



Article Autopoiesis and Eigenform

Louis H. Kauffman D

Mathematics Department, University of Illinois at Chicago, 851 South Morgan Street, Chicago, IL 60607-7045, USA; loukau@gmail.com

Abstract: This paper explores a formal model of autopoiesis as presented by Maturana, Uribe and Varela, and analyzes this model and its implications through the lens of the notions of eigenforms (fixed points) and the intricacies of Goedelian coding. The paper discusses the connection between autopoiesis and eigenforms and a variety of different perspectives and examples. The paper puts forward original philosophical reflections and generalizations about its various conclusions concerning specific examples, with the aim of contributing to a unified way of understanding (formal models of) living systems within the context of natural sciences, and to see the role of such systems and the formation of information from the point of view of analogs of biological construction. To this end, we pay attention to models for fixed points, self-reference and self-replication in formal systems and in the description of biological systems.

Keywords: autopoiesis; molecular substrate; cells; protocells; recursion; fixed points; fixed point theorems; Goedelian self-reference; eigenform; infinity; finity; distinction; self-reference; recursive distinctions; cellular automata; artificial life

1. Introduction

In their paper "Autopoiesis: The organization of living systems, its characterization and a model", Umberto Maturana, Ricardo Uribe and Francisco Varela [1] discuss the meaning of self-organization in living systems, and they give a model for their ideas. The ideas behind this model are of great generality and power. In this paper I discuss this model of autopoiesis and how it can be generalized and understood in a multiplicity of contexts. We place the model of autopoiesis in the context of recursions and fixed points, and consider how this model is seen to emerge from a molecular substrate as part of how structure is seen to emerge from a recursive background.

Consideration of autonomous structures, starting from autopoiesis, strike deeply into the heart of issues of communication, understanding and information. This leads to new ways to think about formal systems, incompleteness and the role of Goedelian coding in the structure of biology, mathematics and human thought.

We mention at the outset that the core of this paper is not biological, but rather the explication of emergence of patterns in recursive domains, among which there are key biological examples. We will, later in the paper, discuss the formal structure of DNA replication as such an example. This is also an example of our method, where the actual biology is schematized in order to be compared with a mathematical model. In some cases, this makes the biological modeling unrealistic. In others, the schematic modeling can be applied to the actuality. This is the case in certain topological models for DNA recombination, as we will mention in Section 3.

We also point out that the phrase "the map is not the territory" applies in spades to the modeling of living systems. Whatever model may be devised to describe a living system, it can never play the role of the living system itself. The imagination of the model maker plays a significant role in forming the viewpoint that makes the model. A living system self-creates information in the process of engaging with the environment in which it is embedded. Nevertheless, we can see some of this process in mathematical models, and that



Citation: Kauffman, L.H. Autopoiesis and Eigenform. *Computation* 2023, *11*, 247. https://doi.org/10.3390/ computation11120247

Academic Editor: Demos T. Tsahalis

Received: 3 November 2023 Revised: 29 November 2023 Accepted: 29 November 2023 Published: 5 December 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). is one of the purposes of the present work. We are about to describe a simple autopoietic model where self-sustaining entities arise from a more neutral background substrate. We do not claim these models as biological models, but as formal analogs to aspects of biology.

Here is a short introduction to the concept of autopoiesis in the form of a quote from the linguistic AI system ChatGTP, recorded on 22 February 2023:

"According to the theory of autopoiesis, living systems are characterized by a set of processes that enable them to create and maintain their own structure and organization. These processes involve the constant production and renewal of the components that make up the system, as well as the interactions between these components.

At the heart of autopoiesis is the idea that living systems are self-producing, self-maintaining and self-reproducing, and that their organization is continually being created and recreated through their interactions with the environment. The theory emphasizes the importance of the system's boundaries or membrane, which separates it from the environment and enables it to maintain its own identity."

This quote is a summary of the main line of thought in the paper by Maturana, Uribe and Varela (MUV). Note that the process of autopoiesis allows the system to create and maintain a boundary with respect to its environment. In that regard, autopoiesis can be thought to be a description of how the dynamics of a form or a distinction can be brought forth into the world.

It is key that a new word, "autopoiesis", is coined for such self-sustaining systems. They need not be alive in the classical sense of biology, but biological systems are indeed seen to be autopoietic. In this paper, we will think of systems as abstract as patterns arising in a recursion to be autopoietic if the patterns satisfy those requirements in the eyes of an observer.

Let us go directly to the model constructed by MUV. In the model there are "molecules" in a plane space and special molecules called "catalysts". In the presence of a catalyst, the other molecules form bonds with one another. The catalyst cannot move past a bond. All the molecules are in motion. Bonds join in high probability near a catalyst, but nevertheless break with some probability as well.

The reader can imagine what will happen if a random assortment of jiggling molecules and catalysts is placed in this plane space.

Around a catalyst, the molecules tend to bond. A catalyst will tend to become surrounded by bonded molecules. Since the space is a plane space, these bonds that form around catalysts will form boundaries that surround the catalysts. Thus, prototypical cells arise, each containing catalysts within the cell boundary. The cells will tend to persist, even though their constituents are changing. I say the constituents are changing because the bonds emerge and decay, and the molecules are in continual motion.

A form of life has arisen in the plane space. The cells have emerged from the constraints and possibilities inherent in the substrate of molecules and catalysts.

A new order has assembled itself from an order that never designed the cells that emerged from it. Figure 1 illustrates the process, just as in the original paper.

Maturana, Uribe and Varela call an entity "autopoietic" if, by its own structure, it maintains that structure by using the elements of a "molecular substrate". The autopoietic entity maintains itself through its own interactions with its environment.

In the case of the MUV model, the protocells assemble themselves from the substrate. We may imagine such self-assembly at the base of biology, but in practice we do not see life emerging from non-life except in the presence of life. Thus, the model suggests more than is actually observed in biological science. We would like to know how biological molecules such as DNA emerge in actual biology. In working with mathematical models, we can see emergence of this kind in the patterns of models, but this is only suggestive of what may happen in actuality. In biological studies, there is the notion of autocatalysis, pioneered by Stuart Kauffman and his collaborators (Stuart Kauffman 1986). Stuart Kauffman's work and the points of view in this paper will be compared in a sequel to the present paper.

How can we describe the kind of closure that was experienced in the example of MUV? We are describing how a distinction can arise and become autonomous. The cell repairs itself in the sense that it tends to maintain its integrity over time. Nevertheless, the cell is not self-referential, since it has no language of reference.



Figure 1. Prototypical autopoiesis.

2. Autopoiesis and Eigenforms

In the MUV model for autopoiesis, we have a space, S, in which distinctions are possible (the plane space of the model), and there are entities called catalysts in the space. Linkages of other entities tend to occur in the neighborhood of a catalyst. The catalyst cannot move across the linkages. Linkages can decay with some probability. These conditions lead to the possibility that a linkage may form in the neighborhood of a catalyst so that the catalyst is captured inside a distinction (a corral) made by the linkages that occur in the neighborhood of the catalyst.

The conditions in the space lead to the production and maintenance of distinctions surrounding catalysts. These distinctions arise spontaneously due to the nature of the space, and they are maintained via the dynamics of linkage in the presence of the catalyst and decay of linkage as well.

There is a general theory of the arising of stable forms, called *eigenforms* [2,3]. In this theory, one is given a transformation, T, that can act on forms, X, to make new forms, TX. Repeated application of T on an initial form, X, will produce a concatenation of this operation applied to X: G = TTTTT...TTTX. One more application of T may make little difference to the shape of the form, so that

TG = G

where the = sign means that TG and G are indistinguishable from one another.

We call G the *eigenform* for T. At the point when there is identity TG = G for an observer of the system, we say that we have an eigenform. If the process is complex, as in autopoiesis, then some time steps will leave the form the same for the observer, and some time steps will have a difference associated with them. Indeed, some time steps can embody a disintegration of the form and a disappearance of the eigenform.

Here is a typographical example using the operator $TX = \langle X \rangle$. T puts brackets around X. If we start with $\langle \rangle$, and iterate T, we obtain

 J_n has *n* brackets around <>, and we see that $J_{n+1} = TJ_n$ has n + 1 brackets around <> and is distinct from J_n .

If, however, we allow a new expression with an *infinite* number of brackets,

```
J = <<<<<<...>>>>>,
```

then in the infinite nest of brackets we have $\langle J \rangle = J$. See Figure 2 for an illustration using nested boxes.



Figure 2. Iterated F with Nested Boxes.

Note that J satisfies TJ = J. If we start with an arbitrary variable, X, then TX is different from X, and the sequence of elements X, TX, TTX, ... are all different. We do not arrive at a solution (in general) to the equation X = TX except by taking this leap to infinity.

Remark 1. While we write J = <<<<<...>>>>, indicating an infinite nest of brackets, we do not give a formal definition here of the type of limit process that is intended to be performed to obtain the infinite nest. There is no actual infinite nest of brackets in the plane. For the purpose of our informal presentation, the reader can imagine this construction and explore how it can be compared with mathematics, which the reader is familiar with.

