Complex Networks from Simple Rewrite Systems

Richard Southwell*, Jianwei Huang*, Chris Cannings@

*Information Engineering Department, The Chinese University of Hong Kong,

Shatin, New Territories, Hong Kong,

richardsouthwell254@gmail.com

jianweihuang@gmail.com

@School of Mathematics and Statistics, University of Sheffield,

Sheffield, S3 7RH, UK,

c.cannings@sheffield.ac.uk

May 1, 2014

Abstract

Complex networks are all around us, and they can be generated by simple mechanisms. Understanding what kinds of networks can be produced by following simple rules is therefore of great importance. We investigate this issue by studying the dynamics of extremely simple systems where are 'writer' moves around a network, and modifies it in a way that depends upon the writer's surroundings. Each vertex in the network has three edges incident upon it, which are colored red, blue and green. This edge coloring is done to provide a way for the writer to orient its movement. We explore the dynamics of a space of 3888 of these 'colored trinet automata' systems. We find a large variety of behaviour, ranging from the very simple to the very complex. We also discover simple rules that generate forms which are remarkably similar to a wide range of natural objects. We study our systems using simulations (with appropriate visualization techniques) and analyze selected rules mathematically. We arrive at an empirical classification scheme which reveals a lot about the kinds of dynamics and networks that can be generated by these systems.

1

1   Introduction

Many advances in complex systems research have come from identifying simple deterministic systems which can generate complex behavior. For example, cellular automata [1] are used to model many systems in physics and biology, string rewrite [2] systems are used to model plant development, and chaotic differential equations [3] are used in many areas. The most popular models of complex network growth are probabilistic [4, 5]. These models have yielded many insights, but they do not offer an explanation as to *where* the complexity in networks comes from. The complexity in probabilistic models comes from the randomizing mechanisms they are based on.

In this paper we study a class of simple deterministic network growth models. Our systems can produce a wide variety of behaviour, ranging from the very simple to the very complex. They offer insights into how complex structures can be generated, because each aspect of their behaviour can be traced back to a deterministic cause. The simple nature of our models allows us to quickly generate exact pictures of their dynamics. This gives us the ability to easily explore the behaviour of large numbers of systems. By taking an unbiased look at the dynamics of many simple rules we see what types of network growth are easily generated by simple computational processes.

In Turing machines [7] a writer moves along a one dimensional tape, and rewrites symbols using local rules. Our *network automata* systems are like a generalization of this idea -where the writer moves around a network, and rewrites it on a local level. The way the writer moves and modifies its local structure is determined by its surroundings. The network the writer runs on is a *trinet* (i.e., each vertex has three connections). Trinets are also known as cubic graphs and three-regular graphs. There are many natural examples of trinets, such as two dimensional foams and polygonal networks formed by biological cells and cracks in the earth. We focus on trinets because they are easy to manipulate and expressive. Any network can be represented as a trinet by replacing vertices that have more than three connections with "roundabout-like" circles of vertices with three connections [19].

One of our goals is to find simple deterministic rules that generate complex dynamics. An obstacle to this goal is the inability of deterministic rules to break symmetries. For example, if the writer is at one of the corners of a cube then it has no deterministic way to select which of its "similar looking" neighbors it should move to next. We circumvent this obstacle by supposing the edges of

2

our trinet are colored red, blue and green in such a way that touching edges have different colors (such an edge coloring is also known as a Tait coloring [6]). This allows us to specify how the writer should move by stating which colored edge it should move along in different situations[1].

Our main goal is to understand about what kinds of behaviour can be generated by colored trinet automata. To achieve this, we do a thorough exploration of the dynamics of a space of simple rules. Using pictures and analytic techniques we sort these rules into three classes (fixed points, repetitive growth and elaborate growth) according to their behaviour. We discuss these classes in sections 2,3 and 4. In section 5 we exhibit more general rules, which can produce other exotic behaviour such as persistent complex behaviour, periodically changing networks and production of hyperbolic space. In section 5 we also present many rules that can produce networks that look remarkably similar to real world objects.

We have tried to keep the main text as free from equations as possible. This was done to increase readership, and because complex phenomena are often better explained using pictures. We also studied our systems mathematically, and the appendix contains proofs which will give the mathematically inclined reader significant insights into some of our system's dynamics.

1.1 Related Literature

Many interesting deterministic network growth models have been previously considered. These include the growth models discussed in [8] and the biologically inspired models considered in [9–12], which can produce rich and complicated self replicating structures, despite extremely simple rules. Self replication in adaptive network models was also considered in [13]. In each of these cases, every part of the network gets updated in parallel. In our systems, by contrast, only a small piece of the network gets updated at any time. This makes our systems easier to implement because there is no need for distant parts of the network to have synchronized clocks. It is true that many real world networks grow in parallel, however evolving our systems for many time steps often leads to structural modifications that are equivalent to global/parrelel rewrite operations (see subsections 4.1 and 4.4).

---

[1]We could remove the reliance of edge color in our systems, by replacing the red, blue and green edges with different uncolored structures, and modifying our rewrite rules accordingly. Although this would make the systems look more complicated.

3

Our work was inspired by the work of Tommaso Bolognesi [19], who studies the dynamics of planar trinet automata. Like our models, these systems involve a writer which moves around trinet, making structural modifications as it goes. Unlike our systems, the models considered in [19] circumvents the symmetry breaking problem by supposing the trinets are embedded in the plane. The way the writer behaves in these systems is governed by the structure of the faces formed by the planar embedding. Although these models are fascinating, the fact that the dynamics depend on how the network is embedded makes it difficult to get a full picture of what is going on at any given time. Our approach (of considered edge-colored trinets, rather than planar embedded trinets) allows us to easily picture the complete dynamics (and rules) behind the systems we consider.

The experimental approach we use was pioneered by Stephen Wolfram [14], who uses simulations to reveal a vast array of simple programs that can generate complex dynamics. Wolfram finds many network growth models that can generate complex dynamics and explores the idea that the physical universe could be generated by a simple network growth model. The idea that the universe can be generated in exact detail, by a simple program, is one of the most interesting conjectures from digital physics [15, 16, 18]. Wolfram suggests that the correct model could be found by doing an automated search of the simplest possibilities [17]. In order to do such *universe hunting* one needs intuition about the kinds of things that simple adaptive network mechanisms can do. One of our aims is to provide some of this intuition.

1.2   How Our Systems Work

A colored trinet automata is a dynamical system where a writer moves around the vertices of an edge-colored trinet; applying rewrite rules as it goes. Every time step some rewrite operation is applied about the writer's current location and then the writer moves. We start by considering only extremely simple rewrite operations, where the writer's current vertex is either replaced with a triangle or left unaltered. The action the writer takes on a given time step is determined by the colors of the edges interlinking its neighbours.

The rules behind a colored trinet automata specify how the writer should move, and modify the network in response to its surroundings (i.e., the colors of the edges interlinking the writer's neighbors). We picture the rules at the top of our figures, and the writer is represented by a black vertex. For example, the
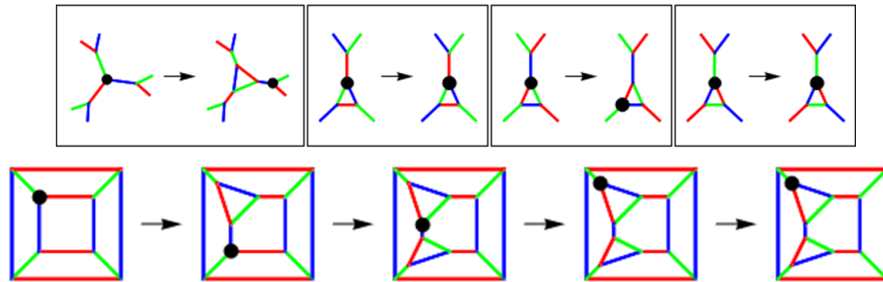
Figure 1: An example colored trinet automata. At the top we show the rules of the system by indicating how the writer (the black vertex) moves and modifies the network in response to its surroundings. Underneath we show the system evolving for four updates, starting from the cube. The system reaches a fixed point after three updates.

rule for the system shown in Figure 1 can be described as follows;

1. If there are no edges linking the writer's neighbours then the writer then moves along a blue edge, and the writer's previous location is replaced with a triangle.

2. If there is exactly one edge linking the writer's neighbours, which is red, then take no action.

3. If there is exactly one edge linking the writer's neighbours, which is blue, then the writer along a red edge and the network is left unaltered.

4. If there is exactly one edge linking the writer's neighbours, which is green, then take no action.

Notice how these instructions correspond to the pictures in the four boxes at the top of Figure 1. In general, a rule is just a specification of which structural modifications and movements the writer should perform in response to the four types of surroundings (depending on colors of edges interlinking the writer's neighbors) that the writer can have when there is no more than one edge linking its neighbors. Our rules do not specify which actions the writer should take in other situations, where there are two or more edges linking its neighbors. In these cases we suppose that no action is taken. Actually, the rules we consider never generate vertices with two or more linked neighbours and so these situations never occur anyway.
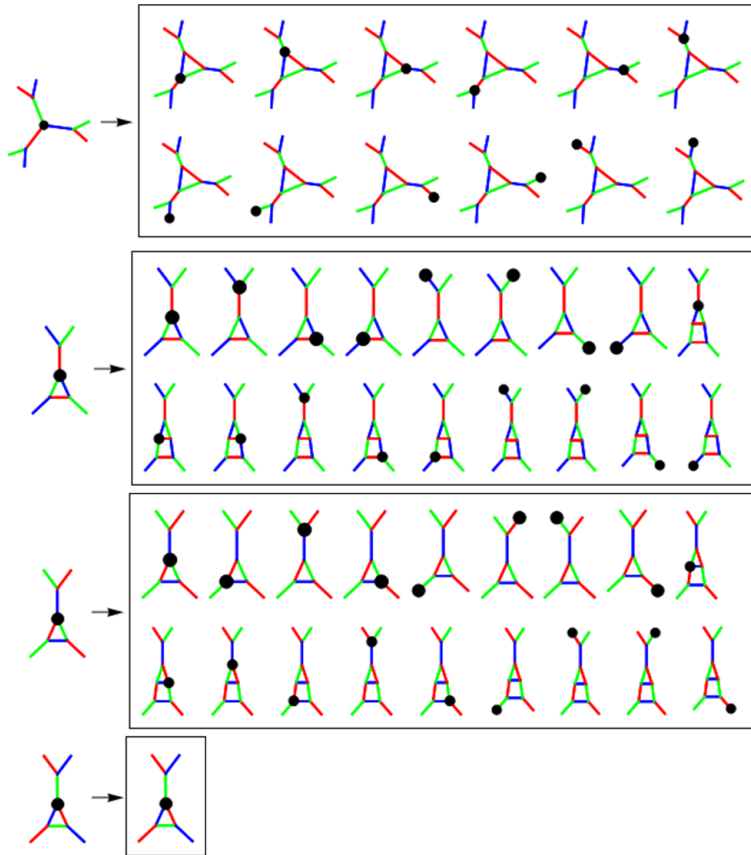
Figure 2: An illustration of the space of rules we explored. A rule is specified by associating each of the four different types of surroundings with one of the images to its right.

1.3 The Rule Space

In each of our investigations we used the cube (shown at the bottom left of Figure 1) as our initial condition. We studied the dynamics of the space of $12 \times 18 \times 18 = 3888$ rules depicted in Figure 2.

These are all the rules which involve possible triangle-replacement, and movement along two or less links, with two exceptions. Our first exception is that we always replace a vertex with no interlinked neighbours with a triangle. We make this exception so that each of our rules is guaranteed to change our initial *cube* network. Our second exception is that we never take action

when there is a green edge linking the writer's neighbors. We make this exception to reduce the number of rules we must consider. If we drop this second constraint we find new kinds of behaviour, but the rule space becomes too large to thoroughly explore.

By severely limiting the kinds of operations our trinet automata can perform, we reduce the rule space to a manageable size and insure that it contains minimal systems that generate certain types of behaviour. We sort our 3888 rules into three classes (fixed points, repetitive growth and elaborate growth) according to the long term dynamics they generate, starting from the cube. The following three sections are devoted to describing the behaviour of the rules in these three classes.

2 Fixed Points

We say a system has reached a fixed point, where there comes a time after which the network no longer changes. 2918 (about 75 percent) of the our rules eventually reach a fixed point. Our example system from Figure 1 reaches a fixed point after three updates. One reason such a large number of rules end up at a fixed point is that our rule space is such that whenever a green edge links a pair of the writer's neighbors a system becomes fixed. Indeed 2562 of our rules halt their evolution because of this effect. These include the rule which grows the largest fixed structure, shown in Figure 3.

In addition to the fixed points where the writer halts, there are *dynamic-writer* fixed point where the writer continues to move forever, even after the network has become static. 162 of the rules in class 1 evolve into dynamic-writer fixed points. In 130 of these rules the writer ends up doing a period two orbit (see Figure 4), in the others the writer ends up doing a period four orbit.

3 Repetitive Growth

The next most complex type of behaviour observed is repetitive growth. This is characterized by the feature that the structure continues to grow, while the writer is "trapped" within a particular region of the network -with the form of its surroundings changing periodically. Repetitive growth occurs because writer's surroundings induce it to generate more structure around itself of the same form. 840 of our rules eventually generate repetitive growth (see Figure
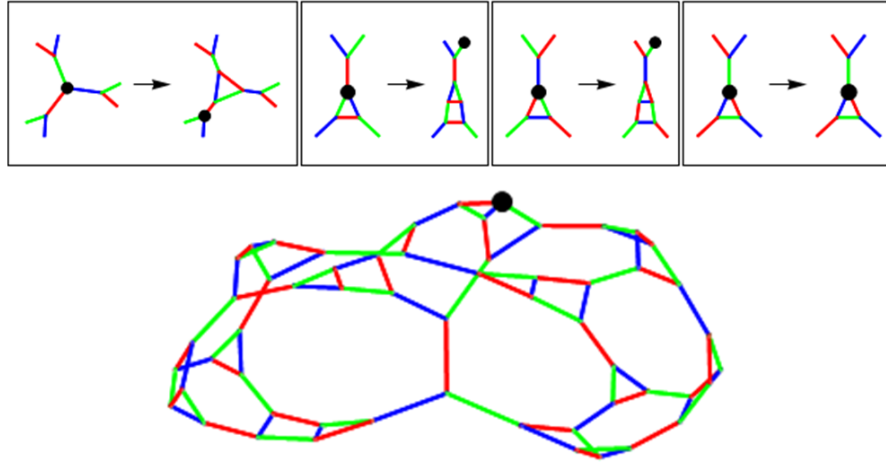
7

