Modeling Cellular Development using L-Systems

André Stauffer and Moshe Sipper

Logic Systems Laboratory, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland. E-mail: {name.surname}@di.epfl.ch, Web: http://lslwww.epfl.ch.

Abstract. A fundamental process in nature is that of ontogeny, whereby a single mother cell—the zygote—gives rise, through successive divisions, to a complete multicellular organism. Over the years such developmental processes have been studied using different models, two of which shall be considered in this paper: L-systems and cellular automata. Each of these presents distinct advantages: L-systems are naturally suited to model growth processes, whereas if one wishes to consider physical aspects of the system, e.g., as pertaining to actual implementation in hardware, then an inherently spatial model is required—hence the cellular automaton. Our goals herein are: (1) to show how L-systems can be used to specify growing structures, and (2) to explore the relationship between L-systems and cellular automata. Specifically, we shall consider the case of membrane formation, whereby a grid of artificial molecules is divided into cells.

1 Introduction

A fundamental process in nature is that of ontogeny, whereby a single mother cell—the zygote—gives rise, through successive divisions, to a complete multicellular organism, possibly containing trillions of cells (e.g., in humans). Studying this process of cellular development is interesting both from a biological standpoint, wherein we wish to advance our understanding of ontogeny in nature, as well as from an engineering standpoint, wherein we wish to build better machines, inspired by such natural processes [6].

Over the years such developmental processes have been studied using different models, two of which shall be considered in this paper: L-systems and cellular automata. Introduced almost three decades ago as a mathematical theory of plant development, L-systems capture the essence of growth processes [2]. Basically, an L-system is a string-rewriting grammar that is coupled with a graphical interpretation—the system can be used to churn out a plethora of finite strings that give rise (through the graphical interpretation) to one-, two-, or three-dimensional images.

Cellular automata (CA) are dynamical systems in which space and time are discrete. A cellular automaton consists of an array of cells, each of which can be in one of a finite number of possible states, updated synchronously in discrete time steps, according to a local, identical interaction rule. The state of a cell at the next time step is determined by the current states of a surrounding neighborhood of cells. This transition is usually specified in the form of a rule table, delineating

M. Sipper et al (Eds.): ICES'98, LNCS 1478, pp. 196-217, 1998.

the cell's next state for each possible neighborhood configuration. The cellular array (grid) is n-dimensional, where $n = 1, 2, 3$ is used in practice [10, 13]. A one-dimensional CA is illustrated in Figure 1 (based on Mitchell [7]).

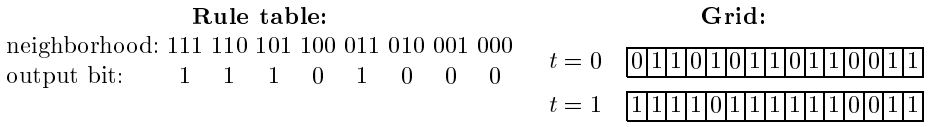

Fig. 1. Illustration of a one-dimensional, 2-state CA. The connectivity radius is $r = 1$, meaning that each cell has two neighbors, one to its immediate left and one to its immediate right. Grid size is $N = 15$. The rule table for updating the grid is shown to the left. The grid configuration over one time step is shown to the right. Spatially periodic boundary conditions are applied, meaning that the grid is viewed as a circle, with the leftmost and rightmost cells each acting as the other's neighbor.

Each of the above models presents distinct advantages. L-systems are naturally suited to model growth processes such as cellular development. On the other hand, if one wishes to consider physical aspects of the system, e.g., as pertaining to actual implementation in hardware, then an inherently spatial model is required—hence the CA.

Our goals herein are: (1) to show how L-systems can be used to specify growing structures, and (2) to explore the relationship between L-systems and CAs. Specifically, we shall consider the case of membrane formation, whereby a grid of artificial molecules is structured into cells. We begin in Section 2 with an introduction to L-systems. Section 3 demonstrates how a number of elemental developmental mechanisms, whose physical embodiment is that of a CA, can be described by L-system rewriting rules. Section 4 delineates the modeling of membrane formation using an L-system, followed by its implementation as a two-dimensional CA. Finally, we end with concluding remarks in Section 5.