In mathematical situations, we can often define a limit so that the limiting "value" of the finite approximations goes to a fixed point. For example,

let Fx = 1 + 1/x. Then, nearly any non-zero starting value, x, will give you a set of numbers

$$x, 1 + 1/x, 1 + 1/(1 + 1/x), 1 + 1/(1 + 1/(1 + 1/x)), \dots$$

whose limiting value is the golden ratio **Phi = (1 + sqrt(5))/2** and can be represented by the infinite form

$$\mathbf{Phi} = 1 + 1/(1 + 1$$

Phi is a direct solution to **FPhi** = **Phi**. That is

$$Phi = 1 + 1/Phi$$

We can create special domains so that very general recursions have formal limits. Even in ordinary mathematics, the results are subtle. For example, Hx = 1 + x. Then, H1 = 2, HH1 = 3, HHH1 = 4 and so on. If HG = I, then 1 + G = I. This entity, G, is not an ordinary natural number since such numbers always increase if you add 1. Thus, finding an eigenform for H involves extending the number system to include "infinite" numbers (as was first done by Georg Cantor). Here, we are suggesting that one can play with these ideas and see what be constructed. You can behave as the observer of your own mathematical creations. For example, you could make up the "infinite number"

$$G = 1 + 1 + 1 + 1 + 1 + \dots$$

and agree that, while **G** cannot be any ordinary number, it makes sense to say that 1 + G = G as a formal pattern.

The pattern of nested boxes in Figure 2 is reminiscent of the observed patterns that we see between facing mirrors, as in Figure 3 (compare [2,4,5]).



Figure 3. Eigenform of facing mirrors.

Of course, one can ask, What is the meaning of an infinite nest of parentheses or an infinite nest of boxes? Certainly, it is not possible to actually write down on the page such an infinity. The ellipsis of the three dots in the notation <<<<...>>>> indicates that we imagine going on forever in a mathematical idealization of brackets within brackets. In formalizing such constructions there are different choices available, and it is not the purpose of this paper to set up such formalization. Rather we appeal to the reader's experience of

numbers and geometry to give the idea of an unending row or nest of identical objects. In the case of the infinite nest, J, it will help to think of J as a limit of nests of boxes, as in

```
<>,
<<>>,
<<<>>,
<<<>>>,
<<<<>>>>,
<<<<<>>>>,
<<<<<>>>>>,
```

As the number of brackets in the nests increases, it becomes more and more difficult to tell that there is a difference between one stage and the next without careful comparison. The infinite nest can be thought of as a very large finite nest that is too big for us to count. In that sense, the equation $J = \langle J \rangle$ is an indication that we are not able to tell the difference between J and $\langle J \rangle$.

Note that $J = \langle J \rangle$ is analogous to the fact that 1 = 0.9999... For if x = 0.9999... then 10x = 9.9999... = 9 + x, and so 9x = 9 and x = 1. When we multiply 0.9999... by 10 and find that 0.9999... has reappeared after shifting the decimal point, it is exactly analogous to the way $J = \langle \langle ... \rangle \rangle$ reappears when you put a bracket around it.

Here is another thought experiment. Consider the following rows of stars.

```
*
**
***
***
```

and let

K = *******...

be an infinite row of stars (with the same injunction to take this as a definite but not-yetformalized idea). Then we have * K = K exactly because there is no difference between * K and K. There is a difference between *** and **, but when the row is infinite then the difference disappears.

Here is another way to think about this. Suppose I tell you that I *postulate an entity*, K, that has the property that K = * K. This means that whenever you see K, you are justified in replacing it by * K. This is an instruction that we can give to a computer.

The facing mirrors of Figure 3 give a clue. It takes two mirrors to produce the infinity between the mirrors. Let S denote the function, F, as an active procedure. Let SS denote the infinite (not stopped) application of F to itself.

Then we have, as in Figure 2,

$$SS = F(F(F(F(F(...)))))$$
 (ad infinitum).

Thus, we have that SS is the eigenform and we can write

F(SS) = SS.

Now it is important at this point to stop and examine Figure 3, the photograph of the facing mirrors. The camera has given us the opportunity to see what we would have seen if we had been standing there between the mirrors. Being an observer is crucial in this experiment. There is no vision of the multiple reflections without an observer, or indirectly a camera that can make a record. You can even imagine two facing mirrors in the dark with no "photons" to bounce between them. So this phenomenon of the multiple nested mirrors needs an observer and does not occur without an observer. In this case, the "reality" of the mirror eigenform does not exist except in the presence of an observer. That this is true quite generally in our world of observation is an insight that can begin to expand from this example.

2.1. Producing an Eigenform

Let **G** be defined as an operator that duplicates x (forming xx) and places xx in a box (indicated by the brackets).

$$\mathbf{G}\mathbf{x} = [\mathbf{x}\mathbf{x}].$$

The adjacent xs that are duplicated can be any distinct things that can be placed next to one another. Thus

and so on.

But consider **GG**. We can take a copy of **G** itself, duplicate it and put it in the box.

In this sense I can apply **G** to "itself". I have to make an exact copy of **G** and then apply (the first) **G** to the copy. This is not problematical because **G** embodies a very simple rule that can be applied to any object, including itself!

It is sometimes said that there is a problem about applying a function to itself, but we are working in an unrestricted way, and **G**, by definition, can be applied to anything at all that can be duplicated. It is when you have restrictions on the domains of functions that there can be trouble.

It follows from the equation Gx = [xx] that

$$GG = [GG],$$

and so GG represents an eigenform for the boxing operator.

We have constructed this eigenform without going to infinity, and without taking any limits.

Since **GG** = **[GG]**, we certainly have that

and so the infinite concatenation, E = [[[[[....]]]]], and the finite composition, GG, have the same behavior to the outside mathematical observer.

$$\mathbf{E} = [\mathbf{E}] = [[\mathbf{E}]] = [[[\mathbf{E}]]] = \dots = [[[[[\dots [\mathbf{E}] \dots]]]]],$$

We can generalize this box fixed point by writing Gx = F(xx) for F representing (the form of) any transformation. Then GG = F(GG). This duplication trick (Gx = F(xx)) produces any eigenform we want. More about this method occurs in Section 4 when we discuss reflexive domains.

To see discussions of formalizations of limits for eigenforms, the reader can examine [6–9]. The duplication trick is the approach of Church and Curry, forming the basis for the lambda calculus [10].

Remark 2. Some eigenforms are not related to an infinite process at all.

Consider FX = "the one who says X". Then X = FX is an X that can be placed in the sentence, so that X = "the one who says X". This is the equivalent to the sentence, "X is the one who says X". Any X that makes this sentence correct and meaningful is an eigenform for F. Since we can correctly say in English, "I am the one who says I", it follows that "I" is an eigenform for F. We did not find this "I" by a limiting process. Another example is seen in the sentence, "This sentence has thirty-three letters." The sentence is true and thirty-three is an eigenform for the transformation

FX = | This sentence has X letters. |

where |S| is the number of letters in a sentence, S.

Just so, the achievement of autopoiesis in the MUV model does not happen by an infinite limit. It happens by the circumstances fitting to themselves in a pattern of closure.

Returning to F(SS) = SS, we can regard this eigenform as descriptive of the MUV model. Then S is the molecular substrate with its rules and SS is that substrate applied to itself and running for some time, producing protocells and patterns. The statement F(SS) = SS can be interpreted as "SS is seen as F(SS), where F is the descriptor "field of protocells". In that sense, SS becomes an eigenform for the behavior that converts the molecular substrate into a field of self-generating protocells.

The idea behind eigenforms goes outside of mathematics, which has fixed or restricted domains of operation. The production of eigenforms is often a challenge to widen these domains. For example, if Fx = -1/x, then an eigenform, i, for F must satisfy i = -1/i, whence $i^2 = -1$. There are no real numbers whose square is -1. Thus, this eigenform does not exist as a real number. In the history of mathematics, this equation ($i^2 = -1$) and the possible meaning of i led to a new context in which the complex numbers, of the form a + bi, where a and b are real numbers, became the basis for a new area of mathematics. In that context, i can be interpreted as a rotation by ninety degrees in the two-dimensional plane. It was not obvious at the beginning of this history that i would be related to fundamental geometry. Advances in mathematics can be accompanied by the emergence of new eigenforms and their interpretations.

Consider any situation where there are objects of perception, or objects of thought. An object is invariant under certain acts of the observation. We find, as we move about, that the shapes of objects change, but we understand them, and even perceive them to be the same object. Physically, the object can be a process such as flowing water. We make the flow into an object. The form of the flow is unchanged under the transformation of the flow.

There is a linkage between myself of a moment ago and myself now.

In this way, I become an eigenform for my own self-observation.

We understand certain modes of speech, such as, "I am the one who says I," as expressing eigenforms for object/beings, as tokens of their eigenbehaviors.

