limited to a single, convex seed pattern, to eliminate the possibilities of two patterns colliding or the some concave portion of the pattern colliding with itself.

What kind of grammar are we going to need? We need a grammar that can handle graphical words, that is context sensitive, and one that updates globally all at one time (like our totalistic CA). The update constraint eliminates Chomsky grammars where grammatical productions are applied sequentially. Lindenmayer Systems (hereafter called L-systems) are massively parallel in that the grammar productions are applied globally. L-systems are an array rewriting grammar system that was developed to handle biological modeling. They were designed to handle graphical words, and they can be context sensitive (with either 1-sided or 2-sided context sensitivity). They can also handle 2-dimensional grammar.

L-systems are defined as a tuple:

$$G = (V, \omega, P) \tag{4.1}$$

Where V is the alphabet, ω is the start symbol or symbols from V that defines the initial state, and P is the set of production rules. The $<$ represents the search for a left context, and $>$ represents the search for the right context. Later on we will be using arrows to indicate directional moves. These will generally appear in pairs, one right or left, and one up or down. The magnitude of the move will be indicated by a number in parentheses after the arrow. These directional pairs will be enclosed in square brackets, which in L-systems notation indicates information that will be ignored for context purposes.

Directional moves in L-systems are defined in terms of Logo's turtle graphics. Turtle graphics involve the commands, "draw" "forward", "move forward" (with pen up), "pen up", "pen down", "turn left 90 deg", and "turn right 180 deg". The turtle has three properties:

1. a position

2. an orientation

3. a pen, which has color, width, and is either up (not writing), or down (writing).

All changes of position or orientation are described relative to the "turtle's" current position.

Table 4.2 is a table of attributes, direction symbols, operators, and characters that are part of the alphabet, V for CAs. Most of these are part of the adaptation of

Tab. 4.2: List of graphical forms and symbols

Name	Pattern	Symbol
point, endpoint	■	■
vertex	⌟	∧
rotate 90°	↻ 90°	$R_{90°}$
rotate 45°	↻ 45°	$R_{45°}$
up 1	move up one square	↑ (1)
down 1	move down one square	↓ (1)
right 1	move one square to the right	→ (1)
left 1	move one square to the left	← (1)
search for left context		<
search for right context		>
skip over in search for context		[ ]
line	—	L
forward diagonal line	╱	D_F
backward diagonal line	╲	D_B
corner	⌝	C
square	□	B
corner dot, $B = 1$	see fig. 4.6	d_1
corner dots, $B = 2$	see fig. 4.9	d_2
corner dots, $B = 3$	see fig. 4.10	d_3
$2R + 1$ space	⊏⊐	SP
NOTES:	Attributes, like ■ (endpoint), and ∧, (vertex), act as modifiers to specify where the attachment point (which serves as the context) is on pattern symbols	■, ∧
	■ is overloaded to mean both point (single square - a new growth), and endpoint (an attribute on an existing pattern).	■
	The choice of counter clockwise direction for the rotate symbol is completely arbitrary, but it is necessary to be consistent about the direction.	↻

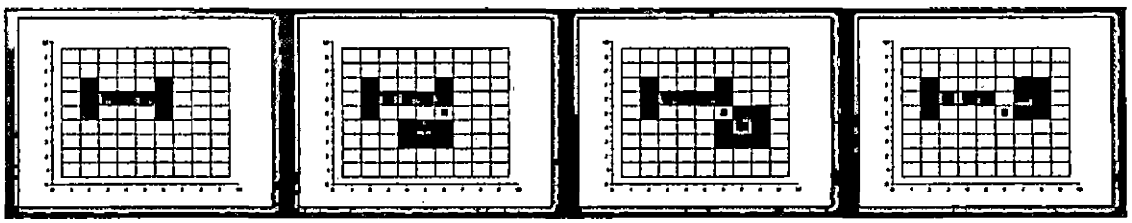

Fig. 4.11: The effect of having other occupied squares within the neighborhood $2r + 1$ *radius*.

L-systems notation to cellular automata. The search for context symbols (<, and <) and the square brackets ([,]) are directly from the L-system specifications [33].

Why are we using SP for the character? We are using SP for the space character that represents the distance, $2R+1$, which is the length of one side of the calculation neighborhood.

Why do we need SP? In figure 4.11, the target square, at the center of the neighborhood is marked with a black dot. The first frame shows how the pattern (that we are using as an example) should develop in the first time step. The neighborhood is shown in a red grid. The target square, at the center of the neighborhood is shown with a black dot, to make it easier to identify. The second, third and fourth frames show what happens when there is a single occupied square at the lower left, lower right, and upper right corners of the the neighborhood. The extra square, in each of these cases keeps the CA from completing its pattern, due to crowding. If we specify this space, the width of the current calculation neighborhood, must be empty, as one side of the context, then the pattern can develop completely. The use of the SP character in the vocabulary, guarantees that the full expression of the rule takes place. It also makes this a 2L-system, with context sensitivity on both sides of the

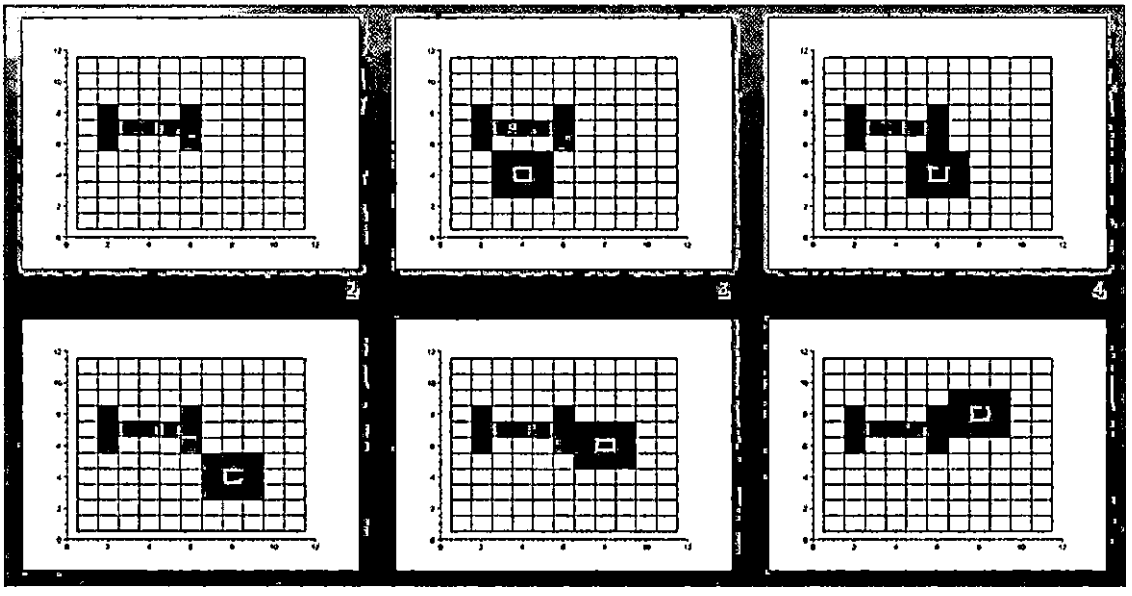

Fig. 4.12: The occupied squares are all outside the $2r + 1$ neighborhood

new term.

In figure 4.12, the first panel, again, shows how the pattern should fully develop. The other panels show occupied squares outside the neighborhood radius. Notice that the line start pattern fully develops as does the single square to large square, because they are sufficiently far apart to eliminate interference, and permit the full, uncrowded growth of both.

The application of the L-system grammar to describe the behavior of cellular automata requires that we stipulate some global constraints. An L-system updates all possible applications of a production (rules) at one time. The following points guide and constrain our approach:

1. We will develop the subset of the alphabet, V, that we will use from points.

In effect we will create the alphabet from 0-dimensions to 1-dimension and use those alphabet characters to create the alphabet for our 2-dimensional, square grid with Moore neighborhood, which is a subset of the alphabet, V, for all cellular automata.

2. New lines are $2r + 1$ in length (the length of one side of the current radius of calculation), and attach the mid-point of the new line at an endpoint of an existing line.

3. Context for production rules comes from endpoints (more about this later), on one side, and enough space to grow on the other.

4. Rules may applied in any rotational multiple of 45° or 90°, as long as the proper context exists in that place, and rotation (the limitation to these two angles is due to our choice of grid, and may be different in other grids- like a hexagonal grid).

Developing an Alphabet Subset

Cellular automata can be 1-dimensional, 2-dimensional, or have even more dimensions than that. From geometry, we know that points are dimensionless. In a CA, a single square, by itself, is a point. In a 1-dimensional CA, the only figures are a point, and a line. (Often, 1-dimensional CAs are plotted against a vertical timeline, which gives them the appearance of 2-dimensions.)