Figure 3: Of all our rules, this one generates the largest static network (with 56 vertices). It takes 26 updates for the cube to change into this network.
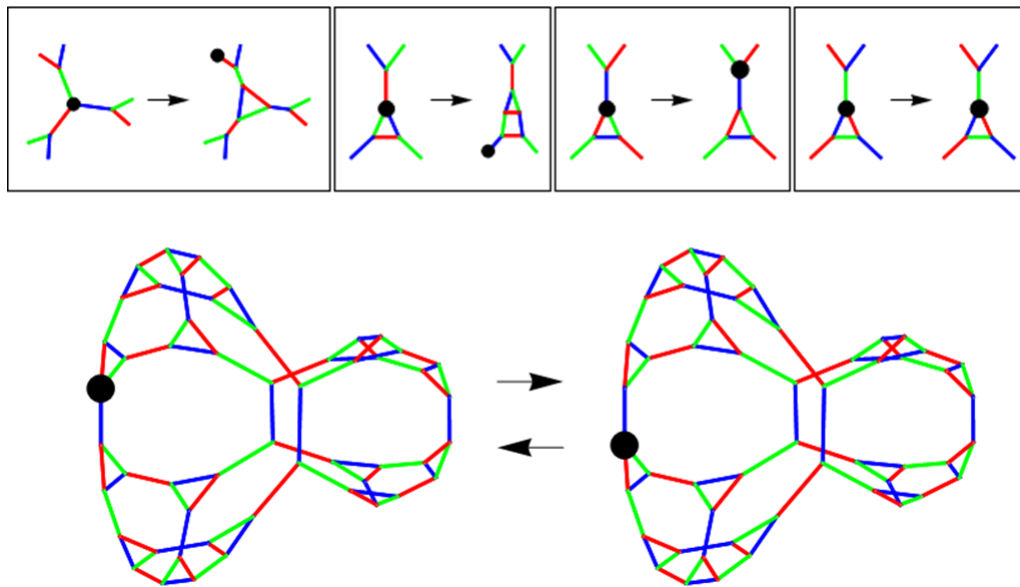

Figure 4: Of all our rules, this one takes the longest (34 updates) to reach a static network (which has 52 vertices). The writer continues to move in the period two orbit shown after this network has been generated.
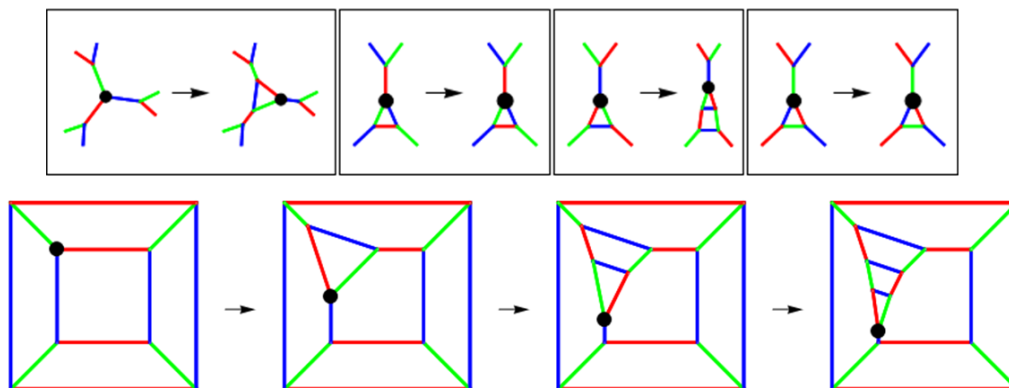
Figure 5: A rule that generates simple repetitive growth evolving over the first four updates. Once the writer's surroundings are such that a pair of its neighbors are linked by a blue edge, the writer performs an update which leads to similar surroundings occurring again on the next time step.

5).

Let us define repetitive growth more precisely. *Period p repetitive growth is occurring at time t* when the network grows arbitrarily large eventually, and there is a distance r such that the network on vertices within a distance r of the writers position at time t looks identical to the network on vertices within a distance r of the writer at time $t + p$, and the writer never moves to an old vertex more than a distance $r - 1$ from where it was at time step t during the interval $[t, t + p]$. The rule shown in Figure 5 falls into period one repetitive growth because after the first update the structure within a distance one of the writer always looks similar. Evolving this rule for $t > 0$ time steps results in a network with $8 + 2t$ vertices and a single triangle on the end of a long "ladder like" substructure.

Although one may test whether dynamics fit our definition using a computer, there are more practical ways to spot repetitive growth (see Figure 6). Networks undergoing repetitive growth tend to have an elongated linear, or circular shape because they are composed of a series of repeating substructures. Plotting the index of the writer over time is an excellent way to see dynamics. However it does depend on the way the vertices are indexed within the computer program, and this inevitably depends on more than just the pure topological dynamics of the system. In our case, when a vertex with index v in an L vertex network is replaced with a triangle, we give the vertex of this triangle with a red external
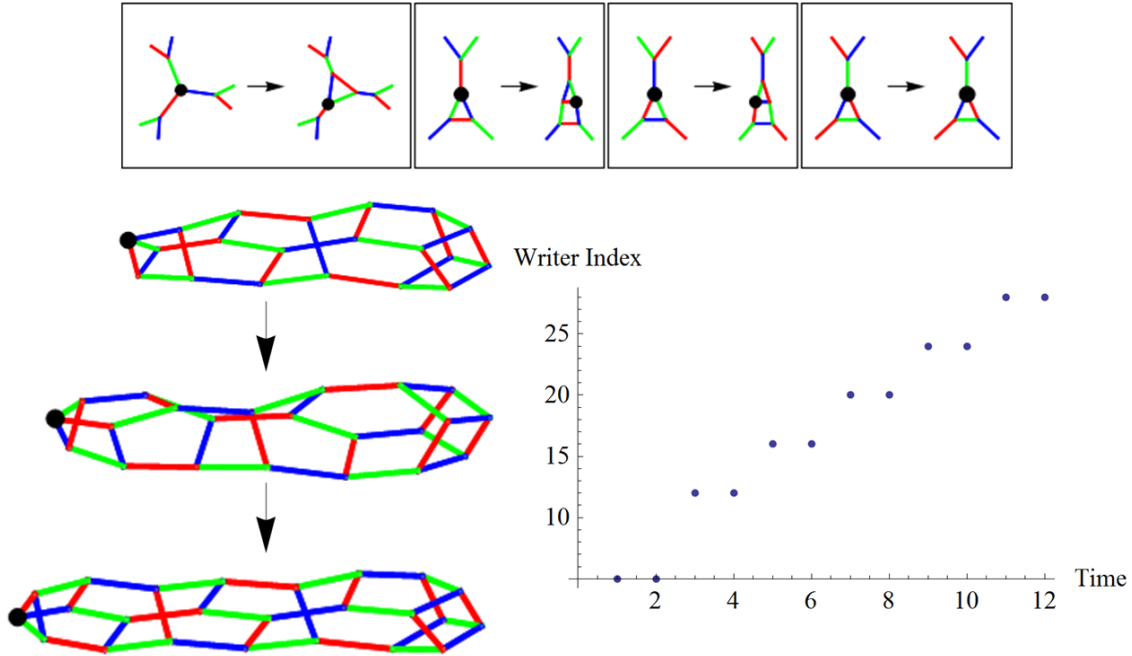
9

Figure 6: On the left we show the networks generated on time steps 6, 7 and 8 (reading downwards), when evolving from a cube. On the right we plot the index of the writer's position over time. Notice how the differences between the indices of the writer's position forms a periodic sequence. This is because, although network keeps growing, the writer keeps moving in the same relative way. This system has period two repetitive growth because the network induced on vertices within a distance two of the writer on time step 6 looks identical to the induced on vertices within a distance two of the writer at time step $6 + 2$.

edge an index v, and we give the other two vertices the other vertices of this triangle, with green and blue external edges, indices $L+1$ and $L+2$, respectively.

3.1   Repetitive growth with long transients

All but four of our repetitive growth rules settle into low (i.e., less than 12) period repetitive growth quickly (in less than 40 time steps). The other four rules eventually produce repetitive growth (see Figure 7), although it would perhaps be best to describe their behaviour as *complex* because the systems have extremely long transients within which the structures appear to grow in a pseudo-random way (this is reminiscent of elementary cellular automata
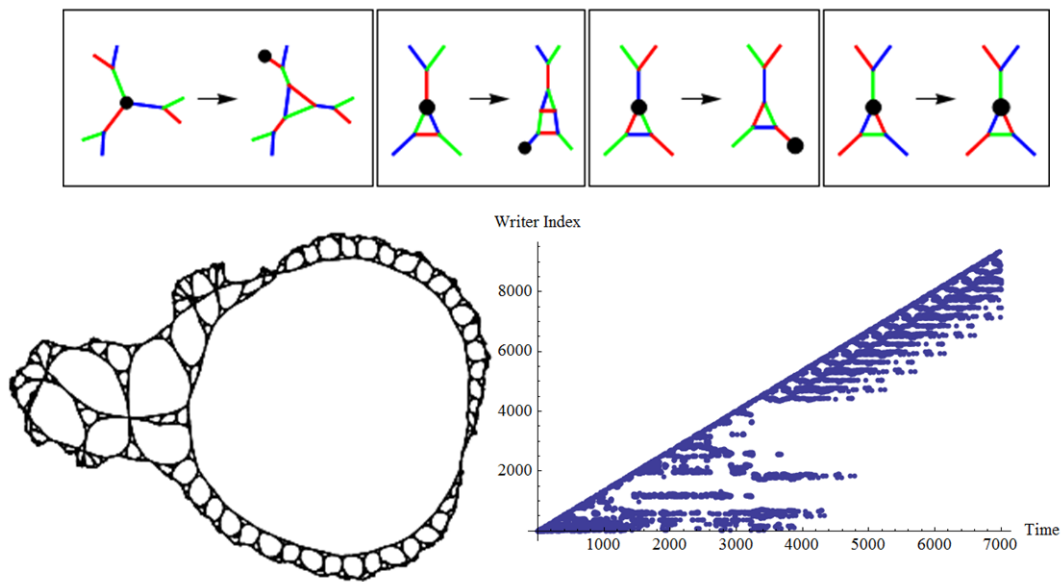
10

Figure 7: This system has complex behaviour for the first 4994 time steps, and then settles into period 454 repetitive growth. We show the rule at the top. On the left we show an uncolored plot of the network present on time step 7000. The large "handle" developing on the right of this picture is a symptom of the repetitive growth. The plot on the right shows the writer index over the first 7000 time steps.

11

110 [14]). Interestingly, evolving the rule shown in Figure 7 from other small networks produces different behaviour. The network known as $K_{3,3}$ is obtained by taking two clusters of three vertices, and linking each vertex in one cluster to each vertex in the other. Initiating this system from $K_{3,3}$ (instead of the cube) leads to dynamics which (again) eventually settle in period 454 repetitive growth, although this time the transient lasts for 29964 time steps. Another of the four rules which induce repetitive growth after a long transient is symmetrically equivalent to the rule shown in Figure 7, because it can be transformed into it by swapping the roles of the red and blue colors.

The other two rules which induce repetitive growth, after a long transient, are not symmetrically equivalent to the system shown in Figure 7, although they are symmetrically equivalent to each other. One of these rules is shown in Figure 8.

The ability of these systems to alter their behaviour after large amounts of time leads to the production of heterogeneous substructures. The fact that these rules produce a complex "ball" of network followed by a long one dimensional structure is reminiscent of the way plants grow by complicating the structure around the initial seed and then growing out a long stalk (see the figures from section 5).

4   Elaborate Growth

The remaining 130 rules produce elaborate patterns of growth, leading to self similar networks. It turns out that in each of these cases, the way the writer moves is effectively one dimensional. The writer repeatedly traverses a one dimensional "track", applying rewrite rules as it goes, and lengthening the track with each traversal. In most cases, the fact that the writer is confined to a one dimensional substructure can be inferred directly from the rules. In the other cases this behaviour is revealed by using appropriate visualization techniques.

The writer can move backwards and forwards on the one dimensional track it is confined to. Although the writer can move in complex ways we can sort the rules into two subclasses according to the general nature of the writer's movement. Either the writer moves around and around a closed loop, in which case we say the rule has *cyclic writer movement*, or the writer moves backwards and forwards over line, in which case we say the rule has *bouncing writer movement*.
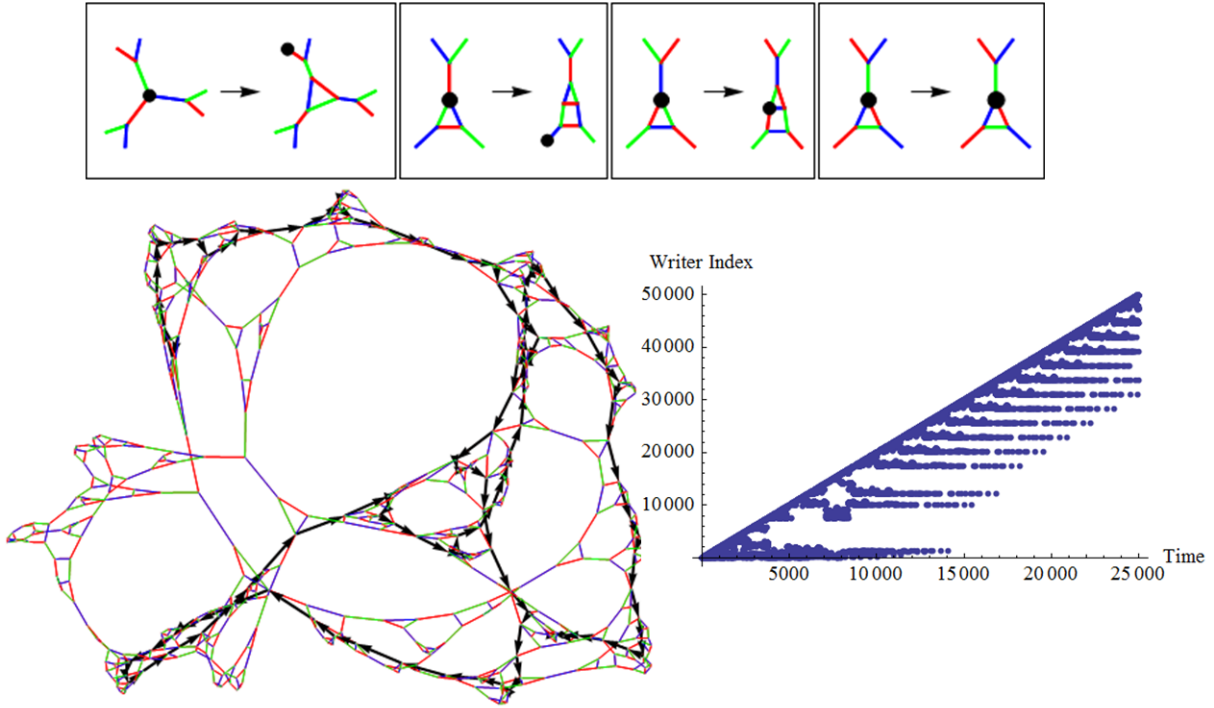
Figure 8: On the left we show the network obtained after evolving this rule for 300 time steps. The black arrows track the course that the writer has taken over the last 150 time steps (note that the positioning of these arrows depends on how we indexed our vertices). The writer movement looks unpredictable during this early transience. On the right we plot the writer's index over time. The transient behaviour lasts 17615 time steps, then the system settles into period 1355 repetitive growth. Similar behaviour ensues when evolving from other initial conditions.
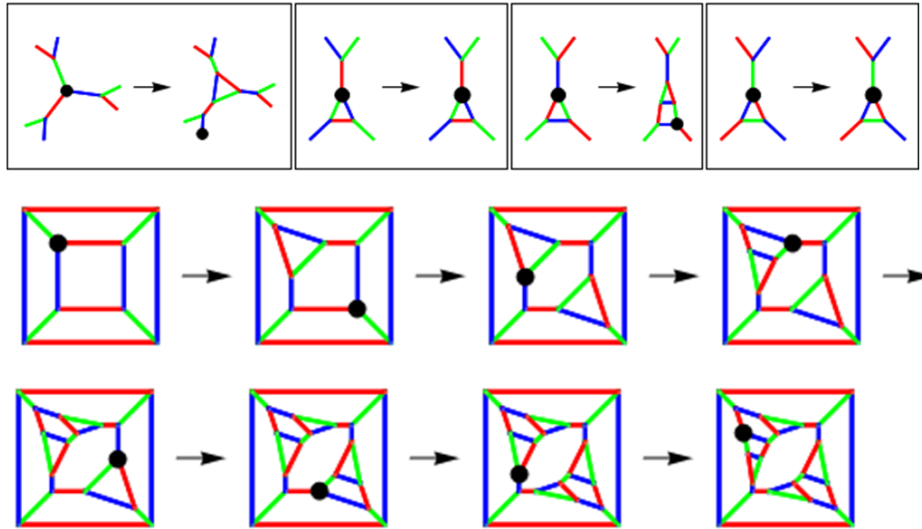
13

Figure 9: One of the simplest rules with cyclic writer movement, evolving for seven updates.