A deep connection runs between the idea of eigenforms as objects of perception and the structure of autopoiesis. In both cases, the observer finds a structure that is unchanged under transformations (perturbations) of the observed system. The interaction of the system with itself can be understood as a form of self-observation. The observation of one system by another can be understood as how an observer finds autopoietic evidence for a living form.

Autopoiesis gives rise to eigenforms. Eigenforms can be seen as a description of autopoiesis and as a description of how the involvement of the observer works in bringing forth the autonomy of the autopoietic form.

The active loop of observation and description generates the eigenform, and there seems to be no need for an external process. Nevertheless, we see in autopoiesis that an external process can generate a condition that we are able to describe as an eigenform. The transformations that propel an animal or plant forward in time and preserve its structure are transformations that we observe and articulate. To the extent that we make these transformations our own, and make the distinctions our distinctions, then the autopoiesis and the eigenform will coincide.

2.2. Catalytic Eigenforms

We return to the catalyst in the MUV model. We construct a one-dimensional model. Suppose we have a (molecular) entity #, so that

 $\#\# \leftarrow \rightarrow \#$

with some probability, regarding this movement as a release $(\#\#\rightarrow\#)$ or a building up $(\#\rightarrow\#\#)$ of linkage. The catalytic entity * will have the likelihood of being linked to other molecules, #:

 $* \rightarrow * #$ and $* \rightarrow #^*$

with some probability.

The catalyst, *, tends to acquire a neighborhood of the form #*#. Let E denote that action of the environment on the catalyst. Then

E*~#*#.

Since ##~# with high probability, we have that ###~# with high probability.

Thus, A = ###...##*###...### has the property that EA is approximately A for one time-step of environmental interaction. The catalyst wants linkages while the molecules tend to unlink by themselves. The eigenform is a balance between linkage and release. We have produced a minimal symbolic example of catalytic autopoiesis. The state of autopoiesis is an eigenform.

3. Self-Replication

We can regard the MUV model as reproducing itself from moment to moment, in the sense that a protocell starts to decay and is then repaired by the action of the catalyst in forming new bonds.

We are familiar with the abstract schema for self-replication that is instantiated by a von Neumann "building machine", **B**. The machine, **B**, equipped with a blueprint, **x**, will produce the object, **X**, described by the blueprint. This can be presented as below:

$$B_{,x} \rightarrow B_{,x} \quad X_{,x}$$

Here, the arrow indicates production, and the result of the building machine's action is that we find the building machine, the blueprint, **x**, the newly built entity, **X**, and a copy of the blueprint, **x**. When the building machine is handed its own blueprint, **b**, then it will build a copy of itself.

 $B,b \rightarrow B,b.$

This is the most abstract and fundamental pattern for self-replication.

How is it related to the self-production of the protocells in the MUV model?

Where is the blueprint in the MUV model for protocells? We see that the protocells themselves are their own blueprints. These blueprints are not supplied beforehand. They arise from the structure and dynamics of the substrate of the MUV model. But once a protocell has emerged, it has the capacity to keep producing itself from the environment with aid of the catalysts that participate in its formation. At the level of these protocells, there is no distinction between the substrate and the coding. It is in the eye of an observer that the protocells acquire existence as self-producing entities. Life can emerge from molecular dynamics, and the first beginnings of code and blueprint can come into being.

Consider DNA (compare [11]). The DNA molecule consists of two interwound and bound-together helical strands, the Watson strand, W, and the Crick strand, C. We can write schematically DNA = WC to indicate the two strands and their binding. Replication occurs via enzymes that cut the bonds between base pairs in the two strands. Then, the environment in the cell provides new bases to pair with the available bases in the separated strands. The separated Watson strand is paired with a Crick strand by the environment and the separated Crick strand is paired with a Watson strand by the environment. The environment, E, produces C from W and it produces W from C, once they are separated.

$$WE \rightarrow WC$$
, $EC \rightarrow WC$

and so we have

$$DNA = WC \rightarrow WC \rightarrow WE EC \rightarrow WC = DNA.$$

The DNA is divided into individuated Watson and Crick strands that each are supplied with their complementary strand by interaction with the surrounding molecular environment.

The separation of the bound strands, **W** and **C**, to individual strands that can interact with the environment is accomplished via polymerase enzymes that act to break the bonds between the Watson and Crick strands.

Thus, the environment in DNA replication acts as a catalyst in that process in analog with the catalysis in the MUV model.

Note that, along with the complexity of molecular handling of the topological complexity, DNA strands are never completely "separated" in the replication fork. Thus, the actuality of DNA replication is simplified by our formalism.

See Figure 4 for a schematic illustration of DNA replication. Note that in the figure we have included the fact that two strands (W and C) are interwound with one another in the DNA. It is known that not only do the strands separate during replication, but also the winding between them is reduced by the action of the topoisomerase enzymes of type I and II. The type I enzyme can break a strand of DNA and allow the other strand to slip through the break and then reseal the break.



Figure 4. DNA replication.

Just as one can unhook a keyring by opening it and then closing it, the topoisomerase enzyme performs a similar function for the interwound strands. The type II enzyme can perform a similar job of unhooking one full DNA strand from another after they are replicated. In this way, enzymatic action handles topological information in the cell and in particular in the course of replication and mitosis.

With this schema in mind, we can ask: Where is the blueprint in the DNA replication? Where is the building machine in the DNA replication?

With the DNA, the Watson strand is the blueprint for the Crick strand and the Crick strand is the blueprint for the Watson strand.

This statement is the exact truth of the matter. Each strand with its bases ready to be paired with complementary bases in the environment produces its complementary strand, building it from the environment. When the enzymes pull the strands apart, the strands each become building machines for the complementary strands and blueprints (via that complementarity) for the resulting combination of strands that is the DNA.

In this way, we see how the structure of a blueprint has emerged from the underlying molecular interactions. Just how the DNA molecule evolved and became involved in life processes is a mystery, but with its presence we can see how a molecular substrate can become the support for self-replicating structures that are close to the abstract von Neumann scheme.

The **DNA** participates in more complex interactions that involve the construction of **RNA** molecules and other cellular architecture. We can begin to see how symbolic computational descriptions such as the von Neumann self-replication are in a continuous range of constructions that start with molecular substrates, have way stations like the **DNA** and continue on upward into information processing systems that are highly discrete, and relatives of the formal systems of mathematicians and computer scientists.

In the **MUV** model, we have a system, **S**, that acts on itself. We observe the self-action **SS**. We find that, when the protocells have emerged, then there is a form of invariance. **SS** consists in a field of protocells. If **F** denotes the functioning of the catalysts and linkages that lead to the formation of protocells, then we can write SS = F(SS) in the sense that **SS** is a field of protocells, and its form, while varying at the local level, is invariant in its collective pattern.

Assembling a Fixed Point

At this point, we reconstruct the results we have so far and pay close attention to process. Consider Gx = F(xx). We have applied G to itself and obtained a fixed point, as in GG = F(GG). Now let Rx denote xx. That is, R is the operation that duplicates its argument. We have

Gx = F(Rx)and GG = F(RG).

At this point, the process would stop unless duplication facilities were available.

There is no problem duplicating a typographical letter, but if we imagine realizing these functions as structures that are to be connected together under the indicated compositions, then to accomplish **RG** means to actually make a second copy of **G** and then put the two copies together.

We imagine that two Gs just happened by and decided to interact, but we used up one of the Gs in composing with the other, and now we are waiting for the duplicating machine. If the duplicating machine comes along, then we can replace **RG** by **GG** and then we would have an operation from **GG** to **F(GG)**, and with **GG** in place we can do it again and obtain **F(F(RG))**, and then we have to wait again for the duplicating machine before we can continue; and so it goes.

If you wanted your recursions to go on without such interruptions, you would have to devise a standard procedure for the duplication. This is what is achieved in **DNA** self-replication. The **W** and **C** strands can separate, and the environment will fill in the missing bases to form a new **C** for **W** and a new **W** for **C**. Environmental complementarity has come to the rescue and provided the duplication. There is, however, something we have not mentioned about **DNA** self-replication. It is a fact of nature that the **DNA** strings contain codes for all the basic molecules and enzymes (combinations of molecules) that participate in these cellular interactions. In principle, the **DNA** contains the coding for its own environment, and via the **RNA** transcription processes this coding can be used to realize these molecules and enzymes and its own replication.

Remark 3. For the purpose of this discussion, I just refer to the biological hypothesis that the **DNA** encodes all essential information for producing the other biological molecules in a cell. And this is of course with the caveat that the **DNA** is working in a cell to begin with. We do not attempt to solve the obvious but mysterious circularity in this proposal.

The whole cellular process is self-sustaining at a cybernetic level more intricate than the original autopoiesis from which we began. Here is the ultimate idea for self-replication that the entity would contain the coding for itself and its environment so that, in principle, it could lift itself by its own bootstraps.