1. We propose that a line, for the purposes of this L-system, is a 1-dimensional figure that consists of two points (endpoints) connected by an edge. New growth

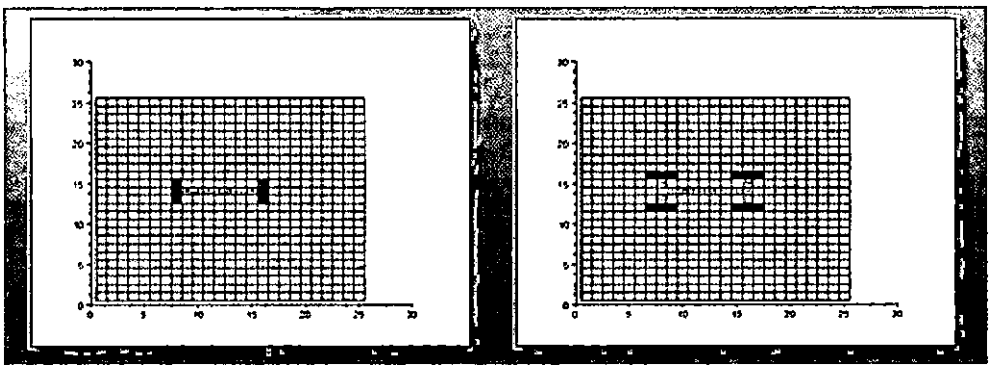

Fig. 4.13: Birth = 1, *radius* = 1, line start, left frame: $t = 1$, right frame: $t = 2$

Tab. 4.3: The L-system production for developing a line from a single square

variables:	■, L, SP
constants:	
start:	■
rules:	SP■SP ⇒ SP ■< ■ >SP (repeat)
	SP■... ■ < ■ > SP⇒ L

can take place only at the endpoints of the line. Endpoints are attributes of lines. A line has 2 points of attachment.

The procedure in table 4.3 builds a line through simple concatenation. This leaves only the two points at the ends of the line available for new growth in the next time step.

2. In a 2-dimensional CA, two mutually perpendicular, equal lines can be connected at one endpoint on each line, essentially, superimposing one line's endpoint on top of the other line's endpoint. This forms a corner with a vertex. On a corner, new growth can only take place at the remaining endpoints of the two lines or at

Tab. 4.4: The L-system production for developing a corner from a line

variables:	L, C, SP
constants:	$R_{90°}$, ■
start:	L
rules:	SP ■L■ SP ⇒ SP ■ L ■ [$R_{90°}$] < ■ L ■ > SP ⇒ C

Tab. 4.5: The L-system production for developing a filled square from a single square

variables:	■, B, SP
constants:	$R_{45°}$
start:	■
rules:	SP< ■ >SP⇒ SP ■< ■ >SP$R_{45°}$ [repeat(8)] ⇒ B

the vertex, where the lines are joined. A corner has the attributes of an endpoint connected to a vertex, which is connect to another endpoint. A corner has three points of attachment.

In the production in table 4.4, the corner formed can be thought of as having two line segments connected together at a vertex. In the cellular automata, the only places to connect to are the endpoints of the lines, and the vertex (which is used as a reference for building a new corner). A visual construction of the corner might look like this:

■—∧—■.

3. A point or single square can form a larger square by creating another adjacent point, rotating 45°. This is repeated eight times.

   In figure 4.5, the single square takes one time step to become a neighborhood-

Tab. 4.6: The L-system production for developing a square from a corner

variables:	B, C, SP
constants:	$R_{90°}$, ■
start:	C
rules:	SP■C■SP ⇒SP ■C■< C >SP[$R_{90°}$] [repeat(4)] ⇒ B

sized square because the *radius* = 1 neighborhood, with the rule, *birth* = 1, can see the single square from the eight different positions all around the single square start.

4. Four instances of a corner can be rotated in multiples of 90 degrees, and joined at their endpoints to form a square. A square has four points of attachment - one at each each vertex. A square has the attribues of a vertex connected to a vertex, rotated 90°, connected to a third vertex, rotated 90°, connected by to a fourth vertex, rotated 90°, connected to the unconnected side of the original vertex.

   The square built in the procedure in figure 4.6has no endpoints to attach to because the endpoints of the corners are butted up against each other. The only places new growth can take place is off the vertices of the four corners that form the square.

5. A diagonal line forms from a single point with unit steps (right or left and up or down). Pick one (right or left), and one (up or down). Say we choose: (right-1, up-1). To make a diagonal line we repeat (right-1, up-1) however many times we need for the length of the diagonal *line* − 1 (we had one point to begin with). This creates a diagonal line that rises up and to the right.

Tab. 4.7: The L-system production for developing a diagonal line from a single point

variables:	■, SP
constants:	↑, →
start:	■
rules:	SP■SP ⇒SP ■[↑(1),→(1)]< ■ >SP

Tab. 4.8: The L-system production for $r = 1$, *birth* $= 1$, no *death*, line start

variables:	L, SP
constants:	■
start:	L_0
rules:	L_0⇒ SP■L_0■SP
	SP■L_0■SP ⇒ SP<L>L_0<L>SP

A diagonal line has the attributes of two endpoints connected step-wise to each other. This is shown in the L-system production in table 4.7.

All four of the prototypical start patterns for the 2-dimensional CAs that we have been investigating have now been created from points and lines. The important thing to remember is that the features that drive the contextual growth *are the endpoints and vertices. The line edges just connect the endpoints.* The symbol L, for line, will be an implied ■L■, which shows the endpoints.

An Example of L-systems Applied to Cellular Automata - a Line Start

We are going use the line start CA. An example of the first two time steps in this cellular automata is shown in figure 4.13.

In table 4.8, the <, and >, symbols connect the rewritten term, <L> for example, to its left<, and >right context. The second rule, says, if enough space exists to the left of the of the original line, L_0, and an endpoint exists, a new vertical line

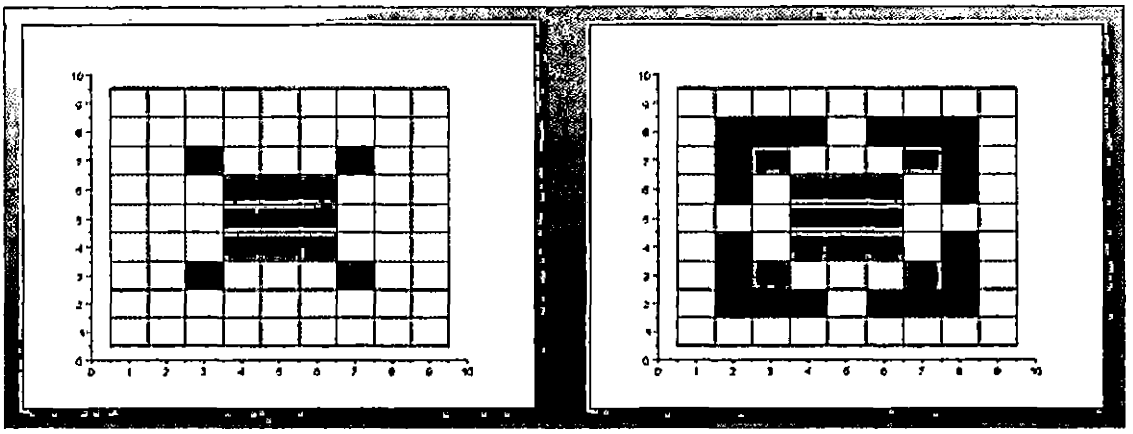

Fig. 4.14: Birth = 1, *radius* = 1, no death, sequence to new corner, each frame represents 1 time step

can attach its mid-point at the left hand endpoint of the original line, and if enough space exists to the right of the other end of the original line, another new line can attach (or grow) with its mid-point at the endpoint of the original line, L_0. The production in table 4.8 was written to do both end lines in one rule. It could have been written in two separate rules, one to do the left hand side of the original line, and one for the right hand side. This production for line start growth describes rules that function in larger birth numbers (and radii). The larger patterns for larger birth number will occur as a result of a larger calculation neighborhood, without requiring any adjustment to L-system production.