4.1 A simple case with cyclic writer movement

104 of the rules with elaborate growth have cyclic writer movement. We show one of the simplest in Figure 9. By looking at the rules of this system one can see that the writer must move in a one dimensional fashion. Whenever there are no red or green edges interlinking the writer's neighbors (i.e., whenever the system is not at a fixed point), the writer moves along a red edge and then a blue edge, and the writer's previous location is replaced with a triangle. The fact that the writer's movement can be expressed as a combination of steps along edges of *only two* different colors implies that the writer's movement is effectively one dimensional. In particular, since any sufficiently long path along edges of alternating red/blue color must eventually return to its starting point the writer must always remain confined to a cyclic track. In this case the cyclic track starts out as the inner face of our cube (see Figure 10). On each update the writer replaces its current vertex with a triangle and moves two edges clockwise around the developing track of edges with alternating red/blue colors. Theorem 1 describes the global rewrite operation which the writer effectively performs with each complete traversal of its cyclic track. The amount of time successive traversals take doubles.
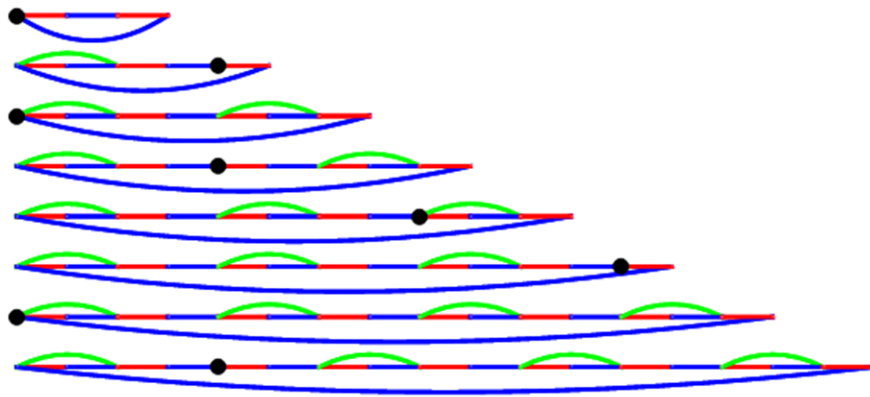
Figure 10: An alternative "one dimensional" representation of the dynamics of the rule shown in Figure 9. Successive rows, reading downwards, show the relevant part of the network on successive time steps. Here the *relevant part of the network* consists of network induced on the vertices which can be reached by moving along red or blue edges from the writer. We do not show green edges which are not part of triangles since these have no effect on dynamics. Each time step the writer replaces its current vertex with a triangle (effectively adding a red edge and a blue edge below a green arc) and moves two edges to the right.
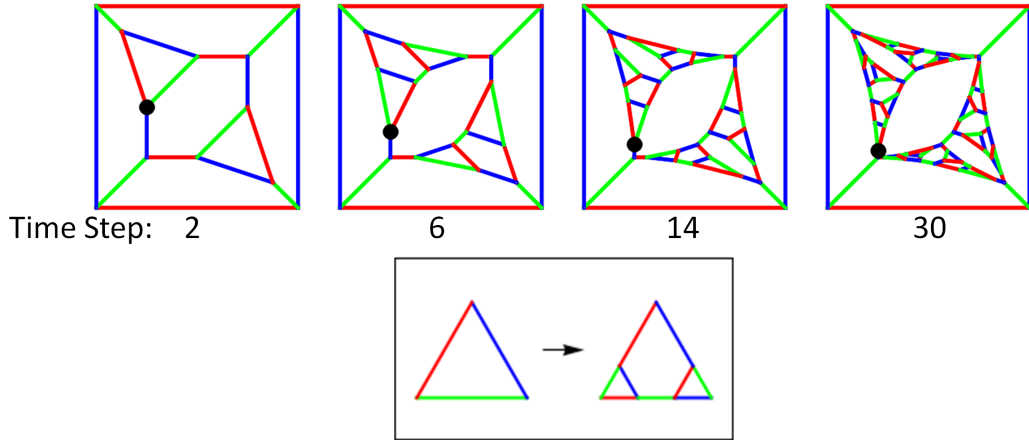
Figure 11: The network generated by the system with rules shown in Figure 9 on the first few time steps of the form $2^n - 2 : n \geq 2$. Between times of this form the writer makes a complete traversal of its cyclic path and effectively performs the rewrite operation shown at the bottom, globally.

Theorem 1 *The way the rule shown in Figure 9 evolves (with the cube as the initial condition) is such that for each $n \geq 2$, the network present on the $(2^{n+1} - 2)$th time step can be obtained by taking the network present on the $(2^n - 2)$th time step and then simultaneously replacing each vertex, with a red or blue edge interconnecting its neighbors, with a triangle (see Figure 11).*

Proofs are in the appendix.

4.2 A rule related to the golden ratio

In Figure 12 we show a rule with cyclic writer movement with a complicated looking growth rate. It turns out that the growth rate can be described exactly in terms of the golden ratio $\phi = \frac{1+\sqrt{5}}{2}$.

Theorem 2 *The number of vertices in the network obtained by evolving the rule shown in Figure 12 for $t \geq 0$ time steps (starting from the cube) is*

$$8 + 2 \left\lceil \frac{1}{\phi^2} \left\lfloor \frac{t}{2} \right\rfloor \right\rceil + 2 \left\lfloor \frac{t+1}{2} \right\rfloor .$$

It is pleasing that this seemingly complex growth rate can be described simply in terms of the golden ratio, because this number appears in so many
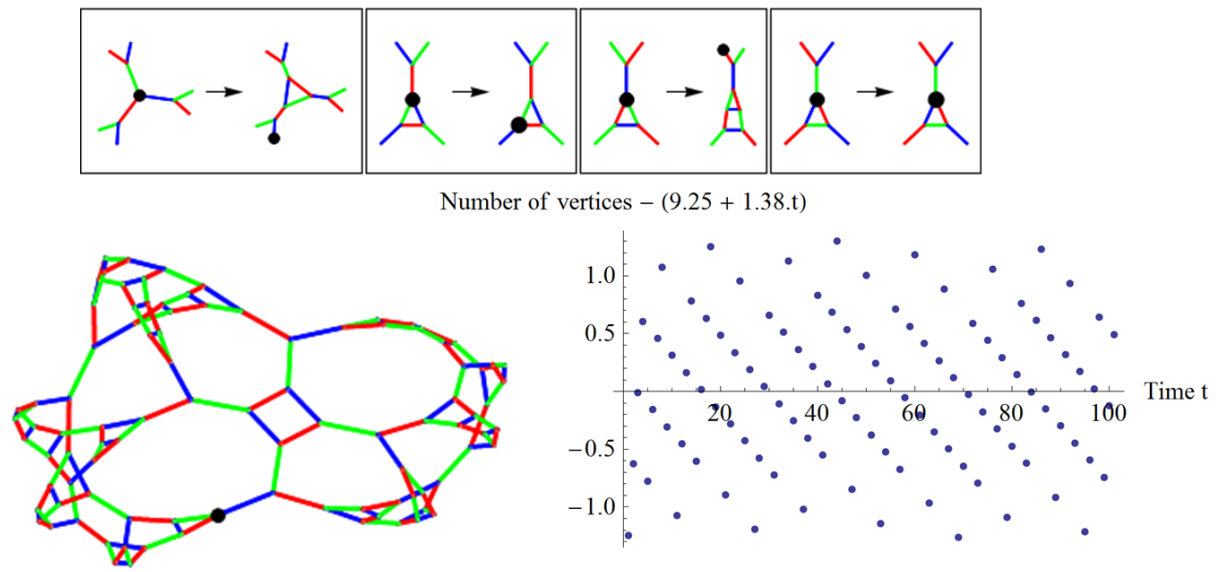
16

Number of vertices − (9.25 + 1.38.t)

Figure 12: On the left we show the network obtained by evolving this rule for 62 time steps. The number of vertices in the network at time step t grows approximately linearly with t. However the way the number of vertices deviates from its best linear fit looks complicated (as shown on the right).

17

interesting places in the natural world. This colored trinet automata has similar qualitative behaviour to the one considered in subsection 4.1. Once again the writer goes around and around a cyclic track consisting of edges with alternating red/blue color. Again the system can be reduced to a one dimensional string rewrite system. The proof to Theorem 2 is based on relating this system to the binary rewrite system with rules $0 \to 01, 1 \to 011$.

4.3   A rule with complex behaviour

The rule with cyclic writer movement shown in Figure 13 has a very complicated looking growth rate. Although this system can be transformed into a one dimensional rewrite system (in a similar way to the systems considered in subsections 4.1 and 4.2) we have not been able to derive a formula for its growth rate. This means this system, together with its equivalent red/blue reflection; stand out as the most complex rules in our entire set[2]. We are unable to predict the long term dynamics of these systems (although it appears the pseudo random growth pattern continues forever).

4.4   A simple case with bouncing writer movement

The 26 remaining rules with elaborate growth exhibit bouncing writer movement. In these rules the writer moves forwards and backwards along a one dimensional track, which grows with successive traversals. The rule in this class with the simplest behaviour is shown in Figure 14. This rule can be represented in a one dimensional manner similar to Figure 10, except that this time the writer is confined to moving on the red and green edges.

It appears that the writer effectively performs a global rewrite operation every time it makes a traversal of the one dimensional track it is confined to. Simulations suggest the that the network present on time step $2^{n+3} + n$ (where $n \geq 0$) can be obtained by taking the network present at time step $2^{n+2} + n - 1$ and then simultaneously replacing each vertex with a triangle that has no red or green edges interlinking its neighbors and is not part of the external face (see Figure 15).

Not all of the rules with elaborate growth can be reduced to one dimensional rewrite systems as directly as the examples we have considered, although the

---

[2]All of the rules which generate fixed points and repetitive growth have trivial long term behaviour. Plotting the writer index over time reveals significant regularities in all the other rules exhibiting elaborate growth
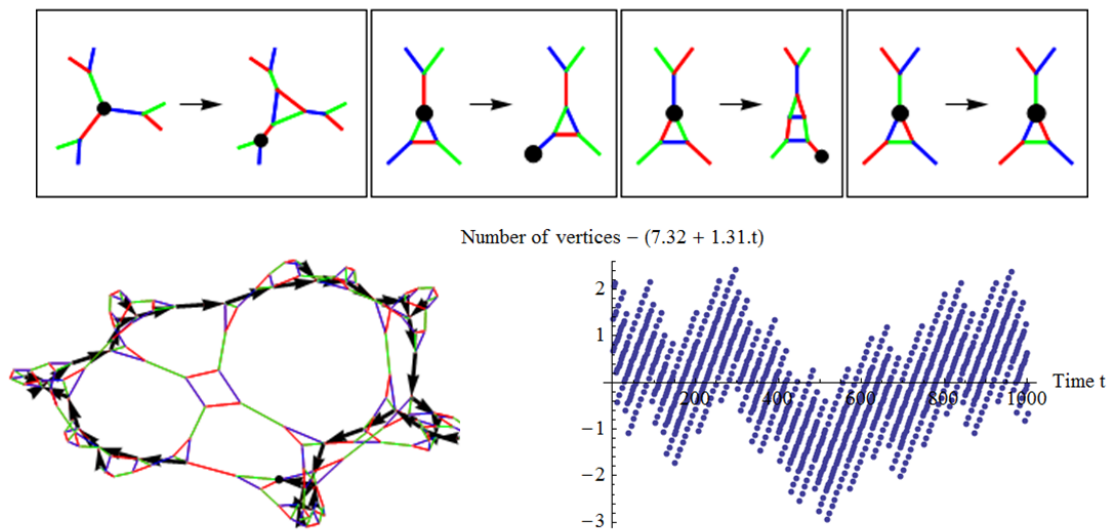
Figure 13: On the left we show the structure obtained by evolving this rule for 100 time steps. The black arrows track the positions of the writer over the previous 45 time steps. On the right we show how the growth rate of the number of vertices deviates from its best linear fit. This plot reveals considerable complexity.
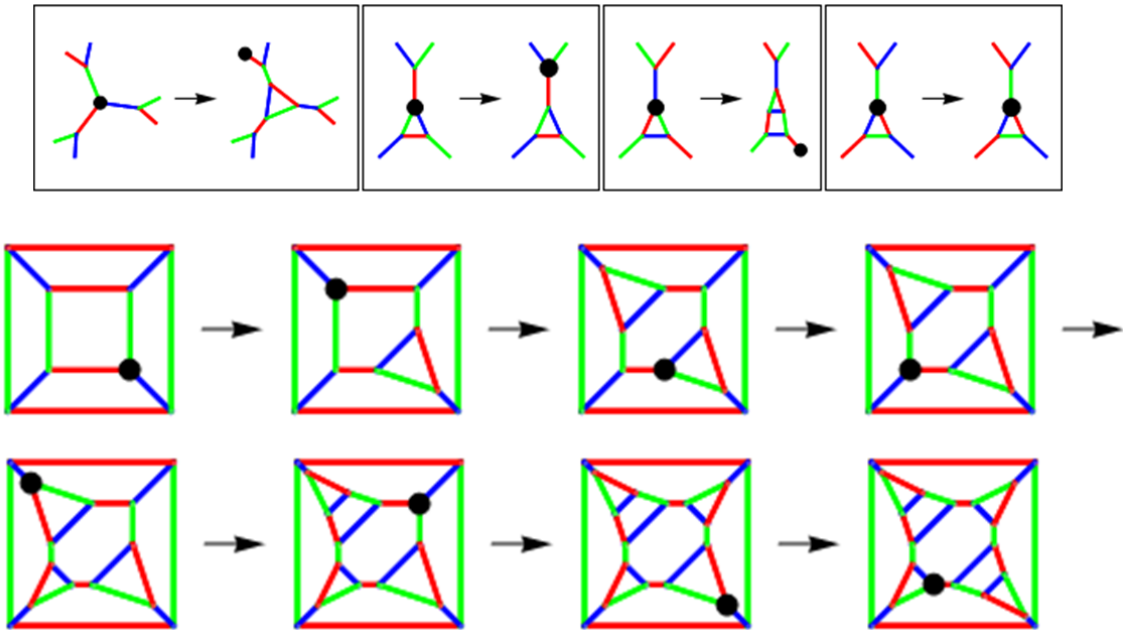
19

Figure 14: A rule with bouncing writer movement evolving for eight updates. The writer travels around the red and green edges of the track that starts out as the central face of the cube. The writer keeps bouncing off the red edge at the bottom of this face and reversing its direction of movement.
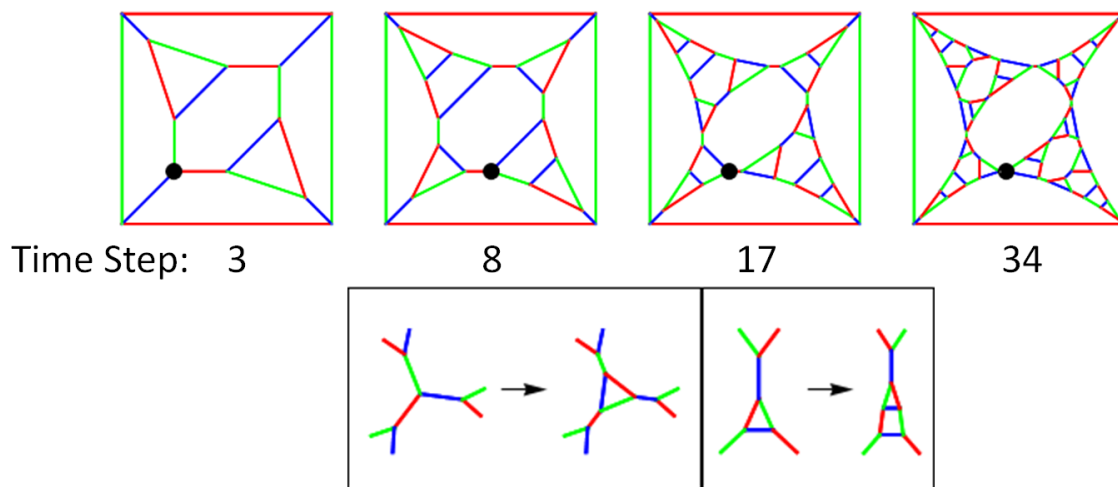
Figure 15: The network generated by the system with rules shown in Figure 14 on the first few time steps of the form $2^{n+2} - n - 1 : n \geq 0$. Between times of this form the writer makes a complete traversal of its linear path and both rewrite operations shown at the bottom, globally.

effectively one dimensional nature of the writer's movement can be revealed by plotting the trail followed by writer over several time steps (as in the left of Figure 13).