2 L-systems

Lindenmayer systems—or L-systems for short—were originally conceived as a mathematical theory of plant development [2, 9]. The central concept of L-systems is that of rewriting, which is essentially a technique for defining complex objects by successively replacing parts of a simple initial object using a set of *rewriting rules* or *productions*. The most ubiquitous rewriting systems operate on character strings. Though such systems first appeared at the beginning of this century [9], they have been attracting wide interest as of the 1950s with Chomsky's work on formal grammars, who applied the concept of rewriting to describe the syntactic features of natural languages [1]. L-systems, introduced by Lindenmayer [2], are string-rewriting systems, whose essential difference from Chomsky grammars lies in the method of applying productions. In Chomsky grammars

productions are applied sequentially, whereas in L-systems they are applied in parallel and simultaneously replace all letters in a given word. This difference reflects the biological motivation of L-systems, with productions intended to capture cell divisions in multicellular organisms, where many divisions may occur at the same time.

As a simple example, consider strings (words) built of two letters, A and B. Each letter is associated with a rewriting rule. The rule $A \rightarrow AB$ means that the letter A is to be replaced by the string AB, and the rule $B \rightarrow A$ means that the letter B is to be replaced by A [9]. The rewriting process starts from a distinguished string called the *axiom*. For example, let the axiom be the single letter B. In the first derivation step (the first step of rewriting), axiom B is replaced by A using production $B \rightarrow A$. In the second step, production $A \rightarrow AB$ is applied to replace A with AB. In the next derivation step both letters of the word AB are replaced *simultaneously:* A is replaced by AB and B is replaced by A. This process is shown in Figure 2 for four derivation steps.

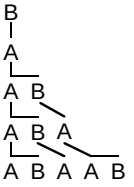

Fig. 2. Example of a derivation in a context-free L-system. The set of productions, or rewriting rules is: $\{A \rightarrow AB, B \rightarrow A\}$. The process is shown for four derivation steps.

In the above example the productions are context-free, i.e., applicable regardless of the context in which the predecessor appears. However, production application may also depend on the predecessor's context, in which case the system is referred to as context-sensitive. This allows for interactions between different parts of the growing string (modeling, e.g., interactions between plant parts). Several types of context-sensitive L-systems exist, one of which we shall concentrate on herein. In addition to context-free productions (e.g., $A \rightarrow AB$), context-sensitive ones of the form $U{<}A{>}X \rightarrow DA$ are introduced, where the letter A (called the strict predecessor) can produce word DA if and only if A is preceded by letter U and followed by X. Thus, letters U and X form the context of A in this production. When the strict predecessor has a one-sided context, to the left or to the right, then only the $<$ or $>$ symbol is used, respectively (e.g., $U{<}A \rightarrow DA$ is a left-context rule and $A{>}X \rightarrow DA$ is a right-context one). Figure 3 demonstrates a context-sensitive L-system. We note that, defining a growth function as one describing the number of symbols in a word in terms of its derivation length, then this L-system exhibits square-root growth: after n derivation steps the length of the string (X symbols excluded) is $\lfloor \sqrt{n} \rfloor + 2$. Other growth functions can also be attained, including polynomial, sigmoidal, and exponential [9].

```
p1: U<A>A -> U
p2: U<A>X -> DA
p3: A<A>D -> D
p4: X<A>D -> U
p5:   U   -> A
p6:   D   -> A
```

(a)

```
0: XUAX     5: XADAAX
1: XADAX    6: XUAAAX
2: XUAAX    7: XAUAAX
3: XAUAX    8: XAAUAX
4: XAADAX   9: XAAADAX
```

(b)

Fig. 3. A context-sensitive L-system. (a) The production set. (b) A sample derivation. Note that if no rule applies to a given letter then that letter remains unchanged.

As noted above, L-systems were originally designed to model plant development. Thus, in addition to a grammar that produces finite strings over a given alphabet (as defined above), such a system is usually coupled with a graphical interpretation. Several such interpretations exist, one example of which is the so-called turtle interpretation, based on a LOGO-style turtle [9]. Here, the string produced by the L-system is considered to be a sequence of commands to a cursor (or "turtle") moving within a two- or three-dimensional space. Each symbol represents a simple command (e.g., `move forward`, `turn left`, `turn right`) such that interpretation of the string gives rise to an image.

In summary, there are two important aspects concerning L-systems, which shall serve us herein: (1) such a system gives rise to a growing, one-dimensional string of characters, (2) which can then be interpreted as a one-, two-, or three-dimensional image.

3 Using L-systems to describe cellular development

In this section we demonstrate how a number of basic components—or operations—related to cellular development can be modeled by L-systems. This developmental model involves four main processes or mechanisms: (1) simple growth, (2) branching growth, (3) signal propagation, and (4) signal divergence.