Square Start

The first two time steps to a new corner for a *birth* = 1, *radius* = 1, square start are shown in figure 4.14. The L-system procedure to generate this pattern is in table ??, where the first rule specifies the original square, at a vertex, will grow a single square,

Tab. 4.9: The L-system production for $r = 1$, *birth* $= 1$, no *death*, square start

variables:	B, SP, C, d_1
constants:	$\wedge$
start:	B
rules:	B $\Rightarrow$ B$\wedge$SP
	$B\wedge$SP $\Rightarrow$ B$<$ d_1 $>$SP
	B d_1 SP $\Rightarrow$ B d_1 $<$ C $>$ SP

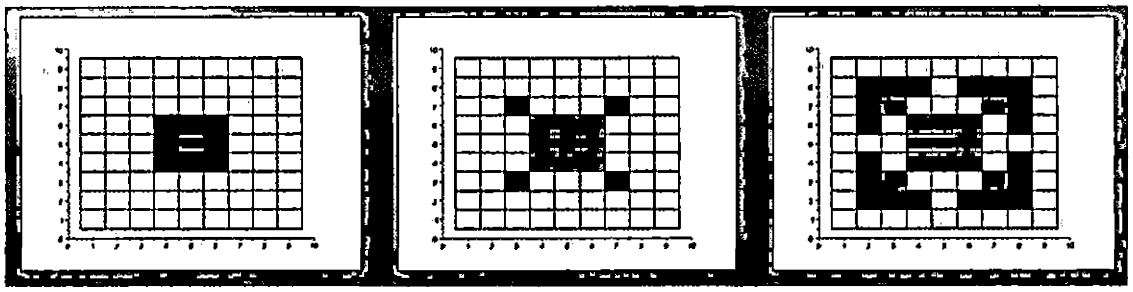

Fig. 4.15: Single square corner sequence, each frame represents 1 time step

providing there is enough space for it to grow. The L-system procedure will globally apply this rule where ever the context (square vertex on one side, and space enough on the other) exists in the start pattern. The second rule specifies that if there is a single square, with enough space on one side, a new corner will form.

Single Square Start

It is worth noting again that the single square start take one time step to become a neighborhood sized square and from then on, it follows the same development as the square start. For a given point in the sequence the single square will get there one time step later.

Tab. 4.10: The L-system production for $r = 1$, *birth* $= 1$, no *death*, single square start

variables:	■, B, SP, C, d_1
constants:	$\wedge$
start:	■
rules:	■ SP$\Rightarrow$8(■ $<$ ■ $>$SP$[R_{45°}]$)
	$\Rightarrow$ B
	B $\Rightarrow$ B$\wedge$ SP
	B$\wedge$SP $\Rightarrow$ B$<$ d_1 $>$SP
	Bd_1 SP $\Rightarrow$ B d_1 $<$ C $>$ SP

Tab. 4.11: The L-system production for $r = 1$, *birth* $= 1$, no *death*, backward diagonal line start

variables:	D_B, C, SP, ■
constants:	■, $\uparrow$, $\downarrow$, $\leftarrow$, $\rightarrow$, $\wedge$
start:	D_B
rules:	$D_B \Rightarrow$ SP $D_B[\uparrow (1),\rightarrow (1)]<$ ■ $>$SP
	$D_B \Rightarrow$ SP $D_B[\leftarrow (1),\downarrow (1)]<$ ■ $>$SP
	■ SP$\Rightarrow$■ $<$ C $>$SP
	■C$\wedge$ SP$\Rightarrow$■ C$<$ d_1 $>$SP

Diagonal Line Starts

The procedures in table 4.11, and table 4.12, work for larger birth numbers as well. The only modification necessary is to substitute the birth number into the first value in the direction tuple in each rule. We will provide a generic example for the forward diagonal line for all birth number in table 4.13. This change generates the space between the parallel diagonal lines that occur with birth numbers > 1. The corner procedure for that birth number will have to be substituted as well. The point to this is that essentially the same rules (in the L-system we have developed) will generate a pattern from a diagonal line start.

Tab. 4.12: The L-system production for $r = 1$, *birth* $= 1$, no *death*, forward diagonal line start

variables:	D_B, C, SP, ■
constants:	■, ↑, ↓, ←, →, ∧
start:	D_B
rules:	$D_B \Rightarrow$ SP $D_B[\rightarrow (1),\downarrow (1)]<$ ■ >SP
	$D_B \Rightarrow$ SP $D_B[\uparrow (1),\rightarrow (1)]<$ ■ >SP
	■ SP⇒■ $< C >$SP
	■C∧SP⇒■ C$< d_1 >$SP

Tab. 4.13: The L-system production for generic birth number, no *death*, forward diagonal line start

variables:	D_B, C, SP, ■
constants:	■, ↑, ↓, ←, →, ∧
start:	D_B
rules:	$D_B \Rightarrow$ SP $D_B[\rightarrow$ (*birthnumber*),↓ (1)]<
	■ >SP
	$D_B \Rightarrow$ SP $D_B[\uparrow$ (*birthnumber*),→ (1)]<
	■ >SP
	■ SP⇒■ $< C >$SP
	■C∧SP⇒■ C$< d_{birthnumber} >$SP

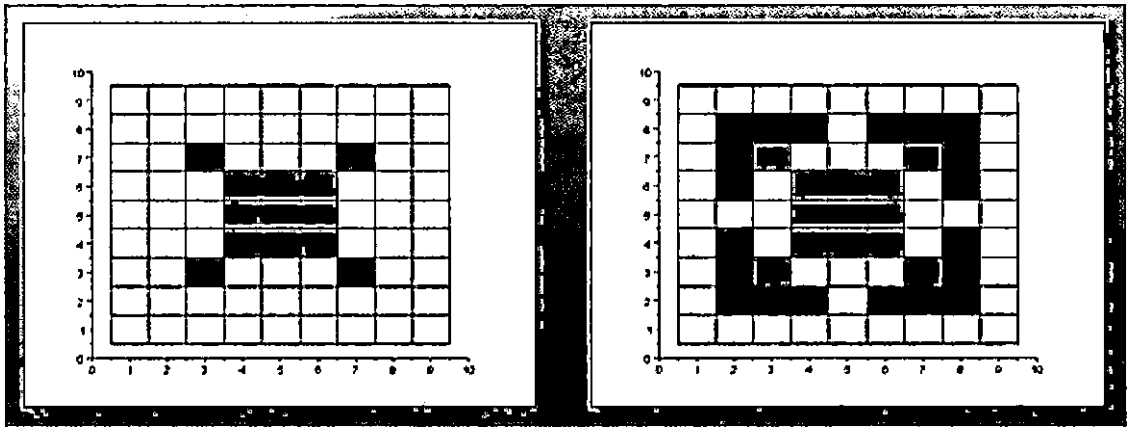

Fig. 4.16: Birth = 1, *radius* = 1, no death, sequence to new corner, each frame represents 1 time step

Tab. 4.14: The L-system production for developing a *birth* = 1 new corner

variables:	■, C, d_1, SP
constants:	↑, →, ∧
start:	C
rules:	C∧SP⇒ C∧[↑(1),→(1)]< ■ >SP ⇒ Cd_1
	Cd_1 SP⇒Cd_1< C >SP

Birth = 1 *Corners*

We are now going to take a look at how the corners at *radius* = 1, *radius* = 2, and *radius* = 3 corners develop. To make it easier to compare them, the figure from the square start section above and procedure are reproduced here.

The procedure in table 4.16 is the L-system procedure to generate the *radius* = 1 corner shown in figure 4.16. Because each of these corners is unique in either number of dots in the corner formation or placement of those dots, it is important to note

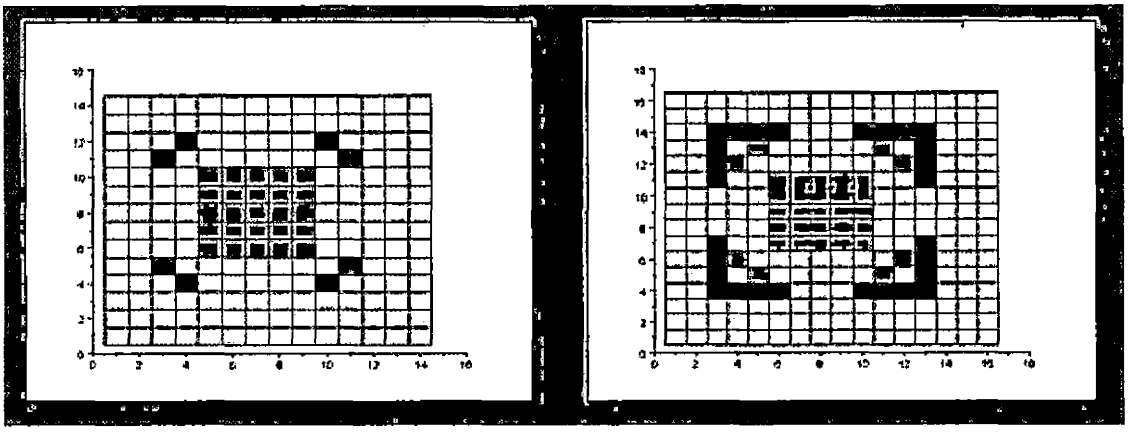

Fig. 4.17: Birth = 2, *radius* = 2, no death, sequence to new corner, each frame represents 1 time step

Tab. 4.15: The L-system production for developing a *birth* = 2 new corner

variables:	■, C, d_2, SP
constants:	↑, →, ∧
start:	C
rules:	C∧ SP ⇒ C∧SP[↑(2),→(1)]< ■ >SP
	C∧ SP ⇒ C∧SP[→(2),↑(1)]< ■ >SP
	⇒ Cd_2
	Cd_2 SP⇒Cd_2< C >SP

that this *radius* + 1 corner has a single square diagonally adjacent to the vertex at each corner of the start square. If the order of the direction arrows is reversed, it still places the dot in the same spot.