5   More general rules

The behaviour of colored trinets with more general rules does not always fall into the classes listed above. The set of 3888 rules we enumerated, and discussed above, did not allow the writer to take any action when the edge between its neighbors is green. If we remove this restriction we can find rules with four *active parts* such as the one shown in Figure 16. In this rule the writer continues to move in a random looking way (that is not "one dimensional" as in the rules with elaborate growth discussed above) for at least the first 100000 time steps. It is an open question whether this rule eventually settles into repetitive growth.

We can also consider rules which include more general kinds of rewrite operations, such as the one shown in Figure 17. In this system triangles can be replaced with vertices. Unlike all of the systems in our previously considered rule set (which either has approximately linear growth rates or reached fixed
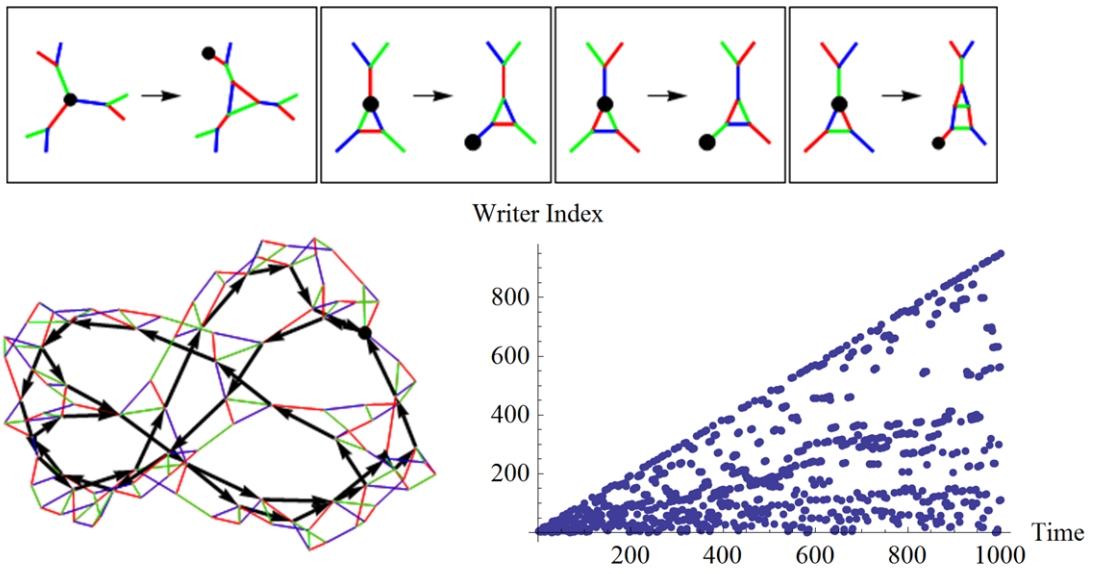
Figure 16: A rule with four active parts that has complex behaviour. On the left we show the structure obtained by evolving this rule for 100 time steps. The black arrows track the positions of the writer over the previous 45 time steps. On the right we plot the index of the writer over the first 1000 time steps.
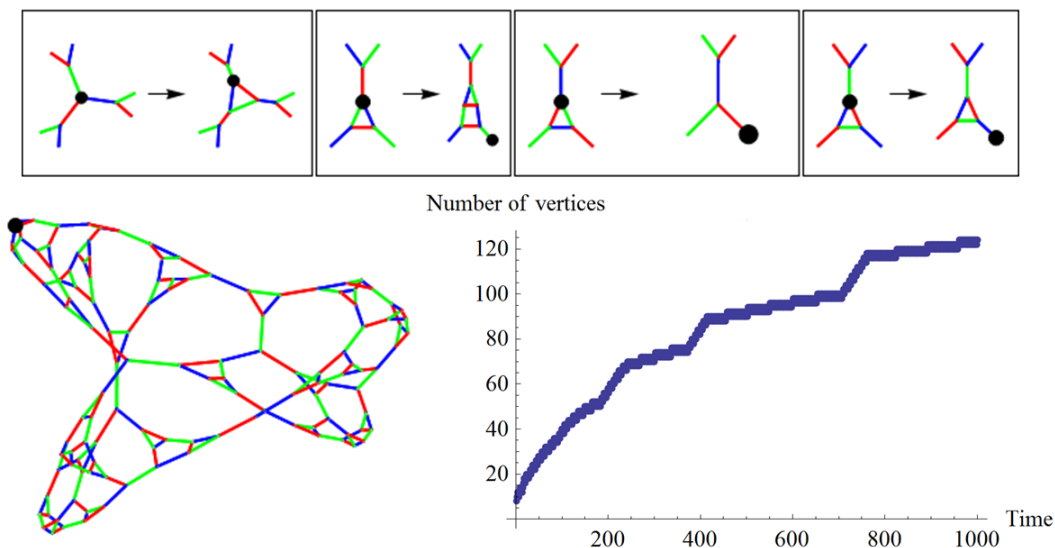
Figure 17: A system with more general rules that may involve replacing a triangle with a vertex. On the left we show the network present on time step 1000. On the right we plot the number of vertices over time.

points) the number of vertices in this case grows sub-linearly over time.

When exploring these more general kinds of rules, one finds many cases with sub-linear growth rates that exhibit repetitive or elaborate growth patterns. In many cases with sub-linear growth the number of vertices grows like the square root of the number of time steps elapsed. Other rules involving triangle shrinkage can lead to networks with shapes that oscillate over time. A trivial example is shown in Figure 18 (although rules exist with much higher period oscillations).

We can also consider rules including the so called *exchange* operation, which rewires an edge [13]. In Figure 19 we show a rule which uses this type of exchange operation. The rule depicted in Figure 19 has some interesting spatial properties. For a vertex v, the number of vertices a distance k from v grows exponentially with k. This suggests the rule generates hyperbolic (i.e., negatively curved) space. Indeed, this can be verified using the techniques described in [22] and [23]. According to these works, a connected network G with vertex set $V(G)$ is *scaled Gromov hyperbolic* (which means it is negatively curved on a
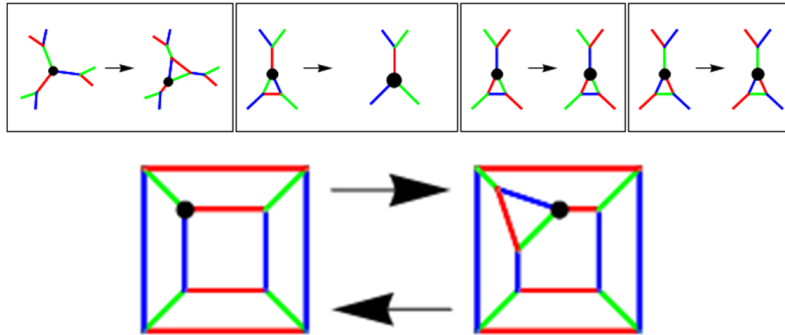
23

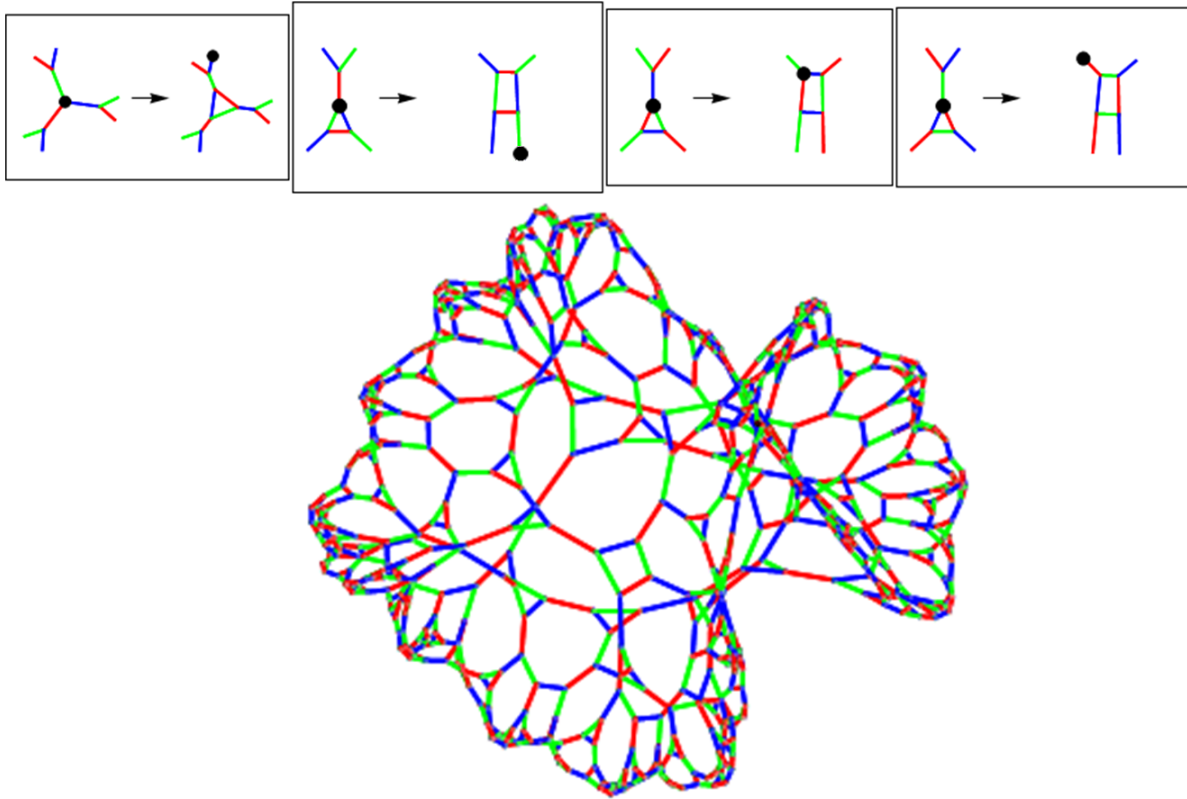Figure 18: In this system the network shape changes periodically.

Figure 19: A system which includes the *exchange* rule, which effectively replaces a triangle with a square. We show the rule at the top. The structure is generated by running this system for 500 time steps.

large scale) when $\frac{\delta(G)}{\text{dia}(G)} < \frac{3}{2}$. Here

$$\delta(G) = \max\{\min\{d(x,y)+d(y,z)+d(z,x) : x \in [b,c], y \in [a,c], z \in [a,b]\}, a,b,c \in V(G)\}.$$

Where $[i,j]$ denotes the shortest path from i to j, and $d(i,j)$ denotes the distance from i to j. Also $\text{dia}(G) = \max\{d(u,v) : u,v \in V(G)\}$ is the diameter of G. In our case, the network G obtained by evolving the rule shown in Figure 19 for 100 time steps has $\frac{\delta(G)}{\text{dia}(G)} = 1.28571$, and so it is indeed scaled Gromov hyperbolic.

The 3888 rules in the set that we initially focused on do not generate negatively curved space. This can be seen by noting the each of these rules generates planar networks, and planar networks have non-negative combinatorial curvature[3] (see the Gauss-Bonnet formula [24]). These rules preserve planarity because the only structural modification they allow is the replacement of a vertex with a triangle (and this can be thought of as chopping the corner off a polyhedron -a planarity preserving operation). On the other hand, the exchange operations within the rule depicted in Figure 19, allow it to generate a non-planar network, when initiated from the cube.