There are amazing partial examples in practice. A virus comes equipped with a polyhedral outer shell that contains the viral DNA. Interaction of the virus with a living cell results in the viral DNA being injected into the cell. The viral DNA then uses the cellular environment and its own coding to produce many copies of the viral DNA and to manufacture many polyhedral shells for this new DNA. The viral process continues until it exhausts the cellular host, and a myriad of viral capsids have been produced. The virus produces its own environment via its own internal codes and the hospitality of the host cell.

Within the cellular **DNA** are codes for building the molecular environment that can decode that **DNA** and produce the very environment that is so encoded.

The key to the flexibility in coding and construction for **DNA** is the fact that subsequences in the string of base pairs can be singled out and specific proteins synthesized from these substrings via the action of enzymes on the **DNA** in combination with **RNA** transcription processes.

We can illustrate mathematical self-replications that are similar to **DNA** replication in form, but do not involve the string complementarity. Consider an element, **P**, in an associative and non-commutative algebra, such that **P** factors as a product of elements **W** and **C**, so that **P** = **WC**. Assume that **CW** is the identity in the algebra, so that **CW** = **1**.

If there is a process that pulls the factors apart, we will have a process of duplication for **P** that is similar to **DNA** replication:

$$P = WC \rightarrow W$$
 1 $C \rightarrow WCWC = WC WC = P$ P_{c}

Conversely, and algebraically, we have

PP = WCWC = W1C = WC = P.

Thus **P** is its own algebraic fixed point. In Figure 5, we illustrate this kind of replication in a topological context.

Topological Replication



Figure 5. Topological replication.

In that figure the reader will see an entity, P consisting in flexible strings attached to points at a top line and a bottom line. One string undergoes a simple topological transformation, making a maximum and a minimum, and then the resulting form is cut at the midpoint. This is analogous to the splitting of DNA. In the figure, we see that this splitting has produced two copies of P.

In Figure 6, we show the algebraic structure of this self-replication. P is seen to itself be a composition of entities A and B, so that P = AB, while BA is identical with the wiggle that we saw in Figure 5. Thus, we have that topologically $BA \sim I$, where I is the "identity", a simple arc. Then we have $P = AB = A I B \sim A BA B = AB AB = P P$.





So
$$P = AB = AIB = ABAB = PP$$
.

Figure 6. The algebra of topological replication.

The self-replication is produced by information that is 'stored inside P'. The form AB stores information that can be used in the form BA to accomplish the replication.

Notice that if we write the interaction in the opposite order, as $PP = ABAB \sim AIB = AB = P$, then P is its own fixed point under the composition with itself. The self-replication is the reverse of the fixed point structure. This topological example is not seen directly in biology, but it illustrates aspects that are mathematically implicit in the structure of DNA replication.

In this example, we have replaced the way the environment produces extra copies of **C** and **W**, via complementary base pairing, with an algebraic identity, 1 = **WC**. This is artificial with respect to the biology and it points out the power of nature's use of string complementarity. Nevertheless, this algebraic remark points out that we can view the **DNA** as its own fixed point under the non-biological reversal process of merging two **DNA** strands into one, which is implicit at the algebra level. So we see **DNA** as a fixed point or eigenform, and as a structural relative of autopoiesis, as well as being a participant in the autopoiesis of the cell.

Finally, in Figure 7, we show how recombination processes in DNA are closely related to topological changes in the DNA. The figure illustrates a twisted but unknotted loop of DNA. We have collapsed two strands into a single strand for the purpose of illustration. The loop of DNA undergoes a recombination process that consists in cutting the DNA and then rejoining it in a twisted crossover pattern, as illustrated in the first step in the figure. The DNA becomes two linked loops of DNA (K1) in this step. A second recombination produces K2, a figure eight knot. A third recombination produces K3, the so-called Whitehead link. The figure illustrates the further step, producing a knot, K4. In this way, processive recombination of DNA produces sequences of topologically knotted and linked molecules. The details of this process can be observed via gas chromatography and electron microscopy, and aspects of the theory of knots and links can be used to understand corresponding

aspects of molecular biology. Experiments and theory of this kind show how information is processed topologically in the workings of the cell. It will surely become more articulated in the future just what is the full relationship of topology, geometry and fixed point and Goedelian logic in the workings of molecular biology.



Figure 7. DNA recombination—topological information.

4. Reflexive Domains and Fixed Point Theorems

A reflexive domain (see [2,5,7]) is a domain, D, whose elements are each transformations of the domain. Algebraically, we define a reflexive domain as follows:

1. Given a and b in D, then there is a binary operation denoted by ab, so that ab is another element of D. D is closed under the binary operation. This makes each element, a, of D correspond to the mapping A: $D \rightarrow D$, defined by A(x) = ax.

The binary operation is *not* assumed to be either commutative (ab = ba) or associative ((ab)c = a(bc)).

2. Given an algebraic parenthesized expression E[x] with a free variable, x, involving elements of D, then there is an element, e, of D, so that E[a] = ea for any a in D. E[a] is the result of substituting a in the expression E[x]. In other words, algebraic operations *on* the reflexive domain are represented by the action of elements *of* the reflexive domain.

Given a reflexive domain, D, we have the

Fixed Point Theorem. *Let* F *be an element of a reflexive domain,* D*. Then there is an element,* p*, of* D*, so that* Fp = p*. Thus, every element of the reflexive domain,* D*, has a fixed point.*

Proof. Let G[x] = F(xx). Then, by property 2 above, there is an element, g, in D so that gx = F(xx) for all x in D. Applying g to itself, we have gg = F(gg). Letting p = gg, we have shown that p = Fp. This completes the proof. \Box

The key property of reflexive domains is that every element of D is also a mapping of D to itself, and all algebraic mappings of D are represented by elements of D. This is articulated as property 2 above. In this sense, we can write D = [D,D], where [D,D] denotes the algebraic mappings of D to itself that include the forms

 $x \rightarrow ax$

where a is some element of D.

The assumption of such a reflexive correspondence, D = [D,D], is in line with social and cybernetic intuitions about systems that involve the participation of the individuals who are the members of the system. The properties of the system arise from the dynamics of its members.

In language, one can introduce new words and new ways of speaking. One allows these new ways to interact with previously created language on an equal basis. The language as a whole grows in this way. In this sense, human languages are prototypical reflexive domains.

In computer languages and formalized languages, we can give rules for the production of new entities. Programs can be written that call previously created programs. A complex of programs can arise in this way, particularly if each program is given a name that is then regarded as a new element in the expanding programming language.

Reflexive domains are an abstraction of the above-described situations. The possibility to write an expression, E[x], in terms of a domain (created up to a given point) is analogous to writing a program in that language.

An experiment for the reader illustrates the complexity inherent in this idea of the reflexive domain. Start with a domain, D_0 , generated by one element, a. Then D_0 contains a, aa, (aa)a, a(aa), a((aa)a), (a(aa))a, ... where these comprise all possible parenthesized compositions of a with itself. This infinity of expressions is the simplest start. Our second requirement adds another infinity of elements by defining functions such as

$$G[x] = (xx)a$$
 for x in D

with a some given element of D_0 . Then the axiom asserts that there is g in D_1 (we introduce D_1 to allow these new elements), such that

$$gx = (xx)a$$
 for any x in D₁.

This means that

gg = (gg)a.

So letting Y = gg, we have Y = Ya.

There is an infinity of such constructions.

We can call D_1 the set generated by all the a's and all the algebraic expressions in one variable involving only the a's. From D_1 we can beget D_2 , and from D_2 beget D_3 . An infinite process of growth occurs. For a very remarkable formalization of this reflexive structure as a limit process, see the work of Dana Scott [6]. Scott constructs a domain, D, that is the limit of all the domains, D_n , so that D = [D,D] as desired.

Reflexive domains quite naturally go outside of initial domains where the constructions begin. For example, suppose we are in a logical domain with only the values True and False, so that negation has no fixed point. Then we can define E[x] = -xx and thus there is an R in the domain D, such that Rx = -xx for any x in the domain. Whence RR = -RR, and we have produced a fixed point, RR, for negation. RR is necessarily outside the initial logical domain. R is the direct analog of the Russell set in the context of logical reflexive domains.

This leads us to consider what it means to say that any a has a fixed point. There is an obvious answer. Let P = a(a(a(a(a...)))) ad infinitum. Then formally a(P) = P.

The infinite composition of a with itself usually leaps out of the given domain. But it is this possibility of infinite repetition that informs language at all times and is related to our fascination with mirrors and self-reflective structures and recursions of all kinds. In behaving politely and finitely in language, we nevertheless skirt at all times the possibilities of infinite repetition and infinite recursion. It is the presence of this edge in language that brings it into the realm of the reflexive domain.

One can think of the three dots as indicating infinity, as in (((((((...)))))))).

S = a(a(a(...))) is via notation a fixed point for a. There is no infinity on the page. The curious property of the fixed point theorem is that it gives us a method to construct the

5. Decorating the Fixed Point Theorem

The idea of the von Neumann building machine is generalized in formalisms by thinking of the machine, **B**, with blueprint, **x**, as a special machine, **B**_x, that can be applied to various inputs, **y**. We do not specify what **B**_x(**y**) will be, but we do specify that **B**_x(**x**) = **X** is the entity described by the blueprint **x**. Here we distinguish between the blueprint as lower case **x** and the actuality as upper case **X**.