Birth = 2 *Corners*

Figure 4.17 shows (at *radius* = 2) the sequence for developing a *birth* = 2 corner. This corner forms two dots at each vertex (not directly adjacent, but just beyond the position of the *radius* = 1 dot), before completing the new corner. The L-system

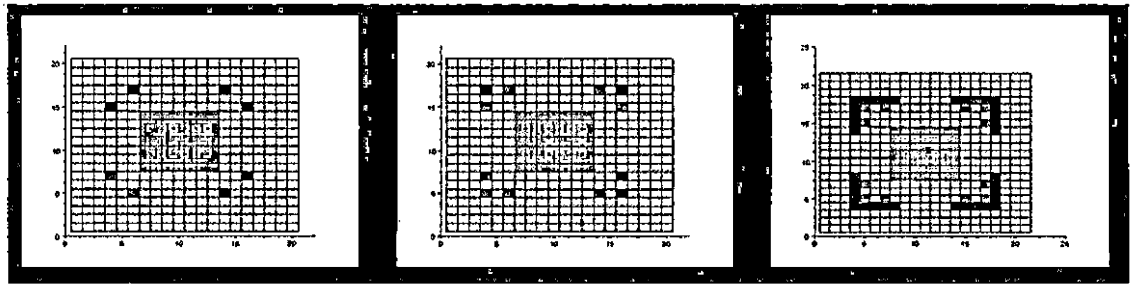

Fig. 4.18: Birth = 3, *radius* = 3, no death, sequence to new corner, each frame represents 1 time step

Tab. 4.16: The L-system production for developing a *birth* = 3 new corner

variables:	■, C, d_2, SP
constants:	↑, →, ∧
start:	C
rules:	C∧ SP ⇒ C∧SP[↑(3),→(1)]< ■ >SP
	C∧ SP ⇒ C∧SP[→(3),↑(1)]< ■ >SP ⇒ Cd_3
	Cd_3 SP⇒Cd_3< d_1 >SP
	Cd_3d_1SP⇒Cd_3d_1< C >SP

procedure to generate the *birth* = 2 corner is in table 4.15. Rules for the dot placement are similar to the *birth* = 1 corner rules, but reversing order of the direction arrows no longer places the dots in the same spot. They are symmetric along each of the diagonal axes of symmetry.

Birth = 3 *Corners*

This *birth* = 3 corner (shown in figure 4.18) is a little different than the previous two corners. It has three dots, but it takes an extra step in the sequence to form the new corner. The first rule in the L-system procedure forms two dots in the first rule are placed, one, vertically adjacent to one of the *birth* = 2 dots, and one, horizontally

Tab. 4.17: The correlation between *radius* number and L-system production directions for developing a new corner

Radius Number	L-system directions
1	[↑(1), →(1)]
2	[↑(2), →(1)] [→(2), ↑(1)]
3	[↑(3), →(1)] [→(3), ↑(1)]

adjacent to the other *birth* = 2 dot. The second rule places a single dot where the row and column of the first two dots intersect. Only then, does the new corner form from the third rule in the L-system procedure. All the corners are symmetric about one of the diagonal axes of symmetry. There is one at each axis (each corner) of the original start square, thus preserving the symmetry of the original pattern.

4.5 Conclusion

The procedures that we developed for line and diagonal line starts work for *birth* > 1. Those developed for single square starts requires at least as many squares for the start as the birth number. There are limits to the ways that these squares can be configured. But, the procedures do correlate to each other. Square starts appear to build corner dot patterns unique to each birth number before the new corner form. However, an interesting result of using the L-system productions in tables 4.14, 4.15, 4.16 and the first associated direction from the original corner vertex mirrors the *radius* number. This is shown in table 4.17.

Each of these methods of growth, act to preserve the symmetry of the original seed pattern. The recursive growth behavior that led us to the grammar presented

in this chapter, combined with the grammar, itself, leads us to a belief that simple cellular automata, with single, convex start patterns are all a discrete variety of fractal pattern, whether their fractal nature is a series of repeating patterns that grow larger with constant time, or is of a type where the cycles of growth wave fronts recursively enclose a similar smaller pattern formed in the previous cycle in constant time – a type that is concentrically fractal.

Other types of grids, using different neighborhoods for calculation, may have different, but largely equivalent prototypical seed patterns and different L-system procedures to accommodate growth with different grids and neighborhoods adapted to those grids.

The entire grammar for cellular automata is exceedingly complex, but it appears there is some nominal subset of these rules about which the rest of the grammar pivots. These three prototypical seed patterns, in the simple patterns that we are investigating, lead to a subset of this context free grammar that defines the way those CAs with the rules, birth=1, and no death, grow. The changes necessary to use those rule in $birth = 2$, and $birth = 3$ cellular automata are minimal. Some changes are required to provide enough resources for the CA to grow, and some seem to be the result of the uniqueness of the dot patterns that lead up to new corners.

The simple subsets of the context free grammar of growth that we have shown in this chapter lead us to believe that there exists a kind of slow manifold made up of basis vectors that describes how we can change the rules and starting conditions to develop families of patterns. In these families of patterns the variables in the rules and the initial conditions are no longer independent of each other, they become a set

of dependent variables. Change one condition to a value outside the underlying range of values and the pattern changes. Stay within the defined values, and the resulting patterns can be nearly identical depending on where they would fall in the manifold. We will discuss this further in the next chapter.

5. THE EFFECT OF SCALING ON DEVELOPING PATTERNS

5.1 Introduction

As we saw in the chapter on fractal patterns, there is an almost harmonic effect between the scaling of the seed pattern and size of the radius of calculation. The question remains, are there other variables (birth number, death range, calculation radius, seed pattern) that can be scaled to produce a particular pattern effect? We have already seen the scaling effect of larger calculation radii longer lines, gaps at endpoints, and for square starts, the width of the lines are scaled, as well.

We have already seen families of patterns that are identical in the birth= 1, radius= 1, square start, with death values of > 1, > 2, and > 3, and the diagonal starts for the same values. The diagonal start is so strongly bound together that the effect of death in the odd time steps is different, but, in the end, at the even time steps, they are all identical.

We have seen a series of cellular automata with both line starts and filled squares using starts that were scaled to the radius of calculation, with birth with exactly one neighbor, and no death. The resulting patterns were scaled versions of each other. This demonstrated that choosing the seed the size of the neighborhood produced scaled versions of the patterns, as well.

The results of scaling birth and radius together in the diagonal starts suggests that

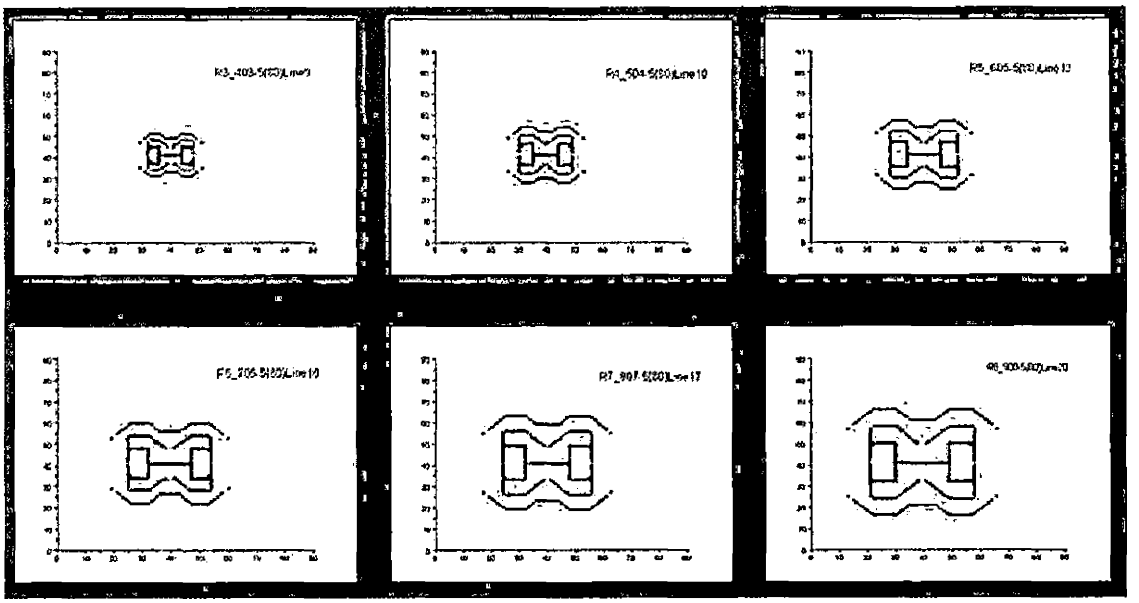

Fig. 5.1: Scaled pattern - multi-radius, multi-birth number, all $t = 5$, top row (L to R): $radius = 3, birth = 4$; $radius = 4, birth = 5$; $radius = 5, birth = 6$; bottom row (L to R): $radius = 6, birth = 7$; $radius = 7, birth = 8$; $radius = 8, birth = 9$

we could, by scaling the start to the calculation radius as well, be able to reproduce a nearly identical scaled pattern with different calculation radii, different start line lengths and different birth numbers.