Simple growth arises from the application of productions p1 to p3 of Figure 4. In these productions the symbol a represents the apex and i the internode. The terminology introduced here is borrowed from the description of tree-like shapes [9]. A tree has edges that are labeled and directed. In the biological context, these edges are referred to as branch segments. A segment followed by at least one more segment is called an *internode*. A terminal segment (with no succeeding edges) is called an *apex*. In the productions, the ( ) and [ ] symbol pairs represent a left and right branch, respectively. These are used in so-called bracketed L-systems with the parentheses being a form of recursive application [9]: a string is interpreted from left to right to form the corresponding image. When a left bracket is encountered then the current position within the image is pushed onto a pushdown stack, with a right bracket signifying that a position is to be popped from the stack. Thus, one can model plants with branches, sub-branches, etc.,

or, in our case, create such constructs as multi-dimensional growing structures. The corresponding CA interpretation of a single derivation step of the simple growth productions is also given in Figure 4.

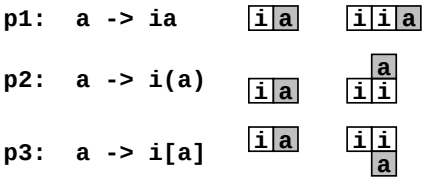

Fig. 4. Some simple growing structures along with their CA interpretation. The ( ) and [ ] symbol pairs represent a left and right branch, respectively. As the cellular space considered is a two-dimensional grid, the branching angle is 90°.

Productions p1 to p4 of Figure 5 give rise to *branching growth*. The figure also depicts the CA interpretation of a single derivation step. Note that in both Figures 4 and 5, all productions are context-free.

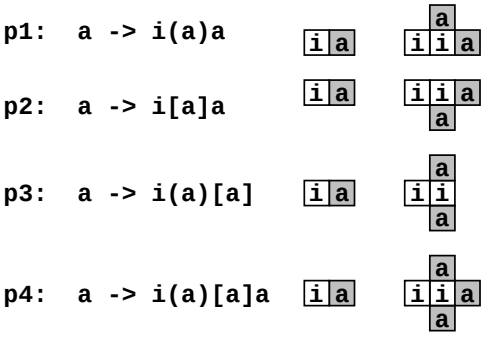

Fig. 5. Some branching structures along with their CA interpretation.

Signal propagation of a given signal s is modeled by productions p1 and p2 of Figure 6. The context-sensitive production p1 means that internode i with a signal s to its left becomes an s, while application of the context-free production p2 transforms signal s to an internode i. Consequently, the CA interpretation of a single derivation step of these productions causes signal s to move one cell to the right (Figure 6).

Productions p1 to p10 of Figure 7 model *signal divergence*, whereby a given signal s divides into three signals t, u, and v. Some of these productions are

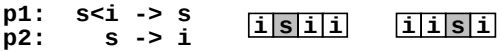

Fig. 6. Productions used to obtain signal propagation, along with their CA interpretation. The CA state s is propagated to the right.

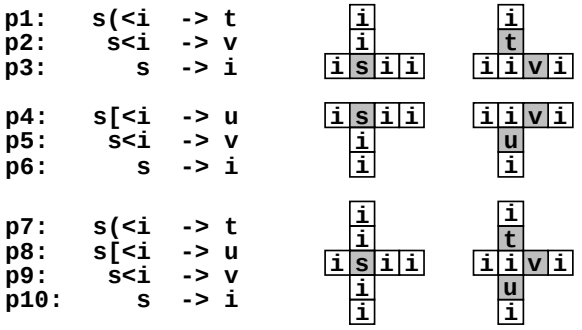

Fig. 7. Productions used to obtain signal divergence, along with their CA interpretation. These implement the cases of a given signal, i.e., CA state (denoted s) that breaks up to yield two or three new states (denoted t, u, and v).

context-sensitive. The corresponding CA interpretation of a single derivation step is also shown.

Using the components described above, as well as a number of others, we have previously shown how L-systems can be used to specify self-replicating structures, thereafter to be implemented as cellular systems [12]. The fabrication of artificial self-replicating machines has diverse applications, ranging from nanotechnology to space exploration. It is also an important aspect of the Embryonics project, described ahead. (A short survey of self-replication is provided in [8]; for detailed information see the online self-replication page at http://lslwww.epfl.ch/~moshes/selfrep/.) Below, we shall focus on another application of our approach, namely, membrane formation.