5.1 The resemblance with real world objects

It is remarkable how varied the forms produced by trinet automata with very similar rules can be. What is even more remarkable is that pictures similar to a diverse range of life forms and objects can easily be produced. In Figures 21, 22, 23,24 and 25 we show networks (produced by evolving colored trinet automata, from the cube) next to real world objects. In many cases there is significant similarity. When looking at these pictures it is important to keep in mind that a trinet may be represented by a wide variety of pictures, because the only relevant information is the connectivity. In other words, one could draw the colored trinets in a different way, and their resemblance to these objects would disappear. Never the less, each of these trinet pictures where produced using Mathematica's standard, *spring embedding* or *spring electrical embedding* graph drawing algorithm. This means the way the networks are drawn was chosen by the plotting algorithm to reflect the topology of the network. In this regard, one can argue that there is similarity between the real objects shown and the colored trinets pictured because the most natural ways to draw the networks

---

[3]Combinatorial curvature is a local measure, but the fact none of these networks have negative curvature on the large scale was verified by checking that each network which grows arbitrarily large has a well defined node shell dimension [16] less than 3.
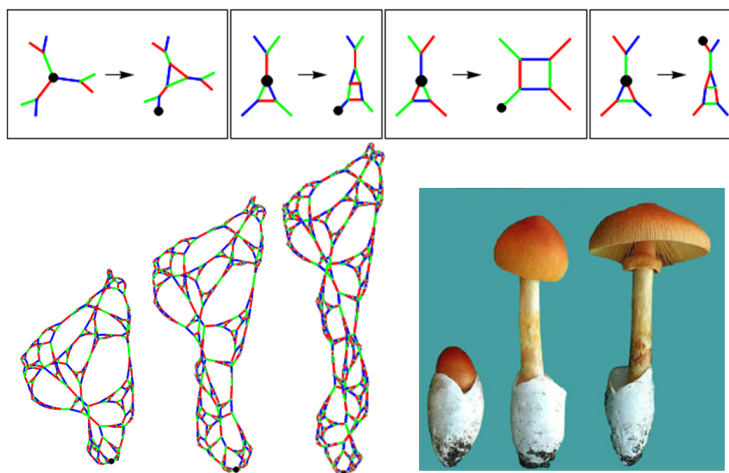
Figure 20: On the left we show the networks generated by evolving this system for 150, 200 and 250 time steps, evolving from the cube. On the right we show a photo depicting the development stages of a mushroom, taken from [25]. The cap is the first part of the mushroom to develop, and then a long stem grows our from the spore at the bottom. Our system generates a ball like network initially, and then grows out a linear structure. These pictures could be made to look more alike by manipulating the network embedding. The networks are drawn using Mathematica's default *Spring Electrical Embedding*
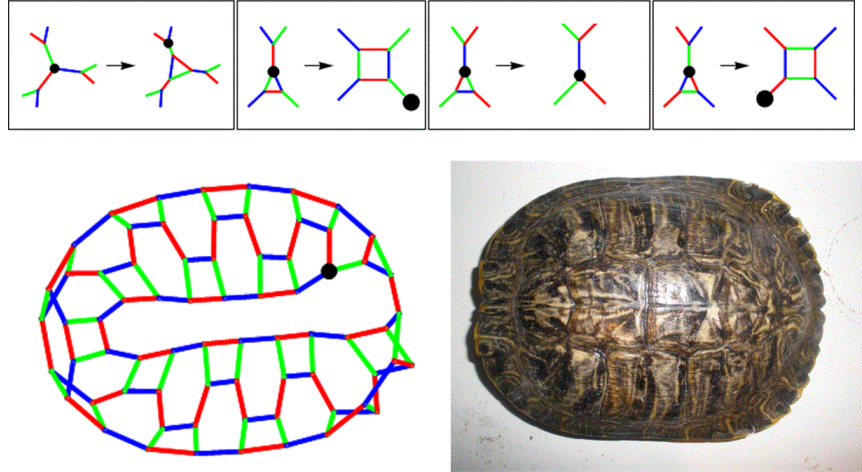
Figure 21: On the left we show the network obtained from evolving our rule for 46 time steps, starting from the cube. The network drawn using mathematica's standard *spring embedding* algorithm. On the right we show a photo of a turtle shell taken from [26].

yield pictures that look like the objects. Figures 21, 22, 23,24, and 25 also show topological similarities between the colored trinets and the real objects.

6 Conclusion

We have explored many aspects of the behavior of colored trinet automata. We found that the systems in our initial rule space could be sorted into three classes - fixed, repetitive growth and elaborate growth. We have shown that each class can be described in terms of writer movement. When we look at the dynamics of more general rules we find other types of behavior. In particular, we find rules which generate sub-linear growth, persistent complex behaviour, periodically changing networks and hyperbolic space. We have also noted that many trinets with more general rules can produce forms remarkably similar to a diverse range of real world objects.

As minimal models capable of growing complex structure, these systems should have some applications. The systems could be realized by creating a robot which pulls itself along chains, ropes or silk and drops lines (like a spider)
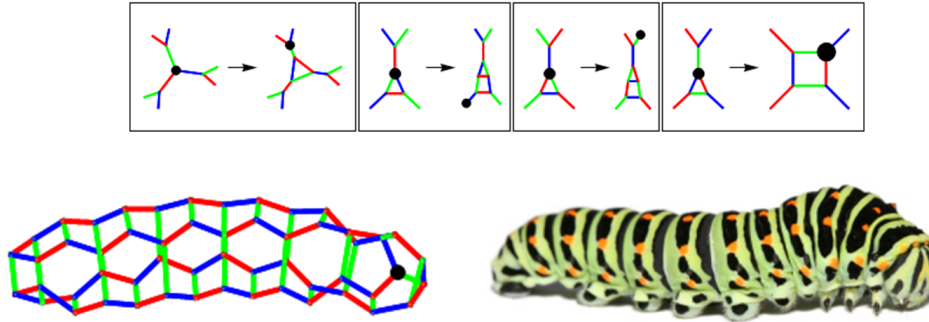
27

Figure 22: On the left we show the network obtained from evolving our rule for 34 time steps, starting from the cube. The network drawn using mathematica's standard *spring embedding* algorithm. On the right we show a photo of a caterpillar taken from [27].

to triangulate vertices. This type of realization could perhaps be useful for weaving or construction work. The ability of these systems to produce exotic network structures with relative ease could make them useful in network design. In existential graph theory there are many open questions as to whether large trinets exist with particular properties and it will be interesting to see whether our systems can resolve any of these issues.

It will also be interesting to see if these systems have any applications in modeling. The way structures grow under repetitive growth is reminiscent of the way plants grow (see Figure 20). Many of the networks produced by rules with elaborate growth rules seem reminiscent of polygonal networks in cracked earth or foams. The general idea of having a single writer that rewires a complex network is reminiscent of brains and databases where the connectivity is altered to store information.

There many directions that this work can be taken in the future. We have explored the dynamics of many colored trinet automata and identified many kinds of behavior, however a more general classification scheme is evidently required to deal with the systems we described in section 5. Exploring more general rules will be exciting, and will surely yield systems with other interesting kinds of behavior and applications. There are also many unanswered questions about the systems presented here, for example: is there a simple formula for the growth rate of system shown in Figure 13? Does the complex behavior in Figure 16 persist forever ? How can rules be characterized according to the structural

Figure 23: At the top we show the rule. In the middle we show the network obtained from evolving our rule for 1600 time steps, starting from the cube. The network drawn using mathematica's standard *spring electrical embedding* algorithm (we have not shown edge colors). At the bottom we show a photo of sliced Telescopium seashells taken from [28].

29

Figure 24: On the left we show the network obtained from evolving our rule for 1430 time steps, starting from the cube. The network drawn using mathematica's standard *spring embedding* algorithm (we have not shown edge colors). On the right we show a photo of a coronal slice through a human brain taken from [29].
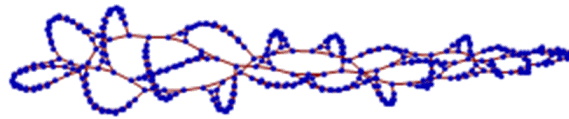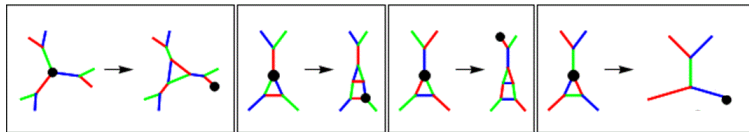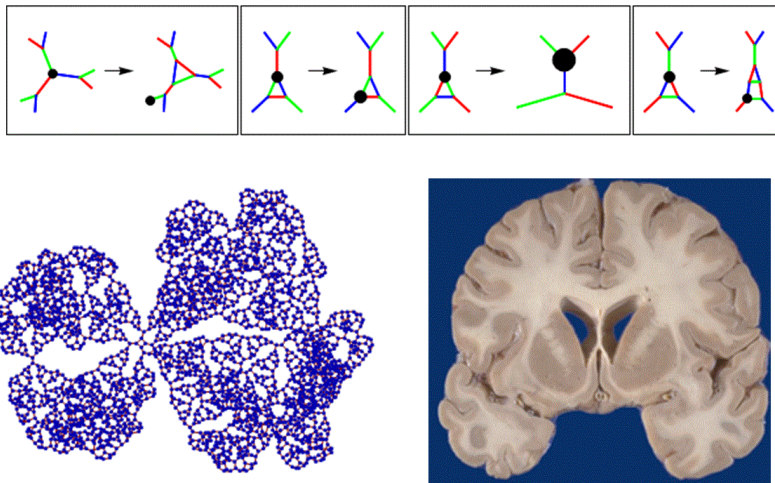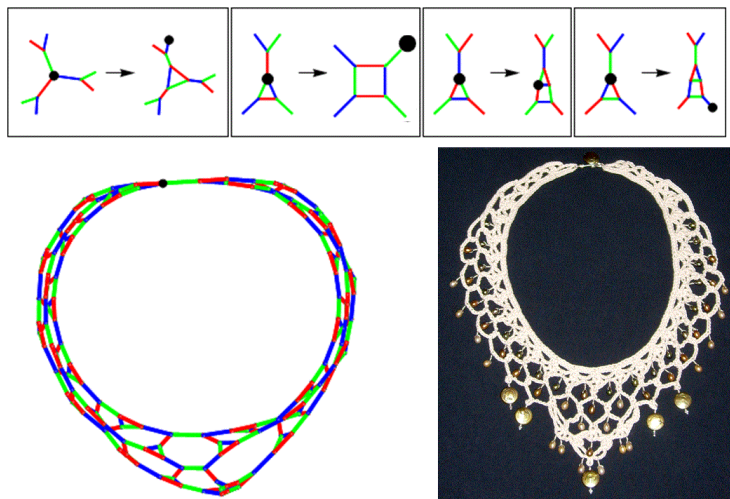

Figure 25: On the left we show the network obtained from evolving our rule for 100 time steps, starting from the cube. The network drawn using mathematica's standard *spring electrical embedding* algorithm. On the right we show a photo of a Necklace made from crochet lace, pearls, and sterling silver taken from [30].

30

properties of the networks they produce ?

Acknowledgments

A   Appendix

A word w with alphabet $\mathcal{A}$ is a sequence $w_0...w_{m-1}$ of $m \geq 0$ elements $w_i \in \mathcal{A}$ from w. The length $|w|$ of such a word is the number elements in it (m in this case). The empty word is a word of length 0. Notice that we index the elements in our words w from 0 to $|w| - 1$. Let $\mathcal{A}^*$ denote the set of all words with alphabet $\mathcal{A}$. Let

$$wx = w_0...w_{|w|-1}x_0...x_{|x|-1} \tag{1}$$

denote the word formed by concatenating words w and x. If $k > 0$ then w^k denotes the word obtained by concatenating k copies of w together (e.g. $w^3 = www$). Also w^0 is the empty word.

We let r, b and g denote the colors red, blue and green, respectively. We define the *type* $\rho(v)_G$ of a vertex v in a colored trinet G such that, if none of v's neighbours are interlinked then $\rho(v)_G = 0$ and if exactly one pair of v's neighbors are interlinked, by an edge of color $c \in \{r, b, g\}$ then $\rho(v)_G = c$.

We let (X^t, G^t) denote the state of a system at time step t (i.e., after t updates, staring from the cube). Here X^t is the writer vertex on time step t and G^t is the colored trinet on time step t. In each of the systems we discuss, we start with a cube as our initial network, G^0 and each structural modification made to the network during an update involves replacing a single vertex with a triangle. It follows that no vertex will have more than one pair of interlinked neighbors, in any network that occurs during the evolution of one of our systems. It follows that the type, $\rho(v)$ of each vertex is well defined. In our systems the current type of the writer vertex determines how the writer will move and whether or not it will replace its old location with a triangle.

The following notation will help us to describe motion. For a word $w \in \{r, b, g\}^*$ we let $v[w]$ denote the vertex one arrives at by starting at v and following the sequence of edges whose colors are described by w. More precisely,

for a color $c \in \{r, b, g\}$ and vertex v in a colored trinet G let $v[c]$ denote the vertex linked to v by an edge of color c. If w is the empty word, we let $v[w] = v$. For a word of the form xc (where $x \in \{r, b, g\}^*$ and $c \in \{r, b, g\}$) we let $v[xc] = v[x][c]$.

A.1 Proof of Theorem 1

Let (X^t, G^t) denote the state of the system on time step t, evolving under the rules shown in Figure 9 (starting with G^0 as the cube). Let $U(t)$ denote the sequence $U(t)_0...U(t)_{|U(t)|-1}$ of vertices in G^t that one reaches by starting at the writer vertex X^t and moving along edges with color alternating between red and blue. More precisely, $\forall k \in \{0, 1, ..., |U(t)| - 1\}$ we have

$$U(t)_k = \begin{cases} X^t[(rb)^{k/2}] & \text{if k is even,} \\ X^t[r(br)^{(k-1)/2}] & \text{if k is odd.} \end{cases} \tag{2}$$

The length of this sequence, $|U(t)| = 2.\min\{k > 0 : X^t[(rb)^k] = X^t\}$, is the number of vertices in the cycle of edges with colors alternating between red and blue, that contains the writer. Let

$$J(t) = J(t)_0...J(t)_{|J(t)|-1} = \rho(U(t)_0)_{G^t}...\rho(U(t)_{|U(t)|-1})_{G^t} \tag{3}$$

be the sequence of types of vertices in $U(t)$. In other words, $|J(t)| = |U(t)|$ and $J(t)_k = \rho(U(t)_k)_{G^t}$, $\forall k \in \{0, 1, ..., |J(t)| - 1\}$.

Lemma 3 $\forall T \geq 1$, we have $J(2T) = (bgr0)^{T+1}$ and $J(2T + 1) = 00(bgr0)^{T+1}$.

Proof For $T = 1$ we have $J(2) = bgr0bgr0$ and $J(3) = 00bgr0bgr0$, and so the result holds in this case.

Suppose the result holds for some $T \geq 0$.

Now, $J(2T + 1) = 00(bgr0)^{T+1}$, by hypothesis. This means the writer has type $\rho(X^{2T+1})_{G^{2T+1}} = J(2T+1)_0 = 0$ on time step $2T+1$. Now, by consulting the rules behind the system one can see that updating, from this state, will consist of the writer moving along a red edge, and then a blue edge (to the position of vertex $U(2T + 1)_2$) and then replacing its previous location with a triangle. This implies that the type sequence $J(2T + 2)$, for the next time step, can be constructed by taking $J(2T + 1)$ and rotating it two steps to the left (to represent to writer's movement) to get $(bgr0)^{T+1}00$, and then replacing the writers previous position (i.e., vertex corresponding to the penultimate element

in $(bgr0)^{T+1}00)$ with a triangle (bgr) to get $J(2T+2) = (bgr0)^{T+1}bgr0 = (bgr0)^{T+2}$.

On time step $2T + 2 = 2(T + 1)$ the writer has type $\rho(X^{2(T+1)})_{G^{2(T+1)}} = J(2(T+1))_0 = b$. Now, by consulting the rules behind the system one can see that updating, from this state, will consist of the writer moving along a red edge, and then a blue edge (to the position of vertex $U(2(T+1))_2$) and then replacing its previous location with a triangle. This implies that the type sequence $J(2(T+1)+1)$, for the next time step, can be constructed by taking $J(2(T+1))$ and rotating it two steps to the left (to represent to writer's movement) to get $r0(bgr0)^{T+1}bg$, and then replacing the writers previous position (i.e., vertex corresponding to the penultimate element in $r0(bgr0)^{T+1}bg$) with a triangle (bgr). It follows that the type sequence on the next time step will be $J(2(T+1)+1) = 00(bgr0)^{T+1}bgr0 = 00(bgr0)^{T+2}$. The reason for this is that, immediately before the update, the writer's previous position (which corresponds to the penultimate element in the word $r0(bgr0)^{T+1}bg$) is part of a pre-existent triangle (together with the vertices corresponding to the last and first elements in the word $r0(bgr0)^{T+1}bg$), and when the writers previous position is replaced with a triangle this pre-existent triangle becomes a square (thus effecting the types of the other vertices which used to be part of it). The effect of this is that we replace the rightmost b in our word $r0(bgr0)^{T+1}bg$ with bgr (to represent the replacement of the writers previous position with a triangle) and we replace the rightmost g and the leftmost r, in our word, with 0 to represent how the types of these vertices change when the triangle they used to be part of is expanded apart by the addition of the new triangle. These modifications lead to the word $J(2(T+1)+1) = 00(bgr0)^{T+1}bgr0$.