We are going to discuss one of these families.We have chosen a pattern without death in the rules, to see if birth values can be part of the scaling process. We will analyze what variables contribute to the pattern development at different radii, with different line lengths for seed patterns, and different birth numbers. The pattern we will be examining is formed from a line start, with birth=radius+1.

Figure 5.1 shows patterns at radius= 4, birth= 5; radius= 5, birth= 6; radius= 6, birth= 7; radius= 7, birth= 8. Note that the odd numbered radii, have odd length

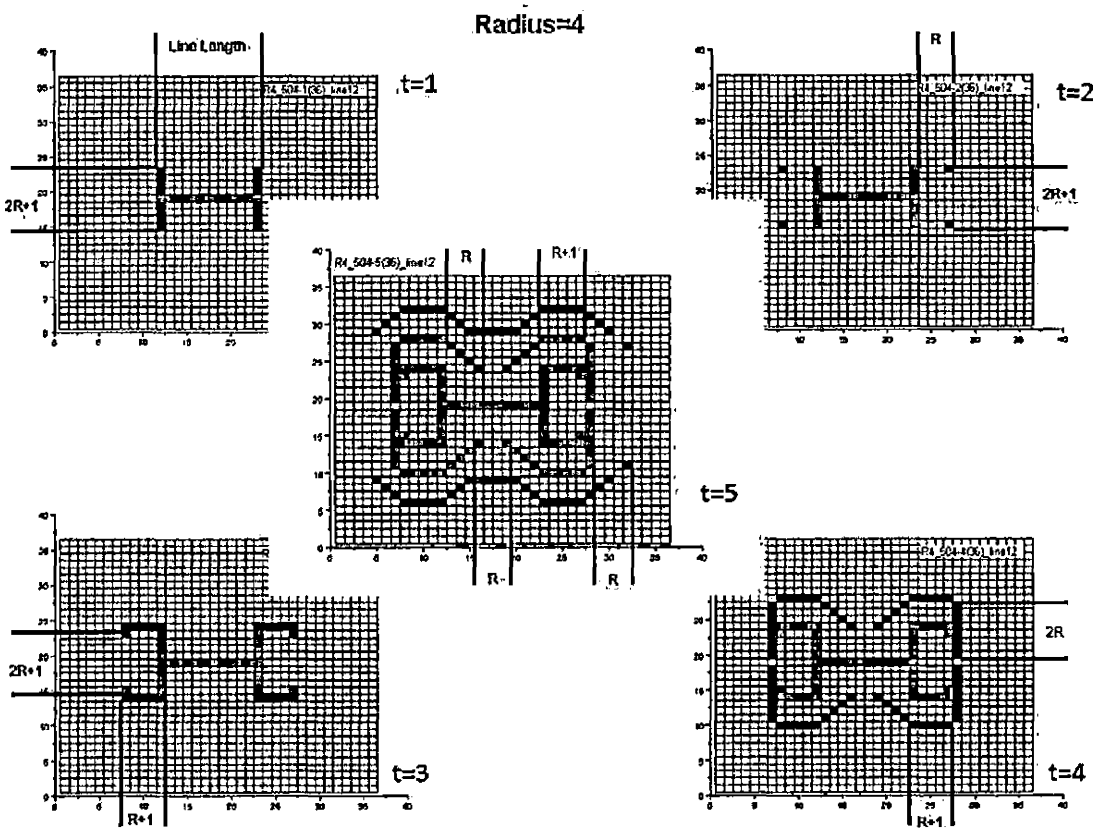

Fig. 5.2: Scaled pattern *radius* = 4, *birth* = 5, time steps 1-5

line starts, and the even numbered radii, have even length line starts.

5.2 Analysis of the Effect of Radius, Seed Start, and Birth Number

"Why," said the Dodo, "the best way to explain it is to do it." –Lewis Carroll [10]

Figures 5.2 and 5.3 show the first 5 time steps for *radius* = 4 and *radius* = 3. Notice that the growth is largely a function of the calculation radius. The vertical lines that form at $t = 1$ in figure 5.3 are $2R + 1$ in length because the calculation

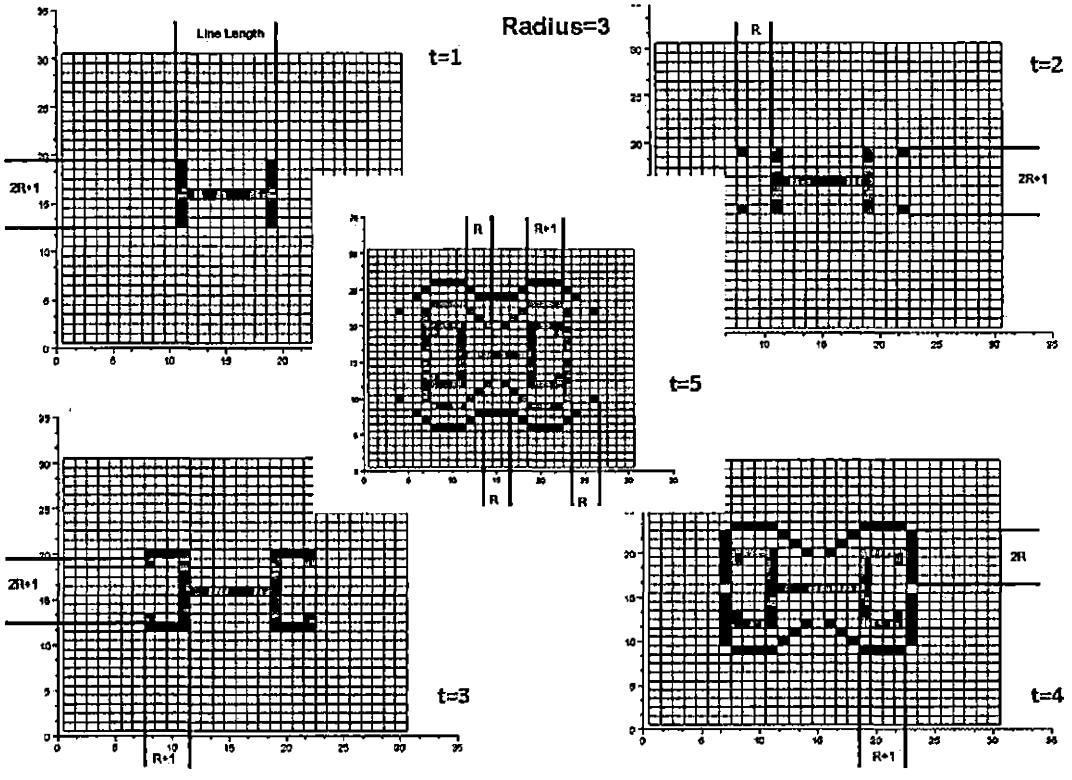

Fig. 5.3: Scaled pattern $radius = 3$, $birth = 4$, time steps 1-5

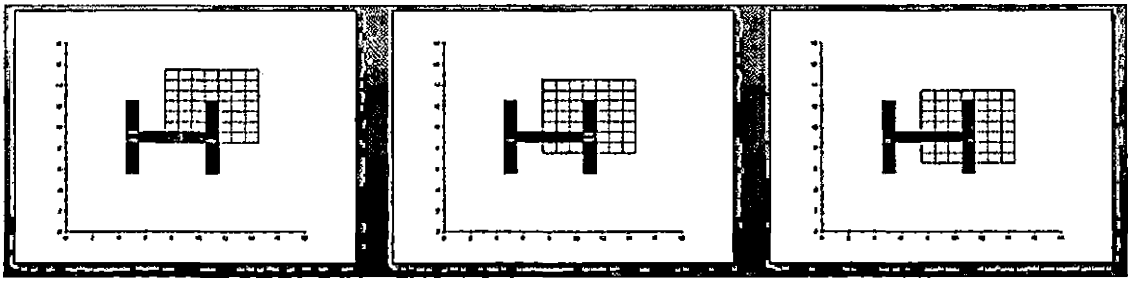

Fig. 5.4: Neighborhood views $radius = 3$, $birth = 4$, $t = 1$, (target squares are in red for clarity)