4 Membrane formation: The space divider

The Embryonics (Embryonic Electronics) project, under development at the Logic Systems Laboratory for the past four years, has as its ultimate objective the construction of large-scale integrated circuits, exhibiting properties such as self-repair (healing) and self-replication. It is based on three features usually associated with the ontogenetic process in living organisms, namely, multicellular organization, cellular differentiation, and cellular division [3–6]. An Embryonics

"organism" consists of a multitude of artificial cells, which are in turn composed of yet finer elements, referred to as molecules. The first operational phase of such a system consists of structuring a *tabula rasa* "sea" of identical molecules into cells, which are defined by their borders, or membranes. The space divider, described in this section, is a two-dimensional CA whose purpose is to structure such a sea of molecules. The structure of this molecular sea ultimately consists of squares of functional molecules—the cells. The structuring process is essentially a growth process, starting from the bottom-left CA cell, where the programming data is continuously fed. (Note: the term cell is used here in two different contexts, as it originates from two different sources. In a CA, the cell is the elemental unit, equivalent to an Embryonics *molecule*, whose cell is one level up, i.e., composed of an ensemble of molecules. Thus, one must take care to distinguish between a CA cell and an Embryonics cell—this can be inferred from the text. Within the framework of the Embryonics project it would probably be better to refer to cellular automata as molecular automata, however, in order to conform to existing literature we shall retain the extant terminology.)

As noted, L-systems are naturally suited for modeling growth processes. Therefore, we first describe the space divider as an L-system developmental model. The graphical interpretation will be that of a CA. The representation of the symbols (letters) used in the developmental model of the space divider is given in Figure 8.

```
h: horizontal apex
v: vertical apex
e: east growing apex
n: north growing apex
l: left branching apex
r: right branching apex
i: internode
b: branching signal
```

Fig. 8. Symbol representation of the space divider L-system model.

The space divider L-system starts out with a single-letter axiom l which corresponds to a left-branching apex. The productions applied to the axiom in order to obtain the square structure are listed in Figure 9a. They fall into three categories: (1) the branching signal propagation productions p1 to p4, (2) the simple growth productions p5 to p14, and (3) the branching growth productions p15 to p18.

The first character of the string—to the left of the vertical separator—is part of the program that is fed to the space divider. This comprises the input to the system—an artificial "genome" that determines the structure that the space will take on (this genome also contains additional information related to the ultimate task that the "organism" will carry out—see references [3, 6] for further details). This input mechanism is a novel element that we have added

```
p1:   b<i   -> b      p10:  i<v   -> n
p2:  b(<i   -> b      p11: i(<v   -> n
p3:  b[<i   -> b      p12:  i<n   -> iv
p4:    b    -> i      p13: i(<n   -> iv
p5:   i<h   -> e      p14:  b<n   -> bv
p6:  i[<h   -> e      p15:  b<h   -> l
p7:   i<e   -> ih     p16:    l   -> i(v)h
p8:  i[<e   -> ih     p17:  b<v   -> r
p9:   b<e   -> bh     p18:    r   -> i[h]v
```

(a)

```
0:  i|l
1:  i|i(v)h
2:  b|i(n)e
3:  i|b(iv)ih
4:  i|i(bn)be
5:  b|i(ibv)ibh
6:  i|b(iir)iil
7:  i|i(bii[h]v)bii(v)h
8:  b|i(ibi[e]n)ibi(n)e
9:  i|b(iib[ih]iv)iib(iv)ih
10: i|i(bii[be]bn)bii(bn)be
11: b|i(ibi[ibh]ibv)ibi(ibv)ibh
12: i|b(iib[iil]iir)iib(iir)iil
13: i|i(bii[bii(v)h]bii[h]v)bii(bii[h]v)bii(v)h
```

(b)

Fig. 9. Space divider L-system model. (a) The production set. (b) A sample derivation. Note: when applying the above productions to derive a string like $x_1|x_2(x_3x_4[x_5]x_6)x_7$ then the context of a letter x_i depends upon its position. Thus, the context letters x_{i_l} and x_{i_r} of $x_{i_l}<x_i>x_{i_r}$ are defined as follows: $x_1<x_2$, $x_2(<x_3$, $x_3<x_4$, $x_4[<x_5$, $x_4<x_6$, and $x_2<x_7$. For example, when deriving the string $a|b(cd[e]f)g$, the context of f is not the e to its immediate left but rather d. (Note that so as to simplify notation vertical separators were omitted from the production set of (a).) Formally, L-systems where the context of a letter can be further down the string are known as IL-systems or (k,l)-systems, meaning that the left context is a word of length k and the right context is a word of length l [9].

to the classical L-systems model, in which development from a given "seed" (axiom) takes place with no external intervention. This feature gives rise to a programmable cellular space, which can be programmed and structured via a signal from an external source. The genome character (leftmost letter of the string), input at each derivation step from the external source, comprises the left context of the letter to the immediate right of the vertical separator. Thirteen derivation steps of the developmental process are shown in Figure 9b. The CA interpretation of this process defines a division of the cellular space into squares of size 2 x 2 cells (Figure 10).