Now we have shown that our result holds for $T = 1$, and if our result holds for any $T \geq 1$ then it holds for $T + 1$. We can hence prove the result holds $\forall T \geq 1$ using induction with T.

Corollary 4 $\forall t \geq 0$, the update at time step t consists of the writer moving along a red edge and then a blue edge, and replacing its previous position with a triangle.

Proof For $t = 0$ and $t = 1$ this result is clearly true. Now suppose $t \geq 2$.

If t is even then $\exists T \geq 1$ such that $t = 2T$, and so Lemma 3 implies that $J(t) = J(2T) = (bgr0)^{T+1}$, and so $\rho(X^t)_{G^t} = J(2T)_0 = b$, and so (according to the rules of our system) the update at time step t consists of the writer moving

along a red edge and then a blue edge, and replacing its previous position with a triangle.

If t is odd then $\exists T \geq 1$ such that $t = 2T + 1$, and so Lemma 3 implies that $J(t) = J(2T) = 00(bgr0)^{T+1}$, and so $\rho(X^t)_{G^t} = J(2T)_0 = 0$, and so (according to the rules of our system) the update at time step t consists of the writer moving along a red edge and then a blue edge, and replacing its previous position with a triangle.

We say that $n \geq 2$ satisfies property (1) when $J(2^n - 2) = (bgr0)^{2^{n-1}}$, and for each vertex v in $G^{2^n - 2}$ of type $\rho(v)_{G^{2^n-2}} \in \{r, b\}$ there is a constant $q(v) \in \{0, 1, ..., 2^n - 1\}$ such that $v = X^{2^n-2}[(rb)^{q(v)}] = U(2^n - 2)_{2q(v)}$ (in other words, each vertex of type r or b can be reached by repeatedly doing rb hops from the writer's current position X^{2^n-2}). Here an rb hop consists of moving along a red edge, and then a blue edge.

Suppose that $n \geq 2$ satisfies property (1). We will show that this implies that $n + 1$ satisfies property (1), and that $G^{2^{n+1}-2}$ can be obtained by taking G^{2^n-2} and simultaneously replacing each vertex of type r or b with a triangle.

Now according to Corollary 4 the writer will visit the vertices

$$X^{2^n-2}[(rb)^0], X^{2^n-2}[(rb)^1], ..., X^{2^n-2}[(rb)^{2^n-1}]$$

in sequence, over the next 2^n updates, and each such vertex will be replaced with a triangle. It follows that the network $G^{2^n-2+2^n} = G^{2^{n+1}-2}$ present 2^n time steps later can be obtained by taking G^{2^n-2} and simultaneously replacing each vertex of type r or b with a triangle. Moreover, each vertex v' of $G^{2^{n+1}-2}$ which is of type r or b will be part of a triangle which was created during an update which occurred upon a time step in $\{(2^n - 2), (2^n - 2) + 1, ..., (2^n - 2) + 2^n - 1\}$. It follows that each such vertex v' will be reachable by doing some number $q'(v') \geq 0$ of rb hops from the writer's current position $X^{2^{n+1}-2}$. Moreover, Lemma 3 implies that $J(2^{n+1} - 2) = (bgr0)^{2^n}$. This implies that each of our vertices v' (with type r or b) can be reached by doing a number of rb hops $q'(v')$ such that $q'(v) \leq 2^{n+1} - 1$. It follows that $n + 1$ satisfies property (1).

Now $n = 2$ clearly satisfies property (1). It follows, by induction with n that $\forall n \geq 2$ we have n satisfies property (1) and $G^{2^{n+1}-2}$ can be obtained by taking G^{2^n-2} and simultaneously replacing each vertex of type r or b with a triangle. QED.

A.2 A complete description of the behaviour of the rule shown in Figure 9

Theorem 1 shows how the system pictured in Figure 9 can be viewed as evolving via global replacement operations. Under this system we can describe exactly how the network structure and writer position change with time.

Let $\mathbb{N}$ and $\mathbb{N}_0$ denote the sets of positive and non-negative integers respectively.

For any pair of non-negative integers $x, y \in \mathbb{N}_0$ let $\zeta(x, y) = \min\{x + 2y, 2x\}$.

We describe the state of a trivalent network automata by listing the vertex set, set of red edges, set of blue edges, set of green edges and the position of the writer, in that order.

Suppose $T \in \mathbb{N} : T \geq 2$. Let $n(T) = \max\{m \in \mathbb{N} : 2^m - 2 \leq T\}$ and $t(T) = T - (2^{n(T)} - 2)$.

Now let us define the state $H(T) = (V(T), E_r(T), E_b(T), E_g(T), W(T))$ at time T such that:

1. The set of vertices is $V(T) = \{-4, -3, .., 2^{n(T)+1} + 2.t(T) - 1\}$.

2. The set of red edges is $E_r(T) = \{\{-1, -4\}, \{-3, -2\}\} \cup \{\{i, i+1 \mod 2^{n(T)+1} + 2.t(T)\} : i \in \{1, 3, .., 2^{n(T)+1} + 2.t(T) - 1\}\}$.

3. The set of blue edges is $E_b(T) = \{\{-4, -3\}, \{-2, -1\}\} \cup \{\{i, i+1\} : i \in \{0, 2, .., 2^{n(T)+1} + 2.t(T) - 2\}\}$.

4. The set of green edges is $E_g(T) = \{\{-4, 0\}, \{-3, \zeta(2^{n(T)-1}, t)\}, \{-2, \zeta(2^{n(T)}, t)\}, \{-1, \zeta(2^{n(T)+1} - 2^{n(T)-1}, t)\}\} \cup \{\{\zeta(2^{\alpha+2}.\beta + 2^\alpha, t), \zeta(2^{\alpha+2}.\beta + 2^\alpha + 2^{\alpha+1}, t)\} : \alpha \in \{0, 1, .., n(T) - 2\}, \beta \in \{0, 1, .., 2^{n(T)-\alpha-1} - 1\}\} \cup \{\{4m+1, 4m+3\} : m \in \{0, 1, .., t(T) - 1\}\}$.

5. The writer position is $W(T) = 4.t(T) + 1$.

Now we claim, for each $T \geq 2$, that $H(T)$ is equivalent to the state of $S(T)$ of the system obtained by evolving the rule shown in Figure 9 for T time steps, starting from the cube. More precisely, we claim that there is a graph isomorphism from the network of $H(T)$ to the network of $S(T)$ that preserves the position of the writer and the colors of the edges. Such an isomorphism is a bijection from $V(T)$ to the vertex set of the network in state $S(T)$ which sends the writer position $W(T)$ to the position of the writer in $S(T)$, and which has the property that, a pair of vertices are linked by an edge of a particular

color, in $H(T)$, if and only if the images of these vertices under our bijection, are linked by an edge of the same color.

This equivalence of $H(T)$ and $S(T)$ can be seen by induction. It can easily be verified in the case where $T = 2$. Suppose the result holds for some generic $T \geq 2$ (i.e., suppose that $H(T)$ is equivalent to $S(T)$). Now we simply have to show that the result also holds for $T + 1$, and we can use induction with T to complete the proof. We can show the result holds for $T + 1$ by showing that the state H^* obtained by updating $H(T)$ under the system shown in Figure 9 is isomorphic to $H(T + 1)$.

The state H^* obtained by updating $H(T)$ is equal to the state obtained by taking $H(T)$, and moving the writer along a red edge followed by a blue edge, and then performing a structural modification upon the network. This structural modification (which yields the network of H^* from the network of $H(T)$) consists of the deleting the vertex $W(T)$ (and edges[4] $\{u_b, W(T)\} \in E_b(T), \{u_g, W(T)\} \in E_g(T), \{u_r, W(T)\} \in E_r(T)$) that were previously incident upon this vertex), and then adding three new vertices, $W(T)_b, W(T)_g, W(T)_r$, together with new blue edges $\{u_b, W(T)_b\}, \{W(T)_g, W(T)_r\}$, and new green edges $\{u_g, W(T)_g\}, \{W(T)_b, W(T)_r\}$, and new red edges $\{u_r, W(T)_r\}, \{W(T)_b, W(T)_g\}$. The net effect of these substitutions is to replace $W(T)$ with a triangle upon vertices $W(T)_b, W(T)_g, W(T)_r$.

We can show H^* is equivalent to $H(T + 1)$ by explicitly constructing an isomorphism f from H^* to $H(T + 1)$ which preserves the position of the writer and the colors of the edges. Here f is a bijection from the vertex set of H^* to the vertex set of $H(T + 1)$ which is defined as follows:

- For each vertex x of H^* such that $x < W(T) = 4.t(T) + 1$, we have $f(x) = x$.

- For each vertex x of H^* such that $x > W(T) = 4.t(T) + 1$, we have $f(x) = x + 2$.

- $f(W(T)_b) = 4.t(T)+1, f(W(T)_g) = 4.t(T)+2$ and $f(W(T)_r) = 4.t(T)+3$.

A.3   Proof of Theorem 2

Let (X^t, G^t) denote the state of the system on time step t, evolving under the rules shown in Figure 12 (starting with G^0 as the cube).

---

[4]Here u_b, u_g and u_r denote the vertices which are linked to the writer by edges of colors blue, green and red, respectively.
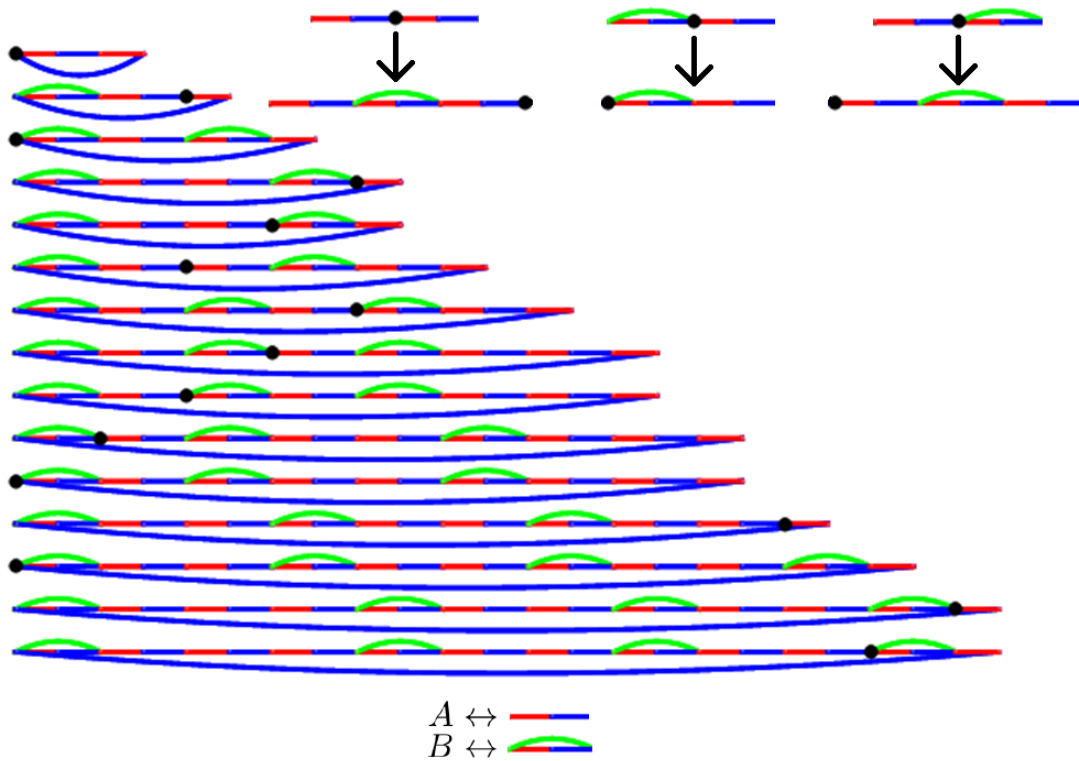
Figure 26: A representation of the (essentially one dimensional) dynamics of the system illustrated in Figure 12. This picture represents how the relevant part of the network changes over successive time steps, like Figure 10. The effective rewrite rules are shown at the top right. At the bottom we illustrate the symbolic substitutions we shall use to discuss this system algebraically.

Our method of proof will be to convert the system into a one dimensional rewrite system, and then derive a formula for the type $\rho(X^t)_{G^t}$ of the writer vertex at time t. We can use this information to characterize the number of vertices in G^t, because the number of vertices in G^t only depends on the type the writer had on previous time steps (i.e. how much the network grew on previous time steps). The way this system can be converted to a one dimensional rewrite system is illustrated in Figure 26. Rows of this figure (reading downwards) show the relevant part[5] of the network on subsequent time steps.

Let A and B denote the two kinds of vertices illustrated at the bottom of Figure 26. We can define these more rigorously, by saying that a vertex v in G^t of kind A is a vertex which is not part of a triangle, that can be expressed as $v = X^t[r(br)^k]$, for some $k \geq 0$ and a vertex v of kind B is a vertex which is part of a triangle, that can be expressed as $v = X^t[r(br)^k]$, for some $k \geq 0$. Now we can represent our system as a word of symbols that evolves like a tag system [14]. In particular, the word $s(t) \in \{A, B\}^*$ gives the sequence of kinds of vertices one visits by starting at $X^t[r]$ (the vertex linked to the writer by a red edge) and repeatedly doing br hops. More precisely $s(t) = s(t)_0 s(t)_1 ... s(t)_{|s(t)|-1}$ and $\forall k \in \{0, 1, ..., |s(t)| - 1\}$ we have

$$s(t)_k = \begin{cases} A & \text{if } \rho(X^t[r(br)^k])_{G^t} = 0, \\ B & \text{if } \rho(X^t[r(br)^k])_{G^t} = g. \end{cases} \tag{4}$$

Here the length of our word is $|s(t)| = \min\{k > 0 : X^t[(rb)^k] = X^t\}$, which is the number of red edges in the alternating rb cycle.

As the system is updated, the writer modifies the alternating rb cycle and moves around it. The way the system gets updated is determined by the type of the writer, which is determined by the forms of $s(t)_{|s(t)|-1}$ and $s(t)_0$ (which represent the kinds of the vertices to the writer's left and right, respectively). In our representation, using $s(t)$, an update corresponds to making $s(t + 1)$ by taking $s(t)$, altering the elements at the extreme left or right hand side (since these are the parts adjacent to the writer) and also rotating the word (to represent the writers movement).