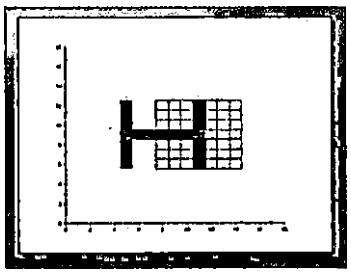

Fig. 5.5: Neighborhood center falls on the line, $Radius = 3$, $birth = 4$, $t = 1$

neighborhood can see the line from *radius* number of positions (shown in figure 5.4) , vertically, above the line start, and the same number of positions, vertically, below the line. When the center of the neighborhood, in figure 5.5, falls on the line, the endpoint of the line start already equals 1 ($Death = 0$ in this family of patterns), so that square stays the same. This center square is the $+1$ in the $2R + 1$ length of the vertical line. This occurs because the line start has an axis of symmetry along the line. Because there is another axis of symmetry perpendicular to the line and at its center point, this growth is mirrored on the other end of the line, as well. All of the dimensions shown in figures 5.2 and 5.3 are generated in a similar fashion. This manner of growth is at the heart of the scaling that we have observed in sequentially larger radii.

Looking at the $radius = 1$ entry in the table which starts on page 91, we see that there is no line length that will produce this pattern. If we look at the minimum line length for $radius = 1$, using equation (4.1), we see that the minimum line length for $radius = 1$ is 5, which exceeds the maximum line length as calculated by equation (4.2). In this case the minimum line length value is actually greater than the max-

Tab. 5.1: Scaling relationships between radii without death

radius	birth number	line length	line length *mod*2	radius+1	minimum line length	maximum line length
1	2	n/a	1	2	5	3
2	3	6	0	3	6	6
3	4	9	1	4	9	9
4	5	10	0	5	10	12
4	5	12	0	5	10	12
5	6	13	1	6	13	15
5	6	15	1	6	13	15
6	7	14	0	7	14	18
6	7	16	0	7	14	18
6	7	18	0	7	14	18
7	8	17	1	8	17	21
7	8	19	1	8	17	21
7	8	21	1	8	17	21
8	9	18	0	9	18	24
8	9	20	0	9	18	24
8	9	22	0	9	18	24
8	9	24	0	9	18	24
9	10	21	1	10	21	27
9	10	23	1	10	21	27
9	10	25	1	10	21	27
9	10	27	1	10	21	27

imum line length, so no line exists that can satisfy the value we need at this radius to form the pattern that we are trying to generate. For $radius = 2$, we note that the minimum line length and the maximum line length are the same, so there is only one line length that will generate the pattern.

$$MinimumLineLength = 2(R+1) + LineLength(mod2) \quad (5.1)$$

$$MaximumLineLength = 3R \quad (5.2)$$

Odd and Even Line Starts and the Resulting Effects on the Pattern

In the table, it becomes clear that the line lengths for even numbered radii are even numbers, and those for odd numbered radii are odd numbers. In the patterns, this is compensated for in the gaps in the center, between the interior lines that form at $t = 4$. In the $radius = 3$ figure 5.3 and in the $radius = 4$ figure 5.2 the difference between odd and even radius can be seen. The gap is always an odd number in odd numbered radii, and an even number in even numbered radii. The size of the gap may change in radii which can form multiple versions of the patterns in this family, but the gap always corresponds to the evenness, or oddness of the pattern's radius.

Additionally, this scaling effect explains the increasing size of the patterns that we observed as we increased the size of the calculation radius.

5.3 Conclusions

In previous chapters we have seen how some of the variables in cellular automata have independent effects on the resulting patterns. The radius of calculation affects the size of the resulting pattern due to the number of neighborhood positions that a given square or squares can be seen from. Each of these positions generate life in the center square of the neighborhood if the rules are satisfied. The birth number is responsible for the patterns of growth for the basic start patterns, line, square (and single square), and diagonal line. In this chapter we have investigated an entirely different family of cellular automata with three variables: radius, birth number, and initial start line length. These three variables work together to generate a family or system of patterns. Change any one of the variables and the pattern is broken. There are mathematical relationships between the variables. In families of cellular automata like these, the variables are no longer independent, they are each dependent on the others.

6. CONCLUSION

6.1 Review of Contribution

The contributions in this thesis stem largely from the the approach we took to experimental design. Cellular automata and other complex systems are non-linear systems. We chose to analyze a small corner of a very large and complex area of study. We chose to use a fractional factorial design [28] for the suite of experiments that we ran. Much of the insight into growth behavior that we derived from these experiments came from the first five to 10 time steps of pattern development. The outcome of these choices was an analysis of growth behavior with a very limited set of variables, to which we added variables one at a time, and then took them out before adding a different variable into the mixture. This allowed us to isolate which variables, or combinations of variables produced certain behaviors. This experimental perspective was integral to the process that allowed us to identify the growth behaviors of elementary CAs, which in turn, gave us the information that we needed generate a grammar using the Lindenmayer system as a framework on which to build it. We were able to isolate birth number as the driving variable in this grammar, and to show that the general pattern of grow was consistent with $birth = 1$ grammar for some patterns and analogous to $birth = 1$ for the rest.

The classification of of growth behavior, in turn, led us to the idea that the cal-

culation neighborhood was not the only way to scale patterns. The variables in the preliminary experiments that classified growth behavior affected pattern development independently. In the chapter, The Effect of Scaling on Developing Patterns, three variables, initial start size, birth number, and calculation radius interact to create a family of nearly identical patterns that span calculation radii from $radius = 2$ up, and birth numbers from $birth = 3$ up. We identified a lower limit to these values, and suspect that there is no upper limit (we tested through $radius = 9$, and $birth = 10$. We identified the functions that determined the minimum and maximum line length to generate this pattern, in terms of the calculation radius, and have shown, given the line length and birth number, that the pattern development is a function of calculation radius. We have also identified the fact that these CAs have an odd and even requirement in the start line length depending on whether the calculation radius is odd or even.

6.2 Future Research

There are a number of areas for future research in cellular automata, and using cellular automata to model biological mats. In the area of cellular automata:

1. One area of future research is further investigation into L-systems and cellular automata. Since L-systems is an array rewriting grammar, further investigation in the use of L-system grammar in cellular automata with death rules.

2. Investigate the relationship between grammar and display grid. Do number of grammar rules relate to number of lines of symmetry on display grid? Do other grid rules relates to square grids? How does the choice of neighborhood affect

the resulting rules of growth?

3. We have, when using weighted calculations, observed that the patterns consistently formed regular polygons. In these polygons, the number of sides were equal to the *birth* number times four. We believe that the times four in this behavior is due to the square grid that we are using. We conjecture that this is the source of the increasingly rounded and organic appearance that patterns display as the radius of calculation increases. The core of the conjecture is that, at $radius = 5$, the polygons would have twenty sides, and, at $radius = 6$, polygons would have twenty four sides. Polygons with this many sides are indistinguishable from circles, at the scale and resolution that we normally use with CAs. What are the driving factors in this behavior? Does it change on a different type of grid?

In the area of biological modeling with cellular automata, we have identified a number of areas of interest for future research:

1. Biological mats in caves have been easier to model because they are not subject to the amount of erosion typically seen on soil crusts. Two potential areas of investigation are the use of deblurring filters on digital photographs of soil crusts to reverse some of the effects of erosion, and the use of blurring filters in the modeling process itself to simulate the effects of erosion.

2. All of the cellular automata we have investigated in this thesis are totalistic (they update simultaneously). Random updating of cells in models to simulate the random availability of resources (water and nutrients) could be especially

useful in models of resource limited ecologies like deserts, the arctic, and even Mars.

3. In areas that are being transformed from grasslands to deserts, the local ecologies are changing from a homogeneous eco-systems created by the grass (which needs a fairly dense neighborhood population to survive) to islands of non-homgeneous ecosystems occupied by plants grow best in isolation from other plants. Creosote, for example, poisons the ground around itself to enforce spacing. Can we model this behavior with CAs? Perhaps we can model it by using different radii of calculation for different eco-systems? What might be most interesting is, what happens at the intersection of two very different ecological niches?