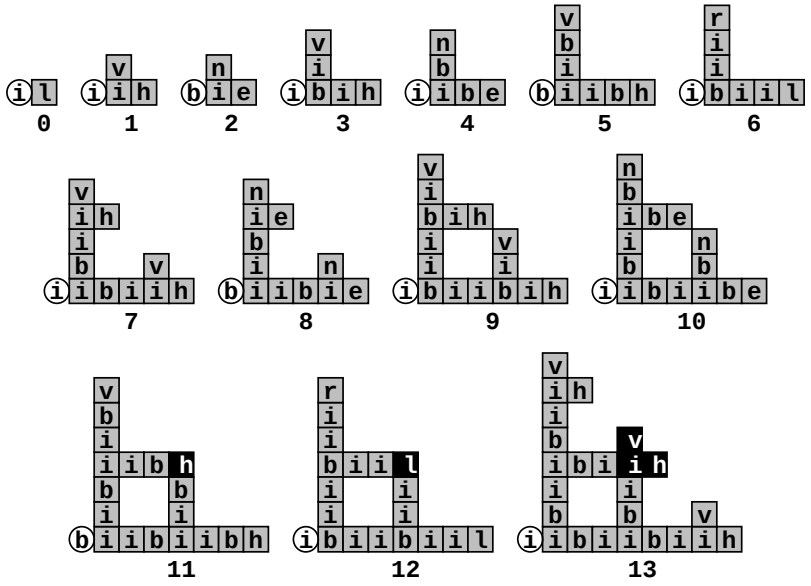

Fig. 10. CA interpretation of the space divider L-system model.

In Figure 10 the successive characters of the programming data (the genome) appear in the bottom-left circle. The membrane size, i.e., the number of cells per square side, is programmable and equals the number of internodes i between two successive branching signals b. The black squares of Figure 10 (at time steps 11, 12, and 13) correspond to the underlined string segments of Figure 9b (at derivation steps 11, 12, and 13, respectively). These represent two confluent branches, resulting from the closing of the square, which thus amount to the same CA state (or states).

The complete specification of the CA derived from the above L-system, as well as a hardware implementation using Field-Programmable Gate Arrays (FPGAs) is described by Stauffer and Sipper [11]. As noted, this comprises part of the Embryonics project, which is ultimately realized in hardware.

5 Concluding remarks

We have shown how L-systems can be used to specify growing structures, specifically concentrating on the case of membrane formation. The L-system is then transformed into a cellular automaton, an inherently spatial model which leads directly to a hardware implementation. The study of systems that exhibit growth is interesting both from a theoretical standpoint as well as from a practical one. This paper has shed light on the possible use of L-systems as an exploratory, and perhaps design tool within the realm of growth and development.

References

1. N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.
2. A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.
3. D. Mange, D. Madon, A. Stauffer, and G. Tempesti. Von Neumann revisited: A Turing machine with self-repair and self-reproduction properties. *Robotics and Autonomous Systems*, 22(1):35–58, 1997.
4. D. Mange, A. Stauffer, and G. Tempesti. Embryonics: A macroscopic view of the cellular architecture. In M. Sipper, D. Mange, and A. Pérez-Uribe, editors, *Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98)*, Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, 1998.
5. D. Mange, A. Stauffer, and G. Tempesti. Embryonics: A microscopic view of the molecular architecture. In M. Sipper, D. Mange, and A. Pérez-Uribe, editors, *Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98)*, Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, 1998.
6. D. Mange and M. Tomassini, editors. *Bio-Inspired Computing Machines: Toward Novel Computational Architectures*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1998.
7. M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.
8. J.-Y. Perrier, M. Sipper, and J. Zahnd. Toward a viable, self-reproducing universal computer. *Physica D*, 97:335–352, 1996.
9. P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.
10. M. Sipper. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Springer-Verlag, Heidelberg, 1997.
11. A. Stauffer and M. Sipper. L-hardware: Modeling and implementing cellular development using L-systems. In D. Mange and M. Tomassini, editors, *Bio-Inspired Computing Machines: Toward Novel Computational Architectures*, pages 269–287. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1998.
12. A. Stauffer and M. Sipper. On the relationship between cellular automata and L-systems: The self-replication case. *Physica D*, 116(1-2):71–80, 1998.
13. T. Toffoli and N. Margolus. *Cellular Automata Machines*. The MIT Press, Cambridge, Massachusetts, 1987.

This article was processed using the LaTeX macro package with LLNCS style