$s(0) = AA$

$s(1) = ABA$

$s(2) = BABA$

$s(3) = ABAAB$

To see the way $s(t)$ evolves with time, note that $\forall t \geq 0$ we have:

1. If $s(t) = AxA$, for some $x \in \{A, B\}^*$ then $\rho(X^t)_{G^t} = 0$ and $s(t+1) = xABA$.

2. If $s(t) = AxB$, for some $x \in \{A, B\}^*$ then $\rho(X^t)_{G^t} = r$ and $s(t+1) = BAx$.

3. If $s(t) = BxA$, for some $x \in \{A, B\}^*$ then $\rho(X^t)_{G^t} = b$ and $s(t+1) = ABAx$.

These three statements are sufficient to describe the evolution of the system. They correspond to the rewrite rules shown at the top right of Figure 26. For a word $x = x_0 x_1 ... x_{m-1}$ let $\text{last}(x) = x_{m-1}$ denote the last element of x. The key to our proof is to establish a correspondence between the dynamics of $s(t)$ and the dynamics of the binary tag system $K(T)$. Here $K(T)$ denotes the word obtained by updating $K(0) = 0$, a number of times equal to T, under the tag system with rules $0 \to 01$ and $1 \to 011$. More precisely, $\forall T \geq 0$ we have:

1. If there exists a word $x \in \{0, 1\}^*$ such that $K(T) = x0$ then $K(T+1) = 01x$.

2. If there exists a word $x \in \{0, 1\}^*$ such that $K(T) = x1$ then $K(T+1) = 011x$.

$K(0) = 0$
$K(1) = 01$
$K(2) = 0110$
$K(3) = 01011$

The correspondence between $K(T)$ and $s(t)$ involves the mapping $\mu : \{0, 1\}^* \mapsto \{A, B\}^*$ defined such that for each binary word $x \in \{0, 1\}^*$ we have that $\mu(x)$ is the word obtained by taking x and replacing each 0 with AB and each 1 with A. The correspondence is not trivial. When $\text{last}(K(T)) = 0$, the single step $K(T) \to K(T+1)$ corresponds to two steps, $s(t) \to s(t+1) \to s(t+2)$, in the s system. When $\text{last}(K(T)) = 1$, the single step $K(T) \to K(T+1)$ corresponds to four steps, $s(t) \to s(t+1) \to s(t+2) \to s(t+3) \to s(t+4)$, in the s system.

$\forall T \geq 0$ let us define the s-time $\gamma(T)$ corresponding with T as

$$\gamma(T) = 2 + 2.T + 2.|\{i \in \{0, .., T-1\} : \text{last}(K(i)) = 1\}|. \tag{5}$$

Here $s(\gamma(T))$ represents $K(T)$, as we describe with our next result.

Lemma 5 $\forall T \geq 0$ *the following equations hold:*

$$s(\gamma(T)) = B\mu(K(T))A \tag{6}$$

$$s(\gamma(T) + 1) = ABA\mu(K(T)). \tag{7}$$

Moreover, if $\mathrm{last}(K(T)) = 1$ *then the following two equations also hold:*

$$s(\gamma(T) + 2) = BA\mu(K(T))BA \tag{8}$$

$$s(\gamma(T) + 3) = ABAA\mu(K(T))B. \tag{9}$$

Proof For $T = 0$ we have $\gamma(T) = 2$, $K(T) = 0$, $\mathrm{last}(K(T)) = 0$, $\mu(K(T)) = AB$, $s(\gamma(T)) = BABA$ and $s(\gamma(T) + 1) = ABAAB$, and so our result hold in this case. Suppose the result holds for some $T \geq 0$. We will show that it holds for $T + 1$.

First let us consider the case where $\mathrm{last}(K(T)) = 0$. In this case there must exist some $x \in \{0, 1\}^*$ such that $K(T) = x0$, and so we have

$$
\begin{align}
s(\gamma(T) + 1) &= ABA\mu(K(T)) \tag{10} \\
&= ABA\mu(x)\mu(0) \tag{11} \\
&= ABA\mu(x)AB. \tag{12}
\end{align}
$$

Here, Eq. (10) follows from our assumption that the result holds for T. It follows (from the update rules of s) that $s(\gamma(T) + 2) = BABA\mu(x)A$. Also, in this case,

$$
\begin{align}
\gamma(T + 1) &= 2 + 2.(T + 1) + 2.|\{i \in \{0, .., T\} : \mathrm{last}(K(i)) = 1\}| \tag{13} \\
&= \gamma(T) + 2. \tag{14}
\end{align}
$$

Moreover, $K(T + 1) = 01x$, and so we have

$$
\begin{align}
B\mu(K(T + 1))A &= BABA\mu(x)A \tag{15} \\
&= s(\gamma(T + 1)), \tag{16}
\end{align}
$$

(as we require). It follows (from the update rules of s) that

$$s(\gamma(T + 1) + 1) = ABA\mu(K(T + 1)) \tag{17}$$

(as we require). Moreover, if $\text{last}(K(T+1)) = 1$ then there exists a binary word $y \in \{0,1\}^*$ such that $K(T+1) = y1$. Now

$$
\begin{align}
s(\gamma(T+1)+1) &= ABA\mu(y)\mu(1) \tag{18} \\
&= ABA\mu(y)A. \tag{19}
\end{align}
$$

Now it follows (from the update rules of s) that

$$
\begin{align}
s(\gamma(T+1)+2) &= BA\mu(y)ABA \tag{20} \\
&= BA\mu(y1)BA \tag{21} \\
&= BA\mu(K(T+1))BA, \tag{22}
\end{align}
$$

(as we require). Now it follows (from the update rules of s) that

$$
s(\gamma(T+1)+3) = ABAA\mu(K(T+1))B \tag{23}
$$

(as we require). So now we have shown that the result holds for $T+1$ in this case (where $\text{last}(K(T)) = 0$).

Now let us consider the other case, where $\text{last}(K(T)) = 1$. In this case there must exist some $x \in \{0,1\}^*$ such that $K(T) = x1$. Our assumption that the result holds for T gives us $s(\gamma(T)+3) = ABAA\mu(K(T))B$. It follows (from the update rules of s) that

$$
\begin{align}
s(\gamma(T)+4) &= BABAA\mu(K(T)) \tag{24} \\
&= BABAA\mu(x)\mu(1) \tag{25} \\
&= BABAA\mu(x)A. \tag{26}
\end{align}
$$

In this case,

$$
\begin{align}
\gamma(T+1) &= 2 + 2.(T+1) + 2.|\{i \in \{0,..,T\} : \text{last}(K(i)) = 1\}| \tag{27} \\
&= BABAA\mu(x)\mu(1) \tag{28} \\
&= \gamma(T) + 4. \tag{29}
\end{align}
$$

Hence $s(\gamma(T+1)) = s(\gamma(T)+4) = BABAA\mu(x)A$. Moreover, $K(T+1) = 011x$,

and so we have

$$B\mu(K(T+1))A \ = \ BABAA\mu(x)A \tag{30}$$

$$= \ BABAA\mu(x)\mu(1) \tag{31}$$

$$= \ s(\gamma(T+1)), \tag{32}$$

(as we require). Now we can establish that

$$s(\gamma(T+1)+1) = ABA\mu(K(T+1)), \tag{33}$$

in exactly the same way we showed Eq. (17). In addition, if $\text{last}(K(T+1)) = 1$ then there exists a binary word $y \in \{0,1\}^*$ such that $K(T+1) = y1$, and we can establish that

$$s(\gamma(T+1)+2) = BA\mu(K(T+1))BA \tag{34}$$

and

$$s(\gamma(T+1)+3) = ABAA\mu(K(T+1))B, \tag{35}$$

in exactly the same way as we showed Eq. (22) and Eq. (23), respectively. So now we have shown that the result holds for $T + 1$ in this case (where $\text{last}(K(T)) = 1$). So now we have shown that, if the result holds for T then it holds for $T + 1$. Now since the result holds when $T = 0$ we can use induction with T to prove the result holds $\forall T \geq 0$.

We can use this lemma to form the following corollary, that describes the sequence $\rho(X^0)_{G^0}\rho(X^1)_{G^1}...$ in terms of the forms of $K(T) : T \geq 0$.

Corollary 6 $\forall t \geq 0$ *we have the following:*

1. *If $t > 0$ and t is even then $\rho(X^t)_{G^t} = b$.*

2. *If $t \in \{0,1\}$ or if t is of the form*

$$3 + 2.T + 2.|\{i \in \{0,..,T-1\} : \text{last}(K(i)) = 1\}|,$$

for some $T \geq 0$, that is such that $\text{last}(K(T)) = 1$ then $\rho(X^t)_{G^t} = 0$.

3. *Otherwise[6] we have $\rho(X^t)_{G^t} = r$.*

---

[6]i.e., when $t > 1$, t is odd, and there does not exist any $T \geq 0$ such that $3 + 2.T + 2.|\{i \in \{0,..,T-1\} : \text{last}(K(i)) = 1\}|$

Proof Clearly $\rho(X^0)_{G^0} = \rho(X^1)_{G^1} = 0$, so the result holds $\forall t \in \{0,1\}$. For $t \geq 2$, let T^* be the maximum integer such that $\gamma(T^*) \leq t$. We must have $T^* \geq 0$ since $\gamma(0) = 2 \leq t$. Also, we must have $\gamma(T^*) \geq t - 3$, since otherwise (i.e., if $\gamma(T^*) < t-3$) we would have $\gamma(T^* + 1) \leq \gamma(T^*) + 4 \leq t$ which would contradict our assumption that T^* is maximum. Now we know $\gamma(T^*) \in \{t-3, t-2, t-1, t\}$ let us consider the different cases.

If $\gamma(T^*) = t$ then t is even and Lemma 5 gives us that $s(t) = B\mu(K(T^*))A$ and so $\rho(X^t)_{G^t} = b$.

If $\gamma(T^*) = t - 1$ then $t = \gamma(T^*) + 1$ is odd and Lemma 5 gives us that $s(t) = s(\gamma(T^*)+1) = ABA\mu(K(T^*))$ and so $\mathrm{last}(K(T^*)) = 0$ implies $\rho(G^t)_{X^t} = r$ and $\mathrm{last}(K(T^*)) = 1$ implies $\rho(G^t)_{X^t} = 0$.

If $\gamma(T^*) = t - 2$ then $t = \gamma(T^*) + 2$ is even and Lemma 5 gives us that $s(t) = s(\psi(T^*) + 2) = BA\mu(K(T^*))BA$ and so $\rho(G^t)_{X^t} = b$.

If $\gamma(T^*) = t - 3$ then $t = \gamma(T^*) + 3$ is odd and Lemma 5 gives us that $s(t) = s(\psi(T^*) + 3) = ABAA\mu(K(T^*))B$ and so $\rho(G^t)_{X^t} = r$.

These are all the possibilities. So we have $\forall t \geq 2$ that, $\rho(G^t)_{X^t} = b$ if and only if t is even. When $t \geq 2$ is odd, we must have $\rho(G^t)_{X^t} \in \{0, r\}$.

Moreover, $\rho(G^t)_{X^t} = 0$ if and only if $t = \gamma(T^*) + 1$ where $\mathrm{last}(K(T^*)) = 1$. In other words, $\rho(G^t)_{X^t} = 0$ if and only if there exists some $T \geq 0$ such that $t = 1 + \gamma(T) = 3 + 2.T + 2.|\{i \in \{0,..,T-1\} : \mathrm{last}(K(i)) = 1\}|$ and $\mathrm{last}(K(T)) = 1$. Also, if $\rho(G^t)_{X^t} \notin \{b, 0\}$ then $\rho(G^t)_{X^t} = r$.

Now we have established Corollary 6, we can gain a description of the sequence $\rho(X^0)_{G^0}\rho(X^1)_{G^1}...$ by getting a formula for $\mathrm{last}(K(T))$. Our tag system $K(T)$ evolves like a slow version of the neighbor independent substitution system $Z(n)$, where every element of the word is updated every time step under rules $0 \to 01$, $1 \to 011$.

More precisely, let us define the mapping $\Omega : \{0,1\}^* \mapsto \{0,1\}^*$ such that $\forall x \in \{0,1\}^*$ we have that $\Omega(x)$ is the word obtained by taking x and replacing each 0 with 01 and replacing each 1 with 011.

Now let $Z(0) = 0$ and $\forall n \geq 0$ let $Z(n+1) = \Omega(Z(n))$

$Z(0) = 0$

$Z(1) = 01$

$Z(2) = 01011$

$Z(3) = 0101101011011.$

For a word, $x = x_0 x_1 ... x_{m-1}$ let $\mathrm{rev}(x) = x_{m-1} x_{m-2} ... x_0$ denote the reversal of the order of characters in the word.

Lemma 7 *The infinite word* $\mathrm{last}(K(0))\,\mathrm{last}(K(1))...$ *is equal to the infinite word* $\mathrm{rev}(Z(0))\,\mathrm{rev}(Z(1))...$

Proof For a word $K(T) = K(T)_0 K(T)_1 ... K(T)_{|K(T)|-1}$ we have

$$K(T+1) = \Omega(K(T)_{|K(T)|-1})K(T)_0...K(T)_{|K(T)|-2} \tag{36}$$

and

$$K(T+2) = \Omega(K(T)_{|K(T)|-2})\Omega(K(T)_{|K(T)|-1})K(T)_0...K(T)_{|K(T)|-3}. \tag{37}$$

Following this pattern, one can see that

$$\mathrm{last}(K(T))\,\mathrm{last}(K(T+1))...\,\mathrm{last}(K(T+|K(T)|))$$

is equal to

$$K(T)_{|K(T)|-1}K(T)_{|K(T)|-2}...K(T)_0 = \mathrm{rev}(K(T)).$$

Moreover, after we have performed $|K(T)|$ updates on $K(T)$, we will get

$$K(T+|K(T)|) = \Omega(K(T)_0)\Omega(K(T)_1)...\Omega(K(T)_{|K(T)|-}) = \Omega(K(T)). \tag{38}$$

From this one can see that $K(T)$ evolves like a slow version of the substitution system $Z(n)$. When $K(T) = Z(n)$ (as is the case when $T = n = 0$) we will have

$$K(T+|K(T)|) = \Omega(K(T)) = \Omega(Z(n)) = Z(n+1) \tag{39}$$

and

$$\mathrm{last}(K(T))\,\mathrm{last}(K(T+1))...\,\mathrm{last}(K(T+|K(T)|)) = \mathrm{rev}(Z(n)). \tag{40}$$

Our result can then be proved by induction.

Lemma 8 $\forall n \geq 1$ *we have* $Z(n+1) = Z(n)Z(n)Z(n-1)...Z(1)1$

Proof $Z(2) = 01011 = Z(1)Z(1)1$ so the result holds for $n = 1$. Suppose that

the result holds for some $n \geq 1$, now

$$
\begin{align}
Z(n+2) &= \Omega(Z(n+1)) \tag{41} \\
&= \Omega(Z(n)Z(n)Z(n-1)...Z(1)1) \tag{42} \\
&= Z(n+1)Z(n+1)Z(n)...Z(2)011 \tag{43} \\
&= Z(n+1)Z(n+1)Z(n)...Z(2)Z(1)1, \tag{44}
\end{align}
$$

and so the the result holds for $n+1$. The result is proved, in general, by induction with n.

For a word $x = x_0 x_1...x_{m-2}x_{m-1} \in \{0,1\}^*$ let $\operatorname{tog}(x) = (1-x_0)x_1...x_{m-2}(1-x_{m-1})$ be the word obtained by altering the characters at x's extreme left and right hand sides.

Let $L(x) = x_1 x_2...x_{m-1}$ denote the word obtained by deleting the left most element from x.

Lemma 9 $\forall m \geq 1$, for any sequence of words $a^1, a^2..., a^m \in \{01, 011\}$ we have $\operatorname{tog}(a^1) \operatorname{tog}(a^2)... \operatorname{tog}(a^m) = L(a^1)a^2...a^m 0$.

Proof This clearly holds when $m = 1$ since $\operatorname{tog}(01) = 10 = L(01)0$ and $\operatorname{tog}(011) = 110 = L(011)0$. Suppose that the result holds for $m \geq 1$. Now $\operatorname{tog}(a^1) \operatorname{tog}(a^2)... \operatorname{tog}(a^m) \operatorname{tog}(a^{m+1})$ is equal to $L(a^1)a^2...a^m 0 \operatorname{tog}(a^{m+1})$, which is equal to $L(a^1)a^2...a^m 0 L(a^{m+1})0$, which is equal to $L(a^1)a^2...a^m a^{m+1}0$ as required. So the result holds for $m+1$. By induction with m the result is proved in general.