4. Currently, expert recognition is the only method available for determining cellular automaton rules from images. Melissa Quintana has developed a technique which uses image processing and histogram analysis of images to determine the birth and death rules, and the appropriate calculation radius for the corresponding cellular automaton model.

APPENDIX A
SCILAB CODE

Listing 1.1: Scilab Test Code for Cellular Automata

```
// Jane Curnutt
// CA testing program
// Thesis
// Fall 2010

clear;

// Size of Field of View
right=9;
bottom=9;

//Calculation Radius
rad=1;

// Time Steps
time=1;

// Black and White Plot Set Up
N=zeros(bottom,right);
Nfig=scf();
Ncmap=[1 1 1; 0 0 0];
Nfig.color_map=Ncmap;
Nfig.figure_size=[700,700];
Nfig.figure_position=[0,0];

// Color Plot Set Up
C=4*(ones(bottom,right));
Cfig=scf();
Ccmap=[1 0 0; 0 0 1; 0 1 0; 1 1 1];
Cfig.color_map=Ccmap;
Cfig.figure_size=[500,500];
Cfig.figure_position=[500,0];

// INITIAL CONDITIONS

// A: row of 7 black squares in center
//N(bottom/2,(right/2)-3) = 1;
//N(bottom/2,(right/2)-2) = 1;
//N(bottom/2,(right/2)-1) = 1;
//N(bottom/2,right/2) = 1;
//N(bottom/2,(right/2)+1) = 1;
//N(bottom/2,(right/2)+2) = 1;
//N(bottom/2,(right/2)+3) = 1;

// B: hollow 4x4 square
//top
//N((bottom/2)+2,(right/2)+1) = 1;
//N((bottom/2)+2,(right/2)-1) = 1;
//N((bottom/2)+2,(right/2)) = 1;
//N((bottom/2)+2,(right/2)+2) = 1;
```

```
//sides
//N((bottom/2)+1,(right/2)+2) = 1;
//N((bottom/2)+1,(right/2)-1) = 1;
//N((bottom/2),(right/2)+2) = 1;
//N((bottom/2),(right/2)-1) = 1;
//bottom
//N((bottom/2)-1,(right/2)+1) = 1;
//N((bottom/2)-1,(right/2)-1) = 1;
//N((bottom/2)-1,(right/2)) = 1;
//N((bottom/2)-1,(right/2)+2) = 1;

// C: delta wing (corner)
//top row
//N((bottom/2)-2,(right/2)+1) = 1;
//N((bottom/2)-2,(right/2)-1) = 1;
//N((bottom/2)-2,(right/2)+2) = 1;
//N((bottom/2)-2,(right/2)) = 1;
//row 2
//N((bottom/2)-1,(right/2)) = 1;
//N((bottom/2)-1,(right/2)-1) = 1;
//row 3 & 4
//N((bottom/2),(right/2)-1) = 1;
//N((bottom/2)+1,(right/2)-1) = 1;

// F:scalable filled square
r=round(right/2);
b=round(bottom/2);
N(r-rad:r+rad,b-rad:b+rad)=ones(2*rad+1,2*rad+1);

// H: Big Triangle
//r=round(right/2);
//b=round(bottom/2);
//N(r+4,b-4:b+4)=1;
//N(r+3,b-4:b+3)=1;
//N(r+2,b-4:b+2)=1;
//N(r+1,b-4:b+1)=1;
//N(r,b-4:b)=1;
//N(r-1,b-4:b-1)=1;
//N(r-2,b-4:b-2)=1;
//N(r-3,b-4:b-3)=1;
//N(r-4,b-4)=1;

//I: 4x4 Hollow Cross
//N(bottom/2+2,right/2:right/2+1) = 1;
//N(bottom/2-1,right/2:right/2+1) = 1;
//N(bottom/2:bottom/2+1,right/2-1) = 1;
//N(bottom/2:bottom/2+1,right/2+2) = 1;


// Point
//N(bottom/2,right/2) = 1;
```

```
//Diagonal
//N((bottom/2)-3,(right/2)-3)=1;
//N((bottom/2)-2,(right/2)-2)=1;
//N((bottom/2)-1,(right/2)-1)=1;
//N((bottom/2),(right/2))=1;
//N((bottom/2)+1,(right/2)+1)=1;
//N((bottom/2)+2,(right/2)+2)=1;
//N((bottom/2)+3,(right/2)+3)=1;

// Diagonal for Neighborhood Grid
//N((bottom/2)-1,(right/2)-1)=1;
//N((bottom/2),(right/2))=1;
//N((bottom/2)+1,(right/2)+1)=1;
//N((bottom/2)+2,(right/2)+2)=1;
// END iNITIAL SEED

temp=N;

// RULES and SUM CALCULATION
for t=1:time
  xinfo("iter: "+string(t));
 for i=2:bottom-1
    // inside square
    for j=2:right-1
      //Moore neighborhood
      neighbors =(2*rad+1)^2-1;
      BorderSum=round(sum(N(i-rad:i+rad,j-rad:j+rad)));
// BIRTH RULE - SINGLE VALUE
     if BorderSum== 1 temp(i,j)=1;
// DEATH RULE - RANGE OF VALUES
        elseif BorderSum >3  then temp(i,j)=0;
        end

 // DEATH IN THIS TIME STEP - RED
      if N(i,j)==1 & temp(i,j)==0 then C(i,j)=1;
      end
// NEW GROWTH IN THIS TIME STEP - BLUE
      if N(i,j)==0 & temp(i,j)==1 then C(i,j)=2;
      end
// RESIDUAL GROWTH from PREVIOUS TIME STEP - GREEN
      if N(i,j)==1 & temp(i,j)==1 then C(i,j)=3;
      end
// EMPTY SQUARE - WHITE
      if N(i,j)==0 & temp(i,j)==0 then C(i,j)=4;
        end
    end
end

N = temp;
```

```
// Black and White Plot
 scf(Nfig);
 clf();
  Matplot(1+N);

// Color Plot
  scf(Cfig);
  clf();
  Matplot(C);
end

// DRAW BACKGROUND GRID
 drawlater();
      for i=0:(right)
        plot([i+0.5,i+0.5],[0.5,bottom+0.5]);
        plot([0.5,right+0.5],[i+0.5,i+0.5]);
      end
  drawnow();


// NEIGHBORHOOD GRID
// Lower Left Corner
      x=1;
      y=5;

// Length of 1 Side of Neighborhood
      radN=2*rad+1;

// Draw Neighborhood Grid
          drawlater();
       for i=x:x+radN
//           vertical lines
           plot([i+0.5,i+0.5], [y+0.5, y+radN+0.5],'black');
        end
       for j= y:y+radN
//          horizontal lines
          plot([x+0.5,x+0.5+radN],[j+0.5,j+0.5],'black');
          end
         drawnow();

end
```

REFERENCES

[1] Catena vol. 37, 1999. Special issue on banded vegetation.

[2] Russ Abbott. Emergence explained: Abstractions: Getting epiphenomena to do real work: Essays and commentaries. *Complex.*, 12(1):13–26, 2006.

[3] P. Boston, J. Curnutt, E. Gomez, K. Schubert, and B. Strader. Patterned Growth in Extreme Environments. In *Proceedings of the Third IEEE International Conference on Space Mission Challenges for Information Technology*, pages 221–226. IEEE Press, July 2009.

[4] P. Boston, J. Curnutt, E. Gomez, Keith Schubert, D. E. Northrup, H. Sun, and C. P. McKay. Biovermiculations, mathematical modeling of complex biological and physical processes in mazelike biomats. Geological Society of America, October 2009.

[5] P.J. Boston. *Encyclopedia of Cave and Karst Science*, chapter Extraterrestrial Caves, pages 355–358. Fitzroy-Dearborn Publishers, Ltd., London, UK, 2004.

[6] P.J. Boston. personal communication, 2007.

[7] P.J. Boston, L.D. Hose, D.E. Northup, and M.N. Spilde. The microbial communities of sulfur caves: A newly appreciated geologically driven system on earth and potential model for mars. In R. Harmon, editor, *Karst Geomorphology, Hydrology, & Geochemistry*, pages 331–344. Geological Soc. Amer., 2006.

[8] P.J. Boston, M.N. Spilde, D.E. Northup, L.A. Melim, D.S. Soroka, L.G. Kleina,

K.H. Lavoie, L.D. Hose, L.M. Mallory, C.N. Dahm, L.J. Crossey, and R.T. Schelble. Cave biosignature suites: Microbes, minerals and mars. *Astrobiology Journal*, 1(1):25–55, 2001.

[9] Paul Callahan. Patterns, programs, and links for conway's game of life. `http://www.radicaleye.com/lifepage/`, 1995.

[10] Lewis Carroll. *Alice's Adventures in Wonderland.* Project Gutenberg, plain text edition, May 2009.