 $\mathbf{B}_{\mathbf{x}}$ is a special machine that will produce the entity described by the blueprint \mathbf{x} .

With its special index, **b**, **B**_b(**b**) will build itself.

An indexed machine applies itself to its own index. This is a relative of the **DNA** using itself to produce itself. The self-relation continues up into the symbolic world.

This curiously profound simplicity of fixed point and self-replication has reflections in a number of domains. First of all, suppose that we articulate the mapping from the reflexive domain to its self-maps. Then we can write

$Z: D \rightarrow [D,D].$

Suppose we are given $F:D \rightarrow D$. Then define $C:D \rightarrow D$ via

$$\mathbf{C}\mathbf{x} = \mathbf{F}(\mathbf{Z}_{\mathbf{x}}\mathbf{x}).$$

Now we have articulated aspects of the process, with Z_x as the operator associated with x in the reflexive domain. This operator can be applied to x. This is the full articulation of "applying x to itself".

If $C = Z_a$, as in the second axiom for a reflexive domain, then

$$Z_a x = F(Z_x x)$$

for all x in D. Whence

 $Z_a a = F(Z_a a).$

If we think of Z_a as a function with code name a, then the fixed point arises via the application of Z_a to its own code.

This is the Goedelian fixed point [12] and it corresponds indeed to the Goedelian reference where the arrow is an arrow of reference from a code number to the statement that it encodes.

$$g \rightarrow B(uu)$$

 $gg \rightarrow B(gg)$

Here, $\sim B(gg)$ is a Goedelian sentence that denies its own provability by referring to its own code. The statement $\sim B(x)$ asserts, in the given formal system, that there is no proof of the statement whose code number is x. Code numbers act on themselves. If $g \rightarrow F(u)$ (with free variable u in F(u)), then $gh \rightarrow F(h)$, where gh means the code number of F(h), the result of substituting h for the free variable u in F(u). The formal system can speak about these compositions of code numbers. Thus, there is a code number for F(u).

We can have

 $g \rightarrow F(uu)$ $gg \rightarrow F(gg)$

and then **F(gg)** is a statement that refers to itself via its own code. Self-reference and fixed points are two sides of the same coin.

We can imagine that in a biological situation the action of Z_g on g represents the production of Z_g itself from its own code, g. Here one should regard Z_g as a building machine that will produce $Z_u u$ when given the code u. In order to produce itself, it needs to

be given it its own code. Then $Z_g g$ produces $Z_g g$. In this logical sense, all the productions of molecular entities from **DNA** codes are in the form of the production of Goedelian fixed points.

What about the negation? When we make $\sim B(gg)$, referring to its own code, we are in fact encoding a flag about the limitations of consistency in the system. The statement $\sim B(gg)$ carries its own markers pointing to its own unprovability within the system. We may regard $\sim B(gg)$ as located and coded, so that it points to what in the system must be eliminated by the system's "immune system". The idea of Goedelian sentences as fundamental to formal biological immune systems is due to Markose [13].

Thus, negative Goedelian sentences can become the indicators of attempts to change or make inconsistent the base code of the system. The system's stability depends on these Goedelian sentences.

We see that, above the level of initial and emergent recursions, the interactions of an autopoietic or living system are necessarily Goedelian and recursive. There is really no choice in this matter. To the extent that the system operates via codes and reference, this structure must emerge.

In the case of simple autopoiesis and cellular automata, we see the beginning of the emergence of Goedelian loops in the eigenform of the system itself.

In examining the relationships between our own understandings and those that come from mathematics and formal systems, it becomes apparent that we, as observers, are a composition of Goedelian entities of the form Z_gg , combining program, reference and self-action. It should not be surprising that our careful excision of such structures (when self—contradictory) from the mathematical formal systems leads to sharp differences between our apparent understanding and the image of understanding that can happen in a formal system. One must turn this situation on its head to make it right.

6. Algorithms and Exiting the Box

One can consider algorithms and their actions by thinking about a sequence of algorithms A_1, A_2, A_3, \ldots , so that each algorithm can be applied to any index, k, to compute some number, $A_n(k)$.

Now, make from this list a new algorithm by the equation

$$Z(n) = A_n(n) + 1.$$

Suppose that $\mathbf{Z} = \mathbf{A}_{\mathbf{k}}$ for some \mathbf{k} . Then we have

 $A_k(n) = A_n(n) + 1$ for any n.

We can let $\mathbf{n} = \mathbf{k}$ and then we obtain

$$A_n(n) = A_n(n) + 1.$$

This implies **0** = **1**, a definite contradiction.

This means that $Z(n) = A_n(n) + 1$ is not on the list of algorithms with which we began. Any list of algorithms is incomplete. Each list can be used to produce an algorithm that is not on the list.

The process of evaluating the algorithms at their own indices has led to something new. It is not just self-replication that is at stake here, but novelty. The act of evaluating on one's own code leads out of the box of already made structures.

We have returned to the autopoiesis and the MUV model in another way. It is a prerequisite for life that it move out beyond the molecular substrate to new structures and self-supporting dynamics. The key for this to happen is by the application of processes to themselves, as in the MUV model originally, and in the molecular substrates that produced the DNA. The application of algorithms to their own codes is the most abstract form of this possibility. It is seems unsatisfactory that any list of well-defined and halting (reaching a conclusion after a finite number of steps) algorithms should be incomplete. The problem is that one cannot make a mechanical procedure to decide whether a given algorithm will always halt. In both mathematical worlds and biological worlds, processes go on, and we do not always insist that they stop. In fact, we study them for the patterns that they produce as they run or live.

Partial Recursive Functions

In this subsection, we consider the meaning of partial recursive functions and give examples. We show how the domain of partial recursive functions can be regarded as a reflexive domain and so as a natural domain for the theory of biological processes. In that way, the Kleene fixed point theorem, proved below, is a centerpiece for theoretical biology.

The idea of ongoing recursions leads to the idea of listing not just halting algorithms, but all algorithmic procedures that can be written in the grammar of a given language. Then we can imagine a list of procedures

A_1, A_2, A_3, \ldots

that expresses all the algorithms in a given (formal) language. It is no longer the case that we know that $A_k(n)$ will halt when we set it to compute. It is an algorithm and it will compute, but it may not stop and give us a specific answer. One says that the A_i are partial recursive functions. They have the advantage that their list is complete and that they represent all the processes that can be defined in a given language. A complete list of partial recursive functions is possible in principle, because one can judge, without running the list, whether a program is grammatically correct. Therefore, list them all in order of size.

Remark 4. *Partial recursive functions (equivalently, partial recursive algorithms) include all grammatically correct algorithms that we know how to write down in a given language. Here is an example. Define the following procedure:*

Start with a natural number n from {1,2,3,4, ...}.

If n is even divide it by 2 to obtain n/2.

If n is odd multiply it by 3 and add 1 to obtain 3n + 1.

Repeat this process.

If in the course of the repetition the number 1 is obtained, stop and notify the operator that the process has halted.

Otherwise, continue the process.

We can call this process C(n) = Collatz(n) after its inventor. The question is: Does Collatz(n) halt for every natural number, n?

It is conjectured that this is so. To investigate, one programs C(n) on a computer and tries different values of n. So far, the process comes to a halt for every number, n, that has been tried. But so far (since the 1940s), there is no known mathematical proof that C(n)

must halt for every n.C(n) is an example of an algorithm that is in the list of all partial recursive algorithms.We do not know whether it halts, but it is a well-defined algorithm and must be listed.Without the problem of knowing whether algorithms halt for every n, one can in principle list all the partial recursive algorithms in a given language.

As we have indicated, it makes sense to consider $A_k(n)$ that do not necessarily halt (*partial recursive functions*). Here we consider partial recursive functions, so that, when the algorithm does halt, it produces a natural number that can be considered as the function value for the given input, **n**. Thus, a given A_k will produce definite natural number outputs, $A_k(n)$, for some input values, **n**. But, for some values, no output may happen.

Under such circumstances, one can assume a 1-1 correspondence

where **N** is the natural numbers and **[N,N]** is the set of partial recursive functions from **N** to **N**, defined relative to a given formal language. This is another way of saying that we can list all the partial recursive functions. This is because it is a mechanical matter to determine if an algorithm is partial recursive. It just needs to be grammatically well-defined. The problem of whether it halts for a given input, **n**, is not under question. It may halt or it may not halt.

Remark 5 (on notation). In the following, we shall write A_{B} to stand for A_{B} . This allows iterated subscripting to be typographically visible.

When we say that a function is *recursive* or that an algorithm is recursive, we mean that it does halt for every natural number, **n**, and hence gives us a definite answer for every input, **n**.