Lemma 10 $\forall n \geq 1$ we have $\operatorname{tog}(Z(n)) = \operatorname{rev}(Z(n))$, $Z(n)_0 = 0$ and $\operatorname{last}(Z(n)) = 1$.

Proof $\operatorname{tog}(Z(1)) = 10 = (Z(1))$, so our result holds when $n = 1$. Suppose the result holds for some $n \geq 1$. We write $Z(n) = Z(n)_0...Z_{|Z(n)|-1}$, now

$$
\begin{align}
\operatorname{tog}(Z(n+1)) &= \operatorname{tog}(\Omega(Z(n)_0)...\Omega(Z(n)_{|Z(n)|-1})) \tag{45} \\
&= \operatorname{tog}(01\Omega(Z(n)_1)...\Omega(Z(n)_{|Z(n)|-2})011) \tag{46} \\
&= 11\Omega(Z(n)_1)...\Omega(Z(n)_{|Z(n)|-2})010, \tag{47}
\end{align}
$$

also,

$$
\begin{aligned}
\text{rev}(Z(n+1)) &= \text{rev}(\Omega(Z(n)_{|Z(n)|-1}))...\text{rev}(\Omega(Z(n)_0)) &(48)\\
&= 110\,\text{rev}(\Omega(Z(n)_{|Z(n)|-2}))...\text{rev}(\Omega(Z(n)_1))10 &(49)\\
&= 110\,\text{rev}(\Omega(Z(n)_1))...\text{rev}(\Omega(Z(n)_{|Z(n)|-2}))10 &(50)\\
&= 110\,\text{tog}(\Omega(Z(n)_1))...\text{tog}(\Omega(Z(n)_{|Z(n)|-2}))10 &(51)\\
&= 110L(\Omega(Z(n)_1))\Omega(Z(n)_2)...\Omega(Z(n)_{|Z(n)|-2})010 &(52)\\
&= 11\Omega(Z(n)_1)\Omega(Z(n)_2...\Omega(Z(n)_{|Z(n)|-2})010 &(53)
\end{aligned}
$$

The reason Eq. (49) implies Eq. (50) is that $\text{rev}(Z(n)) = \text{tog}(Z(n))$ implies

$$Z(n)_1...Z(n)_{|Z(n)|-2} = Z(n)_{|Z(n)|-2}...Z(n)_1$$

is palindromic, which implies

$$\text{rev}(\Omega(Z(n)_{|Z(n)|-2}))...\text{rev}(\Omega(Z(n)_1)) = \text{rev}(\Omega(Z(n)_1))...\text{rev}(\Omega(Z(n)_{|Z(n)|-2})).$$

The reason Eq. (50) implies Eq. (51) is that we always have $\Omega(Z(n)_i) \in \{01, 011\}$, and so $\text{rev}(\Omega(Z(n)_i)) = \text{tog}(\Omega(Z(n)_i))$. We can use Lemma 9 to derive Eq. (52) from Eq. (51), and Eq. (53) follows because the left most entry in $\Omega(Z(n)_1)$ is 0.

Now, since Eq. (53) is identical to Eq. (47), we have proved that our result holds for $n+1$. The fact that the result holds $\forall n \geq 1$ then follows by induction.

Now we can state our critical result. Recall that $\text{last}(K(0))\,\text{last}(K(1))...$ is the infinite sequence consisting of the last elements of the words $K(0), K(1)...$ (evolving under our tag system).

Lemma 11 $\text{last}(K(0))\,\text{last}(K(1))... = \lim_{n\to\infty} Z(n)$

Proof Lemma 8 gives us that $\forall n > 0$,

$$Z(n+1) = Z(n)Z(n)Z(n-1)...Z(1)1, \tag{54}$$

and so

$$\text{rev}(Z(n+1)) = 1\,\text{rev}(Z(1))\,\text{rev}(Z(2))...\text{rev}(Z(n))\,\text{rev}(Z(n)), \tag{55}$$

Now, Lemma 10 implies $\text{rev}(Z(k)) = \text{tog}(Z(k))$, $\forall k > 0$. So now we have

$$
\begin{align}
Z(n+1) &= \text{rev}(\text{rev}(Z(n+1))) \tag{56} \\
&= \text{tog}(\text{rev}(Z(n+1))) \tag{57} \\
&= 0\,\text{rev}(Z(1))\,\text{rev}(Z(2))...\,\text{rev}(Z(n))R(\text{rev}(Z(n)))1, \tag{58}
\end{align}
$$

where $R(x)$ denotes the word obtained by deleting the right most element from x. Now, taking the limit as $n \to \infty$ we have

$$
\begin{align}
\lim_{n \to \infty}(Z(n)) &= \lim_{n \to \infty}(Z(n+1)) \tag{59} \\
&= 0\,\text{rev}(Z(1))\,\text{rev}(Z(2))... \tag{60} \\
&= \text{rev}(Z(0))\,\text{rev}(Z(1))\,\text{rev}(Z(2))... \tag{61} \\
&= \text{last}(K(0))\,\text{last}(K(1))\,\text{last}(K(2))... \tag{62}
\end{align}
$$

Here Lemma 7 gives us that Eq. (61) implies Eq. (62).

The above result (Lemma 11) is important because a formula is known [20] for the infinite sequence of $\lim_{n \to \infty}(Z(n))$ (which we have just shown to be equal to $\text{last}(K(0))\,\text{last}(K(1))...$). Using this formula we have $\forall i \geq 0$, that

$$
\text{last}(K(i)) = \left(\lim_{n \to \infty}(Z(n)) \right)_i = \lfloor (i+1)\phi \rfloor - \lfloor i.\phi \rfloor - 1, \tag{63}
$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Other results are also known for this sequence. For $T \geq 0$ the index of the Tth 1 in the sequence $\lim_{n \to \infty}(Z(n))$ is well known [21] to be given by the formula $\lfloor (T+1)\phi \rfloor$. In other words,

$$
\left\{ T \geq 0 : \left(\lim_{n \to \infty}(Z(n)) \right)_T = 1 \right\} = \{ T \geq 0 : \text{last}(K(T)) = 1 \} = \{ \lfloor (n+1)\phi \rfloor : n \geq 0 \} \tag{64}
$$

Suppose $h \geq 2$. Now, according to Corollary 6, $\rho(X^h)_{G^h} = 0$ if and only if $\exists T \geq 0$ such that $h = 3 + 2T + 2|\{i \in \{0, 1, ..., T-1\} : \text{last}(K(i)) = 1\}|$ and $\text{last}(k(T)) = 1$.

Now since $T \geq 0, \text{last}(k(T)) = 1$ if and only if $\exists n \geq 0 : \lfloor (n+1)\phi \rfloor = T$ we have the following:

47

$\rho(X^h)_{G^h} = 0$ *if and only if* $\exists n \geq 0$ *such that* $h = 3 + 2\lfloor(n+1)\phi\rfloor + 2.\Theta$, *where:*

$$\Theta = |\{i \in \{0, 1, ..., \lfloor(n+1)\phi\rfloor - 1\} : \text{last}(K(i)) = 1\}| \qquad (65)$$

$$= |(\{0, 1, ..., \lfloor(n+1)\phi\rfloor - 1\} \cap \{\lfloor(0+1)\phi\rfloor, \lfloor(1+1)\phi\rfloor, ...\})| \qquad (66)$$

$$= |\{\lfloor(0+1)\phi\rfloor, \lfloor(1+1)\phi\rfloor, ..., \lfloor((n-1)+1)\phi\rfloor\}| \qquad (67)$$

$$= n. \qquad (68)$$

Hence we have

$$\{h \geq 2 : \rho(X^h)_{G^h} = 0\} = \{3 + 2\lfloor(n+1)\phi\rfloor + 2.n : n \geq 0\}. \qquad (69)$$

Now, the number of vertices $|G^t|$, in G^t will be equal to

$$|G^t| = 8 + 2|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = b\}| + 2|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = 0\}| \qquad (70)$$

because the number of vertices only changes when the writer is of type 0 or b, and it increases by 2 each such time. Now Corollary 6 implies

$$|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = b\}| = \left\lfloor\frac{t+1}{2}\right\rfloor - 1 + \delta_{t,0} \qquad (71)$$

(where $\delta_{i,j}$ is the Kronecker Delta).

Also, $\forall t > 5$, we have:

$$|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = 0\}| = 2 + |\{i \in \{2, 3..., t-1\} : \rho(X^i)_{G^i} = 0\}| \qquad (72)$$

$$|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = 0\}| = 2 + |\{n \geq 0 : 3 + 2\lfloor(n+1)\phi\rfloor + 2.n < t\}|, \qquad (73)$$

Where we use use Eq. (69) to infer Eq. (73) from Eq. (72). Now $\forall n \geq 0 \; \forall t > 5$ the following four inequalities are equivalent:

$$3 + 2\lfloor(n+1)\phi\rfloor + 2.n < t \qquad (74)$$

$$\lfloor(n+1)\phi\rfloor < \left\lfloor\frac{t}{2}\right\rfloor - (n+1) \qquad (75)$$

$$(n+1)\phi^2 < \left\lfloor \frac{t}{2} \right\rfloor \tag{76}$$

$$n < \left\lceil \left\lfloor \frac{t}{2} \right\rfloor \frac{1}{\phi^2} \right\rceil - 1 \tag{77}$$

The equivalence of inequalities (74) and (75) can be derived using the fact that $2a < b \Leftrightarrow a < \left\lfloor \frac{b+1}{2} \right\rfloor$, for integers a and b. The equivalence of inequalities (75) and (76) can be derived using the fact that $\lfloor r \rfloor < m \Leftrightarrow r < m$ for a real r and an integer m, and the fact that $1 + \phi = \phi^2$.

It follows that when $t > 5$ we can rewrite Eq. (73) as

$$
\begin{aligned}
|\{i \in \{0, ..., t-1\} : \rho(X^i)_{G^i} = 0\}| &= 2 + \left|\left\{ n \geq 0 : n < \left\lceil \left\lfloor \frac{t}{2} \right\rfloor \frac{1}{\phi^2} \right\rceil - 1 \right\}\right| \tag{78} \\
&= 2 + \left|\left\{ 0, 1, ..., \left\lceil \left\lfloor \frac{t}{2} \right\rfloor \frac{1}{\phi^2} \right\rceil - 2 \right\}\right| \tag{79} \\
&= 2 + \left\lceil \left\lfloor \frac{t}{2} \right\rfloor \frac{1}{\phi^2} \right\rceil - 1 \tag{80}
\end{aligned}
$$

Now by substituting Eq. (71) and Eq. (80) into Eq. (70), we get that when $t > 5$, the number of vertices in G_t is

$$|G^t| = 8 + 2\left(\left\lfloor \frac{t+1}{2} \right\rfloor - 1 \right) + 2\left(2 + \left\lceil \left\lfloor \frac{t}{2} \right\rfloor \frac{1}{\phi^2} \right\rceil - 1 \right), \tag{81}$$

as required. It can also be checked that Eq. (81) gives the correct answer for each $t \in \{0, 1, 2, 3, 4, 5\}$. Q.E.D.

References

[1] Wolfram, S., *Statistical Mechanics of Cellular Automata. Reviews of Modern Physics* **55** (1983), pp. 601–644.

[2] Prusinkiewicz, P. and Lindenmayer, A., *Algorithmic Beauty of Plants*, Springer-Verlag (1990).

[3] Alligood, K., Sauer, T. and Yorke, J., *Chaos: an introduction to dynamical systems*, Springer-Verlag (1997).

[4] Watts, D. and Strogatz, S., *Collective dynamics of 'small-world' networks*, *Nature* **393** (1998), pp. 440–442.

[5] Barabsi, A. and Albert, R., *Emergence of scaling in random networks,* *Science* **286** (1999), pp. 509–512.

[6] Gottlieb, E. and Shelton, K., *Color-induced subgraphs of Grunbaum colorings of triangulations of the sphere. Australasian Journal of Combinatorics* **30** (2004), pp. 183–192.

[7] Turing, A., *On Computable Numbers, with an Application to the Entscheidungs problem, Proceedings of the London Mathematical Society* **242** (1937), pp. 230–265.

[8] Comellas, F., *Complex Networks: Deterministic Models, NATO Security through Science Series - D: Information and Communication Security* **7** (2006), pp. 275–293.

[9] Southwell, R. and Cannings, C., *Games on graphs that grow deterministically, Gamenets Conference* (2009), pp. 347–356

[10] Southwell, R. and Cannings, C., *Some Models of Reproducing Graphs: I Pure Reproduction, Applied Mathematics* **1** (2010) pp. 137–145.

[11] Southwell, R. and Cannings, C., *Some Models of Reproducing Graphs: II Age Capped Vertices, Applied Mathematics* **1** (2010) pp. 251–259.

[12] Southwell, R. and Cannings, C., *Some Models of Reproducing Graphs: III Game Based Reproduction, Applied Mathematics* **1** (2010) pp. 335–343.

[13] Tomita K,., Kurokawa, H. and Murata, S, *Graph automata: natural expression of self-reproduction, Physica D* **171** (2002), pp. 197-210.

[14] Wolfram, S., *A New Kind of Science* (Wolfram Media, Inc., 2002).

[15] Fredkin, E., *Five big questions with pretty simple answers, IBM Journal Res. and Dev.* **48** (2004), pp. 31–45.

[16] Bolognesi, T., *Causal Sets from Simple Models of Computation, IJUC* **6** (2010), pp. 489–524.

[17] Zenil, H., *Randomness Through Computation*, (World Scientific, 2011).

[18] Lamb, A., *Dense Graphs, Node Sets, and Riders: Toward a Foundation for Particle Physics without Continuum Mathematics, Complex Systems* **19** (2010), pp. 115–130.

[19] Bolognesi, T., *Planar Trinet Dynamics with Two Rewrite Rules, Complex Systems* **18** (2008), pp. 1-41.

[20] *The On-Line Encyclopedia of Integer Sequences. A096270. Fixed point of the morphism* $0 \rightarrow 01$ *and* $1 \rightarrow 011$ (2012), https://oeis.org/A096270

[21] *The On-Line Encyclopedia of Integer Sequences. A000201. Lower Wythoff sequence (a Beatty sequence)* (2012), https://oeis.org/A000201

[22] Jonckheere, E., Lohsoonthorn, P. and Bonahon, F., *Scaled Gromov hyperbolic graphs, Journal of Graph Theory* **57** (2008), pp. 157–180.

[23] Narayan, O. and Saniee, I., *Large-scale curvature of networks, Phys. Rev. E* **84** (2011), pp. 1–8.

[24] Armsen, M., *A variational proof of the Gauss-Bonnet formula, Manuscripta Mathematica* **20** (1977), pp. 245-253.

[25] Wikispaces (2012), goldenmonkeyclan, http://goldenmonkeyclan.wikispaces.com/Fungi

[26] Website (2012), http://www.acanthusstudios.com/catalog/i39.html

[27] Website (2012), http://www.butterflypictures.net/4-caterpillar-pictures.html

[28] Website (2012), http://www.etsy.com/listing/88989950/10-telescopium-seashell-slices-2-275-in

[29] Website (2012), http://fishprint.wordpress.com/2008/02/23/slices/

[30] Website (2012), http://en.wikipedia.org/wiki/Necklace