[11] I.D. Couzin, J. Krause, R. James, G.D. Ruxton, and N.R. Franks. Collective memory and spatial sorting in animal groups. *J. Theor. Biol.*, (218):1–11, 2002.

[12] J. Curnutt, E. Gomez, and K.E. Schubert. Patterned Growth in Extreme Environments. In Dan Werthimer, Karen Meech, Janet Siefert, and Michael Mumma, editors, *Proceedings of Bioastronomy-2007: Molecules, Microbes and ExtraTerrestrial Life*, San Juan, Puerto Rico, July 16-20 2007. Astronomical Society of the Pacific Conference Series.

[13] G.E. Cushing, T.N. Titus, J.J. Wynne, and P.R. Christensen. Themis observes possible cave skylights on mars. *Geophysical Research Letters*, 34(L17201), 2007.

[14] D'Arcy Thompson. *On Growth and Form.* Dover, dover edition edition, 1992.

[15] Barry E. DiGregorio. Rocky road to life. *The New Scientist*, 205(2747):40 – 43, 2010.

[16] D.L. Dunkerley. Banded vegetation: Development under uniform rainfall from a simple cellular automaton model. *Plant Ecol.*, (129):103–111, 1997.

[17] M. Gardner. The fantastic combinations of John Conway's new solitaire game, 'Life'. *Scientific American*, 223:120–123, 1970.

[18] R. HilleRisLambers, M. Rietkerk, F. van den Bosch, H.H.T. Prins, and H. de Kroon. Vegetation pattern formation in semi-arid grazing systems. *Ecology*, (82):50–62, 2001.

[19] Michael Hiltzik. *Dealers of Lightning, Xerox PARC and the Dawn of the Computer Age.* Harper Collins, first edition, 1999.

[20] D.J. Hoare, I.D. Couzin, J.-G.J. Godin, and J. Krause. Context-dependent group size choice in fish. *Animal Behavior*, 67:155–164, 2004.

[21] P. Hogeweg. Cellular automata as a paradigm for ecological modeling. *Appl. Math. Comput.*, 27(1):81–100, 1988.

[22] L.D. Hose, A.N. Palmer, M.V. Palmer, D.E. Northup, P.J. Boston, and H.R. Duchene. Microbiology and geochemistry in a hydrogen sulphide-rich karst environment. *Chemical Geology*, 169:399–423, 2000.

[23] Louise D. Hose, Arthur N. Palmer, Margaret V. Palmer, Diana E. Northup, Penelope J. Boston, and Harvey R. DuChene. Microbiology and geochemistry in a hydrogen-sulphide-rich karst environment. *Chemical Geology*, 169(3-4):399 – 423, 2000.

[24] C.A. Klausmeier. Regular and irregular patterns in semiarid vegetation. *Science*, (284):1826–1828, 1999.

[25] J. Krause and R.W. Tegeder. The mechanism of aggregation behavior in fish shoals: Individuals minimize approach time to neighbours. *Anim. Behav.*, (48):353–359, 1994.

[26] E. Meron, E. Gilad, J. von Hardenberg, M. Shachak, and Y. Zarmi. Vegetation patterns along a rainfall gradient. *Chaos, Solutions and Fractals*, (19):367–376, 2004.

[27] M. Mitchell. *Complexity: A Guided Tour.* Oxford University Press, 2009.

[28] M. J. Moroney. *Facts from Figures.* Pelican Books, second edition, 1974.

[29] NASA. Spirit. online, August 2005. Press release, 8/19/2005.

[30] M. Nowak. *Evolutionary Dynamics: Exploring the Equations of Life.* Belknap Press of Harvard University Press, 2006.

[31] T.J. Pitcher, O.A. Misund, A. Fernö, B. Totland, and V. Melle. Adaptive behaviour of herring schools in the norwegian sea as revealed by high-resolution sonar. *ICES Journal of Marine Science*, (53):449–452, 1996.

[32] Open Directory Project. Open directory of life. `http://www.dmoz.org/Computers/Artificial_Life/Cellular_Automata/Conway's_Game_Of_Life`, year = 2007,.

[33] P. Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[34] M. Reitkerk, Stefan C. Dekker, Peter C. de Ruiter, and Johan van de Koppel. Self-organized patchiness and catastrophic shifts in ecosystems. *Science*, 305(5692):1926–1929, September 2004.

[35] E. Reitman. *Exploring the Geometry of Nature.* Windcrest Books, 1989.

[36] Palash Sarkar. A brief history of cellular automata. *ACM Comput. Surv.*, 32(1):80–107, 2000.

[37] N. J. Savill and P. Hogeweg. Competition and dispersal in predator-prey waves. *Theoretical Poplulation Biology*, 56:243–263, 1999.

[38] William H. Schlesinger, James F. Reynolds, Gary L. Cunningham, Laura F. Huenneke, Wesley M. Jarrell, Ross A. Virginia, and Walter G. Whitford. Biological feedbacks in global desertification. *Science*, 247(4946):1043–1048, 1990.

[39] K. Schubert. Keith on numerical analysis. www.r2labs.org, February 2008.

[40] K. Schubert, E. Gomez, J. Curnutt, and P. Boston. To live and die in ca. In H. Arabnia and Q. Tran, editors, *Proceedings of the 2010 International Conference on Bioinformatics and Computational Biology*, volume 2, pages 662–665, July 2010.

[41] J. Shi. Partial differential equations and mathematical biology. Online.

[42] Stephen Silver. Stephen Silver's Life Page. `http://www.argentum.freeserve.co.uk/life.htm`, 2005.

[43] Sipper, Moshe, and Reggia. Self replication on a chessboard. *Scientific American*, 285(2):26–35, August 2001.

[44] Alvy Ray Smith. *Cellular Automata Theory.* PhD thesis, Stanford University, December 1969.

[45] Alvy Ray Smith. Cellular automata and formal languages. In *SWAT '70: Proceedings of the 11th Annual Symposium on Switching and Automata Theory (swat 1970)*, pages 216–224, Washington, DC, USA, 1970. IEEE Computer Society.

[46] Alvy Ray Smith. Cellular automata complexity tradeoffs. *Information and Control*, 18(5):466–482, July 1971.

[47] Alvy Ray Smith. Two-dimensional formal languages and pattern recognition by cellular automata. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:144–152, 1971.

[48] Alvy Ray Smith. Digital paint systems: An anecdotal and historical overview. *IEEE Annals of the History of Computing*, 23:4–30, 2001.

[49] Alvy Ray Smith, III. Real-lime language recognition by one-dimensional cellular automata. *J. Comput. Syst. Sci.*, 6(3):233–253, 1972.

[50] M. N. Spilde, D. E. Northup, V. J.Polyak, P. J. Boston, A. Clement, and K. J. Huges. Snowy river: A new discovery from an old cave. In *Geologica Society of Amercia Abstracts with Programs*, volume 41, page 399. Geological Society of America, 2009.

[51] B. Strader. Simulating spatial partial differential equations with cellular automata. Master's thesis, CSU San Bernardino, San Bernardino, CA, USA, December 2008.

[52] B. Strader, K. Schubert, E. Gomez, J. Curnutt, and P. Boston. Simulating spatial partial differential equations with cellular automata. In Hamid R. Arabnia and Mary Qu Yang, editors, *Proceedings of the 2009 International Conference on Bioinformatics and Computational Biology*, volume 2, pages 503–509, July 2009.

[53] B. Strader, K.E. Schubert, E. Gomez, J. Curnutt, and P. Boston. *Software Tools and Algorithms for Biological Systems*, chapter Estimation, Modeling, and Simulation of Patterned Growth in Extreme Environments. Advances in Experimental Medicine and Biology, AEMB. Springer, 2010.

[54] Jason Summers. Jason's life page. `http://entropymine.com/jason/life/`.

[55] J. Thiéry, J.M. d'Herbès, and C. Valentin. A model simulating the genesis of banded vegetation patterns in niger. *J. Ecol.*, (83):497–507, 1995.

[56] Tommaso Toffoli. *Cellular Automata Mechanics.* PhD thesis, Ann Arbor, MI, USA, 1977.

[57] J. von Hardenberg, E. Meron, M. Shachak, and Y. Zarmi. Diversity of vegetation patterns and desertifcation. *Phys Rev Lett*, (87:198101-14), 2001.

[58] Stephen J. Willson. On convergence of configurations.

[59] S. Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, (T9):170–183, 1985.

[60] S. Wolfram. *A New Kind of Science.* Wolfram Media, January 2002.

[61] Will Wright and Brian Eno. Will wright and brian eno play with time. online video, June 2006. The Long Now Foundation.

[62] Konrad Zuse. *Rechnender Raum.* MIT Technical Translation, online edition, 1969.