We now examine the fixed point theorem for reflexive domains in the context of partial recursive functions as a reflexive domain. As we shall see, this yields naturally a version of the Kleene fixed point theorem.

For a given natural number, **z**, we can write

$$A_{A_{z}(z)} = A_{g(z)}.$$

Here, $A_{\{z\}(z)}$ is partial recursive, but it has a definite index, g(z), since every partial recursive function is listed by its index.

Thus, we have $g: N \rightarrow N$ as a recursive function describing these indices.

If $f: N \to N$ is also recursive, then we can consider f(g(z)), and we have that

$$f(g(z)) = A_{v}(z)$$

for some special v in N and A_{v} is recursive. This means that

$$A_{f(g(z))} = A_{A_{v}(z)}$$

and hence

 $A_{f(g(v))} = A_{A_{v}(v)} = A_{g(v)}$

Letting $\mathbf{k} = \mathbf{g}(\mathbf{v})$, we have

We have proven the

Kleene Fixed Point Theorem. Given a recursive function, f, there is a natural number, k, such that

$A_{f(k)} = A_k$.

Here, $\{A_k\}$ k = 1,2,3,... denotes a listing of all the partial recursive functions in a given formal language (see [14]).

By considering partial recursive functions with definite indices, we can project the fixed points to live on the indices of the recursive functions. We have a fully reflexive domain to work with via the partial recursive functions. Markose [13] has pointed out that this gives powerful access of recursive function theory to biological and, indeed, autopoietic structures. The Kleene fixed point theorem is the right level for the recursions of computer science and also for biology, where the ongoing functions are open-ended and partial recursive.

Remark 6. The reader may wonder why we do not invoke the usual fixed point procedure for reflexive domains that we have already described earlier in the paper.

 $A_{f(k)} = A_{k}.$

Consider what would happen if we tried that. Let $f: N \rightarrow N$ be a given recursive function. Define $Cn = f(Z_n(n))$. Then Z_n is partial recursive and so C is partial recursive. Therefore, $C = Z_g$ for some natural number, g. Hence

$$Z_g(n) = f(Z_n(n))$$

and we can let n = g and conclude that

$$Z_{g}(g) = f(Z_{g}(g)).$$

However, $Z_g(g)$ may not halt, in which case we have not found a numerical fixed point for f. The strategy in the Kleene fixed point theorem that uses the well-definedness of the indices of partial recursive functions makes it possible to have an effective fixed point theorem for partial recursive functions.

Now the relationship with autopoiesis comes back into view. It is the chemistry of the operator Sx = F(xx) that can produce the autopoietic SS = F(SS). But S must be allowed to combine with itself. Just so, in the realm of the protocells produced by a catalyst, the cells act on themselves. The action of the molecular field S on itself can be seen by us, as observers, as happening at a different level from S acting on other entities. In the reflexive domain and indeed in the chemical domain, the actions just go on without halting. It is our singularity of vision that we concentrate on SS. In that domain there are the many catalysts and molecules. When a protocell occurs, it is possible for it to persist through the support of the environment.

This means that, in thinking about recursions in relation to biology, we should be sensitive to the fact that distinctions we make for the sake of logical analysis are not necessarily fundamental for the process itself, and yet they serve our understanding.

The system must interact with itself. **SS** is just the simplest symbolic expression of this action. Wittgenstein [15] says that a sign is that part of the symbol that is available to the senses. Here we see the relation of the observer and the world, meeting at the nexus of sign and symbol. Here we verge on ontopoiesis [16].

In mathematics proper, we see that it is often the case that assuming a reflexive correspondence asks many systems to go beyond their usual boundaries. We are very familiar with our own participatory stance. We make boundaries to avoid what would appear to be logical contradictions. The very fact that, in order to separate and reason about a mathematical object, we succeed in isolating it from being fully in a reflexive domain makes certain aspects of our understanding more mysterious than need be. It is hoped that the story of structures emerging from a molecular substrate can be used to reorient our vision for much of science, biology, language and mathematics.

7. Fixed Points, Fractals and Describing

The fixed point is a point of self-assembly of a form or organism.

It is useful to see examples of this and to see how the unfolding of a complex form can be encoded in the simplicity of a fixed point for a transformation.

We can view a fixed point as the summary of a recursion.

Thus, if we write

 $\mathbf{J} = \{\{\{\{\{\ldots, \ldots\}\}\}\}\},\$

then it is apparent that $J = \{J\}$. The fixed point property is a description of the way in which J is seen to reenter its own indicational space. This is a description from the point of view of an observer.

Thus, we may examine the Koch fractal, as in Figure 8, and see that Koch appears as four reduced copies of itself within itself. We may write an abstract schema such as

$\mathbf{K} = \{\mathbf{K}\{\mathbf{K}\mathbf{K}\}\mathbf{K}\}$

to indicate how two of the copies are above the baseline for the other two.



Figure 8. Koch fractal.

In the case of the Koch fractal, observation leading to the description of its reentry is quite different from the production at the coded or symbolic level of the fixed point. Yet these operations are dual. We can describe the Koch fractal by the recursion that generates it. The coded language that we use arises from the recursive dynamics of the systems that we are.

Another example of description in relation to recursion is the Mark [4] fractal of Figure 9. Here, the final fractal can be seen to consist of two reduced copies of itself at ninety degree angles to one another (i.e., one is rotated ninety degrees to the other). The two copies overlap, and so it takes a special observation to see this pattern. The sequences of recursions that yield that fractal all proceed by replacing segments by two segments at ninety degrees to one another, with endpoints at the same positions as the original segments.



Figure 9. (a) Mark fractal recursion pattern. (b) The Mark fractal.

The diagrammatic formula in Figure 9a is an abstract description of this fixed point. It requires the actual playing out of the recursion, as in Figure 9b, to see the complexity and beauty of the final form that is implicate in the order indicated by the fixed point.

In Figure 10, we illustrate three sequences, A, B, C, such that, with certain conventions, C describes B, B describes A and A describes C. The convention for description is like this: if we have a sequence, such as 1112331, then its description is "three ones, one two, two threes, one one." And then we transcribe this description into a sequence of digits by replacing number words by the corresponding digits, and removing the commas and punctuation. Thus the coded description is 31122311. It is in this sense that the three sequences, A, B, C, describe one another. At this point, the reader may like to examine Figure 10 carefully, and see that our claims are true. It is, of course, of interest to find out how to produce the sequences and to see that they can be extended as far as one likes.

$A = 11131221131211132221\dots$

B = 3113112221131112311332...

C = 132113213221133112132123...





Note that one can start with something very simple and make a sequence of descriptions. For example, if we start with 1, then we find

This is sometimes called the "look and say" sequence. It was studied by John Horton Conway [17]. Conway tells the story that he was given the sequence as a puzzle (find the next row) at a party and that he did not solve it! Upon being told the solution, he was so fascinated that he worked out a great deal of structure about this recursion, resulting in the research paper we have mentioned. It is remarkable how much complexity can arise in the process of successive description.

You can notice that if we take 3 as the initial element to be described, then Row_{n+3} is an exact extension of Row_{n}, where Row_{n} denotes the n-th row in the sequence. Here is this look and say sequence below, starting with 3.

The sequence uses only the digits 1, 2 and 3. We can construct A, B, C by extending the first three rows by using later rows. Thus we start with

	3
	13
	1113
and extend to	
	3113
	132113
	1113122113
and then extend this to	
	311311222113
	13211321322113
	1113122113121113222113
and continue in this fashion.	

ar

With that idea in mind, we see the following pattern:

3. 13. 1113.

Then go back to the top and extend it so that it defines the third row.

Extend the second row to describe the first row and extend the third row to describe the second row. I have put periods in the sequences to show the new additions to the rows.

3.113. 13.2113. 1113.122113.

Now go back to the top and continue.

3.113.11222113. 13.2113.21322113. 1113.122113.121113222113.

As you can see, the very first period placement, 3.113, is to be read "three ones, one three." After that, we write descriptions of each packet as it comes down. Thus, the description of 3.113. is 13.2113.

We can now continue this process as far as we like.

We can symbolize it as

3 = X13 = DX1113 = D²X

where DX denotes "the description of X".

If we let 3.113 = 3 "+" 113 = X + Y, then we see that we have

$$D^3X = 3113 = 3.113 = X + Y.$$

Our process takes the form

$$A = X + Y + D^{3}Y + D^{6}Y + D^{9}Y + \dots$$
$$B = DX + DY + D^{4}Y + D^{7}Y + D^{10}Y + \dots$$
$$C = D^{2}X + D^{2}Y + D^{5}Y + D^{8}Y + D^{11}Y + \dots$$

And we see that DA = B, DB = C and DC = A. This works because $D^3X = X + Y$. We see that it is because $D^3 = 3113$ that the entire A, B, C structure arises. With that beginning, the sequences A, B, C assemble themselves via the recursion and produce the three-fold fixed point (A, B, C) for the transformation

$$T(R, S, U) = (DU, DR, DS)$$
 so that $T(A, B, C) = (DC, DA, DB) = (A, B, C)$.

The way in which this fixed point assembles itself depends on the particular properties of the forms of description. It is striking that the exact forms of the infinite sequences arise from a simple recursive process.

The simplest question in this context is the problem:

Find a string S so that DS = S.

"Two twos" describes itself. Thus, D(22) = 22.

The look and say sequences have extraordinary complexities, and in the quiet center of all of that sits the sequence 22, a fixed point of the process of description.

8. Recursive Distinctions

Here we describe another version of distinguishing and describing where the alphabets that are generated only indicate states of difference or sameness in rows of characters [18].

I will use a minimal alphabet, where * indicates that the characters on either side of a given character are different. Thus, if we have ABC, then B is described by *.

I use] to indicate that the left character is equal but the right character is not equal. Thus, in BBC, the middle B is described by].

I use [to indicate that the right character is equal to the middle, but the left character is not equal. Thus, in ABB, the middle B is described by [.

Finally, if both characters are equal to the one in the middle, I use an empty character to describe the middle. Thus, in BBB, the middle character is described by an empty character.

Now, if you start with a row of characters, you can replace it by their descriptive characters. This is again a row of characters, and so the game can continue recursively.

For example, suppose we start with the sequenceBBBBBBBBBBBBBB...

Then we will find the replacement, as shown below.

. . .

...BBBBBBBABBBBBB...]*[

. . .

Note the appearance of the blank characters. Doing this twice more, we see the following.

]*[
]***[
]*[]*[

At the first description, a protocell in the form]*[appears where the boundaries of the cell denote the places of partial difference in the initial character string. The next description opens up the protocell with more indication of difference in its interior. And the next description sees two partial difference characters appear from the environment, and a blank character appears, yielding two protocells. The original protocell has undergone mitosis! Thus, this recursive distinction process has produced a protocell and then witnessed its reproduction. There is clearly a dialogue in this process between sameness and difference, with the appearance of the partial difference characters marking the boundaries in this process. Here we have an example of elementary dialectics arising from a process of recursive distinction. The analogy of this process with our abstract description of DNA reproduction should be clear. In that process, we had the DNA as a composite of the Watson and Crick strands (analogs of the left and right characters). The Watson and Crick strands are separated in the environment, and then the environment conspires to replace new Crick and Watson strands to form the two new DNA structures. Our diagrammatic for this has the form

<w c="" =""></w>				
<w td="" <=""><td colspan="2">Environment</td><td> C></td></w>	Environment		C>	
	<w c></w c>	<w c></w c>		
	DNA	DNA		

DNA

The formal analogy with the recursive distinguishing process is remarkable. Figure 11 shows how the recursive distinguishing process continues.

Figure 11. RD pattern.

After the first protocell division, there is a separation with a wider environment. Then another production of two cells. Then, in two steps, four cells are produced. These join and then there will be eight cells, and so on. What are we to make of this recursive distinguishing structure?

We see self-assembly and reproduction starting from only the situation of characters that indicate distinctions between adjacent characters. The notion of adjacency is more general than the representation of characters in a linear array, but the use of the linear array allows an elementary demonstration of the principle. In principle, one can imagine a network of entities with a concept of adjacency among them (a graphical network, for example), and recursively these entities are replaced by "letters" that describe the distinction or lack of distinction in their adjacency relations. The recursion goes from one network of entities to the next, with each successive network a description of the previous network.

9. Cellular Automata

In this section, we discuss cellular automata in the light of autopoiesis. In a cellular automaton there is no fixed number of particles. The background changes synchronously (all at once) and has rules for creation and destruction of particles on the lattice of a board, B. With this choice of background, it is the rules that make all the difference. In this sense, the rules of the automaton are the environment for its productions. If X is the state of the cellular automaton board at the beginning, and R stands for the rules, we can write RX for the new board after the rules have been applied. Then we have a sequence of boards: B, RX, RRX, RRRX, and so on. It is the pattern and evolution of these boards that is our concern. From a mathematical point of view, we are studying the recursion

$$X_n = R^n (X).$$

In this section, we illustrate aspects of autopoiesis, replication and self-assembly associated with cellular automata.

The rule of HighLife, a variant of Conway Life, is **B36/S23.** This means that we work on a rectangular grid, and a square on that grid is said to be 'alive' if it is colored white and 'dead' if it is colored black. The rule says that a square can give birth if it has either 3 neighbors or 6 neighbors (the total number of neighbors of a given square is 8). In order to survive, a white square has to have either 2 or 3 neighbors. These rules are followed sequentially, where the contents of the entire board are changed according to the rules. This means that one checks each square for birth or death and then simultaneously converts all the squares at once.

We begin this section with HighLife because it has a "replicator" that has the same pattern as the sequence of productions of protocells in the recursive distinctions of the previous section. See Figure 12 for an illustration. The details of the pattern in HighLife are more complex than in the recursive distinguisher. It would be an interesting and significant project to analyze the relationships between these two recursions.



Figure 12. HighLife replicator.

In the rest of this section, we work with a variant of Conway Life that I call Life7. The rule for Life7 is **B37/S23**. This differs from Conway Life, where the rule is **B3/S23**.

In Conway Life, a prolixity of patterns are produced, but most configurations settle down to oscillatory states. In Life7, the productions are more volatile and configurations often grow without bounds.

Note that in the MUV model we used an asynchronous environment. In the cellular automaton, 'particles' (the white squares) are created and destroyed according to the rules. In the MUV model, particles move about and can link up with one another.

Within each board, the observer finds patterns and organizes them into descriptions of interactions related to the recursive evolution of these forms. While the substrate is different from the MUV model, the observational process is very similar.

In Figure 13, we see a configuration of 5 points (I will refer to white squares as points) that is called a *glider*. When the recursion is applied twice, the glider is transformed into a mirror image version of itself, and in two applications of the recursion the glider is transformed into a copy of itself that is shifted diagonally down the board by one vertical step and one horizontal step. Thus, the glider reproduces itself and "moves" down the board.



Figure 13. A glider in Life7.

In Figure 14, I illustrate an entity that I call the *rotor*. The rotor after 10 steps turns by 90 degrees (going through different intermediate forms). After 40 steps, the rotor returns to its original position. The rotor is a very rare configuration. The glider is highly likely to appear from any random starting configuration.



Figure 14. The rotor in Life7.

In Figure 15, we show at the top a mirrored pair of entities. After ten steps (iterations of the recursion), the mirrored pair reappears in mirrored form, with created material to its left. After 20 steps, the mirrored pair reappears in its original form, trailed by even more new material. This generation process continues unheeded, apparently forever on a sufficiently large board. The mirror pair is robust in terms of its own production and also occurs (with low probability) on the boards of its own accord.

In Figure 16, we illustrate a configuration that started from a smaller and randomly chosen beginning. It has now evolved to the point where this observer judges that it will continue to live and grow. Note the glider that has emanated in the upper right.



Figure 15. Mirrored generator pair in Life7.



Figure 16. A living configuration in Life7.

One can carry out the experiment and see that, indeed, this configuration continues to live and grow. At this point, it would be appropriate to call it autopoietic, as its "life" is not affected by small perturbations (adding or removing some points).

Note that the configurations in Figure 16 appear to have a number of clusters that can be regarded as protocells. These clusters change shape and are seen to interact with one another as the recursion goes forward. They coalesce, emit gliders and generally engage in complex dynamics. One can experiment with the dynamics by changing the rules. For example, Conway Life has rule **B3/S23**. If we remove the rule of (born on) 7 in Life7, we obtain Conway Life. Then a configuration such as the above will lose its dynamism and become a collection of separate small entities that are either unchanging or periodic.

Life7 affords the possibility of (generalized) autopoiesis by the dynamics that are added through the rule that gives birth on 7 neighbors. In this dynamic, one observes local structures that interact with one another and a global structure that is relatively impervious to perturbation. The reasons why the larger structure is maintaining its form are no longer obvious just from the base rules. More work is needed to understand how autopoiesis has been achieved.

More generally, we ask the question: How is autopoiesis achieved in cellular automatic systems? Such systems are characterized in our modeling by a recursive rule, R, that takes boards B to boards RB. The rules for evolution create many local changes in the board but the rules can be described very simply, just as we have described Life7 by the rule B37/S23.

Autopoiesis is a condition that can be assessed by an observer of the boards. The observer notes that a pattern on the board is persisting across intervals of time and judges that this pattern is an intrinsic property of the dynamics of the boards. Note that, without observers, these recursive systems would not be interpreted via the concepts of internal dynamics and autopoiesis.

The observer of the cellular automaton is involved in making discriminations at multiple levels. On the one hand, the observer is aware of the global recursion and is aware of its definition. On the other hand, the observer sees a nearly fractal array of substructures and sub-recursions in action as the boards are surveyed. This double level is analogous to that of a molecular biologist who is aware on the one hand of the complex chemistry that drives the dynamics of organisms, and is also aware on the other hand of the self-sustaining nature of the complex of interactions that constitute the existence of living things.

Eigenforms appear via the interaction of the observer and the board.

Thus, we have the situation of a recursion going on, $R^nX = X_n$, and observers noting how entities and patterns occur on the board. Observers can intervene by finding new initial states and combinations.

Eigenforms in this discussion are not just fixed points. They are fixed points for some levels of observation. In this way, objects become identified as tokens for eigenbehaviors (von Foerster 2002).

In observing recursive systems, one is continually confronted with the possibility that the algorithms will or will not halt. The observer can contemplate questions of computational decidability and computational undecidability in the context of recursive autopoiesis. In observing the dynamics of a cellular automaton, like Life7, one sees the evolution of the large-scale structure in terms of the interactions of its parts. These parts, like the protocells in the MUV model, have independent action and they can interact with one another. One can investigate those parts just as we investigated the glider, the rotor and the large board configurations. One finds that the parts are continually constructed and reconstructed from the substrate of the board and the recursive rules. By looking at autopoiesis in this way, we open the door to the vastness of the whole structure of recursion as a generator of process, dynamics and the forms of the living.

10. Summary

We began with the MUV model for autopoiesis, pointing out that the state of autopoiesis can be regarded as an eigenform that is a stability for an observer of the system and a stability for the system itself when this can be defined. We raise the question of understanding the relationship in autopoiesis of the observer and the observed. We extend this discussion by using the concept of eigenforms, and we suggest that autopoiesis can be generalized to include stabilities for an observer of a general recursive system, including the observation of cellular automata. The paper includes an exposition of the properties of Life7, a relative of the Game of Life of John Horton Conway. Life7 has the property that many starting configurations yield dynamically stable configurations in the eye of the beholder. We see that autopoiesis, in this generalized sense, can be observed in the successive boards of a recursion, in an unlimited sequence of recursive forms.

In the end, it is not surprising that recursions and fixed points are seen to be fundamental to biological observation. DNA reproduces by the structure of its own self-reference. The MUV model of autopoiesis must act on itself and achieve a fixed point, SS = F(SS).

We have the general fixed point obtained from gx = F(xx), giving gg = F(gg). Variants of this fixed point construction include the Kleene fixed point theorem and the Goedelian sentence: $g \rightarrow \sim B(uu)$ yielding $gg \rightarrow \sim B(gg)$, whereupon the system has a sentence that states its own unprovability. That sentence, $\sim B(gg)$, constructs its own meta-assertion. The building machine inherits its pattern from the fixed point construction as well: $B,x \rightarrow B,x$ X,x whence $B,b \rightarrow B,b$ B,b.

Anything that 'returns to itself' is a fixed point, just as anything that returns the negation of itself is a 'paradox'. Recall also now that DNA does not just reproduce itself, it also produces RNA by controlled enzyme action, and the RNA participates in production of protein in the cell. So the DNA as a fixed point, as self-replicating, is also productive and uses its code for much more than self-replication.

It is in studying the actual contexts of recursions such as cellular automata that we see such intricacies happen as a consequence of geometry of computational space interacting with rule structures. Thus, gliders appear in the Life automata, and these gliders both move and reproduce themselves. The emergence of forms under recursion has been a theme of this paper and it is part and parcel of the structure of all systems that involve process, biological, linguistic and mathematical.

We end with references to other literature on autopoiesis that also starts in simplicity and extends the ideas in many directions. The paper by Rao Mikkilineni [19] faces the questions: How did intelligence evolve? What is the relationship between mind and body? Therefore, it asks the questions about how simple autopoiesis can evolve to become complex living structures. We ask these same questions. It is worth continuing the exploration in the direction of understanding complex structures. The paper by Stuart Kauffman [20] suggests autocatalysis as the basis of the evolutionary structure of molecular biology. Here, there is the direct suggestion that a form of autopoiesis is fundamental to biological evolution. Furthermore, the full work of Stuart Kauffman is highly relevant to the present discussion, as he is concerned with how mechanisms arise in the development of biology and how these mechanisms often participate in many inter-connected roles. Such structuring goes beyond the 18th and 19th century ideals of predictive mathematics, but is not beyond the more architectural work of studying formal systems, recursions and cellular automata that we have described here. Thus, there is potential for the present work to contribute to these ideas about the evolution of multiple roles.

Mathematics, topology recursion and graph theory can come together in interdisciplinary studies such as [21], This sort of work relating molecular biology, non-locality, music and poetry finds its fulcrum in the formal mathematics that we have used to examine autopoiesis. We can expect new ideas about the fundamental questions of evolution of structure to emerge from this kind of approach.

Finally, we mention the concept of ontopoiesis [16], a philosophical concept that involves the communicative engagement of self with the world and the world with the self. Ontopoetics is a view of reality and understanding of the world as a communicative whole. Just as in our concerns with distinction and recursion, we have seen that systems arise from wholes by dividing into a part that sees and a part that is seen. How autonomy and autopoiesis, or indeed poiesis, are described depend crucially on this opening of a world observed and a world that acts to observe. Autopoiesis is enfolded in ontopoiesis in the creation of the universe from itself. Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

- 1. Maturana, U.; Uribe, R.; Varela, F. Autopoiesis: The Organization of Living Systems, Its Characterization and a Model. *Biosystems* **1974**, *5*, 187–196.
- Kauffman, L.H. Eigenform. In *Kybernetes*; Emerald Group Publishing Limited: Bingley, UK, 2005; Volume 34, pp. 129–150, No. 1/2.
- 3. von Foerster, H. Objects—Tokens for (eigen) behaviours. In *Understanding Understanding*; Springer Publishing Company: New York, NY, USA; Berlin/Heidelberg, Germany, 2002; pp. 261–272.
- 4. Spencer-Brown, G. Laws of Form; George Allen and Unwin Ltd.: London, UK, 1969.
- Kauffman, L.H. Laws of Form, a survey of ideas. In *Laws of Form—A Fiftieth Anniversary*; Kauffman, L.H., Cummins, F., Dible, R., Conrad, L., Ellsbury, G., Crompton, A., Grote, F., Eds.; Series on Knots and Everything; World Scientific Pub Co.: Singapore, 2022; Volume 72, pp. 1–88.
- 6. Scott, D. Continous Lattices; Springer Lecture Notes in Mathematics; Springer: New York, NY, USA, 1972; Volume 274.
- Varela, F.J. Principles of Biological Autonomy; The North Holland Series in General Systems Research; Klir, G., Ed.; Elsevier North Holland Pub.: New York, NY, USA, 1979.
- 8. Varela, F.J.; Goguen, J.A. The arithmetic of closure. J. Cybern. 1978, 8, 291–324. [CrossRef]
- 9. Kauffman, L.H.; Varela, F.J. Form dynamics. J. Soc. Bio. Strs. 1980, 3, 171–206. [CrossRef]
- 10. Barendregt, H. The Lambda Calculus—Its Syntax and Semantics; North Holland Pub.: Amsterdam, The Netherlands, 1984; 642p.
- Kauffman, L.H. Biologic. In AMS Contemporary Mathematics Series; Northeastern University: Boston, MA, USA, 2002; Volume 304, pp. 313–340.
- 12. Nagel, E.; Newman, J.R. Goedel's Proof; New York University Press: New York, NY, USA; London, UK, 2001.
- Markose, S.M. Genomic Intelligence as Über Bio-Cybersecurity: The Gödel Sentence in Immuno-Cognitive Systems. *Entropy* 2021, 23, 405. [CrossRef] [PubMed]
- 14. Machtey, M.; Young, P. An Introduction to the General Theory of Algorithms; North Holland: New York, NY, USA, 1978.
- 15. Wittgenstein, L. Tractatus Logico-Phiolsophicus; Routledge and Kegan Paul Ltd.: London, UK; New York, NY, USA, 1922.
- 16. Tymieniecka, A.-T. *Imaginatio Creatrix: The Pivotal Force of the Genesis/Ontopoiesis of Human Life and Reality;* Springer Science & Business Media: Dordrecht, The Netherlands, 2004.
- 17. Conway, J.H. The weird and wonderful chemistry of audioactive decay. In *Open Problems in Communication and Computation;* Cover, T.M., Gopinath, B., Eds.; Springer: New York, NY, USA, 1987; pp. 173–188.
- 18. Isaacson, J.D. Autonomic String-Manipulation System. U.S. Patent No. 4,286,330, 25 August 1981. Available online: https://www.isss.org/2001meet/2001paper/4286330.pdf (accessed on 1 November 2023).
- 19. Mikkilineni, R. A new class of autopoetic and cognitive machines. Information 2022, 13, 24. [CrossRef]
- 20. Kauffman, S. Autocatalytic sets of proteins. J. Theor. Biol. 1986, 119, 1-24. [CrossRef]
- 21. Planat, M.; Aschheim, R.; Amaral, M.M.; Fang, F.; Irwin, K. Graph coverings for investigating non local structures in proteins music and poems. *Sci* 2021, *3*, 39. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.