Graph Self-Replication System

D Venkata Lakshmi★
Jeganathan L†
School of Computing Science and Engineering
Vellore Institute of Technology
Vandalur-Kelambakkam Road, Chennai, India-600 127
★*venkata.lakshmi2011@vit.ac.in*, †*jeganathan.l@vit.ac.in*

The self-replication introduced by John von Neumann is a process that produces a copy of itself. As a novel approach, this paper studies the self-replication process through the process of reproduction. In this paper, we propose a comprehensive graph reproduction system (GRS) and identify a specific reproduction system that turns out to be a graph self-replication system (GSS), with which a copy of any given graph can be produced through an algorithmic process. Unlike the GRS studied by Richard Southwell, our model considers the evolution of edges along with the evolution of vertices. We analyze some of the existing reproduction models through our system and identify the models that are self-replicable.

Keywords: self-replication; reproduction models; graph reproduction system; graph self-replication system; self-replication of graph; inherent graph self-replication system

1. Introduction

The question of whether a machine can make a copy of itself, called the problem of self-replication, was initiated by von Neumann [1] in 1940. For more than 60 years, the focus of investigation was primarily on understanding the logic necessary for replication in comparison with the biological process of replication, necessary conditions that are to be satisfied by a replication process and the fundamental algorithms involved in self-replication.

This study of the artificial self-replication process will establish the much-sought mapping between the biological evolution process and the computational evolution process, finally showing the continuum between living things and nonliving things. The realization of artificial self-replicating machines can have diverse applications, ranging from the fabrication of nanomachines to space exploration. Further, the concept of self-replication is used to model various complex biological and physical systems such as robotics, formation of snowflakes, and so on.

https://doi.org/10.25088/ComplexSystems.28.3.313

Based on the models used to study self-replication, the explorations carried out so far can be classified into four categories as described by Sipper in [2]: cellular automata–based self-replication, computer program–based self-replication, string-based self-replication and mechanical model–based self-replication. A formal framework to investigate self-replication was provided by the cellular automaton (CA), a dynamical system, discrete in time and space, proposed by Ulam and von Neumann. A cellular automaton consists of an array of cells, each of which can be in one of a finite number of possible states, updated synchronously in discrete time steps, according to a local, identical interaction rule. The state of a cell at the next time step is determined by the current states of a surrounding neighborhood of cells.

Von Neumann designed [3] the universal self-replicator, a two-dimensional CA with 29 states per cell with a five-cell neighborhood that implements self-replication as well as construction universality. Construction universality refers to the ability to construct all the structures in a given specific product set, given its description. The first hardware implementation was attempted by Beuchat and Haenni [4], wherein small systems were realized. Beuchat and Haenni could not realize the complete structure due to the requirement of a larger number of cells. While universal construction may represent a sufficient condition for attaining self-replication, Langton observed that it is not a necessary condition. Langton attempted to design the simplest CA capable of self-reproduction.

Langton's loop [5] uses eight states for each of the 86 nonquiescent cells making up its initial configuration, a five-cell neighborhood and a few hundred transition rules. Codd [6] developed a self-replicating structure consisting of two-dimensional, eight-state, five-neighbor cellular automata (CAs). Further simplifications to Langton's automaton were introduced by Reggia in his work. Apart from the CA models, Lohn and Reggia [7–9] used genetic algorithms to generate the rules that result in self-replicating structures. Sipper discusses a complete literature survey on self-replication in his two papers [2, 10].

Bratley and Millo [11] and Burger, Brill and Machi [12] devised self-replicating computer programs, wherein a computer program produces a copy of the program. Ray [13, 14] investigated the self-replicating programs to study evolution using his Tierra system, a virtual world consisting of computer programs that can undergo evolution. So far, in the CA-based models as well as in the computer program–based models, the items to be self-replicated are essentially predetermined directly or evolve through an artificial process. There are self-replicating systems where the objects to be replicated are dynamically constructed. Such a self-replicated system was studied by Laing [15], wherein string-like objects, such as interconnected chains, are the basic component of the structure.

A study of any self-replicating machine will have two aspects, the logical component and the mechanical part (how the machine is built). The first three models discussed so far focus on the logic required to build a self-replicating machine. In one sense, the first three models bring out the computational aspects of self-replicating machines. The fourth model describes the mechanical aspect of constructing the self-replicating machine. Penrose [16] built simple mechanical units, an ensemble of which was placed in a box with all the required physical components. The box was subjected to some physical activity so as to produce the required energy for the physical components to act to produce a copy of the ensemble unit. Following this mechanical model, Rebek Jr. [17] designed a process to produce self-replicating molecules. In [18], Ichihashi and Yomo gave a constructive approach, in which life-specific functions are recreated in a test tube from specific biological molecules. Using this approach, they were able to employ design principles to reproduce life-specific functions, and the knowledge gained through the reproduction process provides clues as to their origins.

In recent years, Southwell and Cannings generated different graph-reproducing models [19], which produce different types of graphs from the given initial graph. These models are used to describe the growth of interactions between individuals within a population. In their models [20, 21], every vertex produces a new vertex that is connected to the existing vertices based on different constraints, thereby generating eight different models of the graph-generating mechanism. In their models, edges get eliminated because of the death of vertices. Their models do not consider the edges as a valid parameter for reproduction. Since graphs are characterized by both vertices and edges, we observe that any graph reproduction system (GRS) should also consider the edges as well as the vertices of the graph. From the given graph, the Southwell model focuses on the generation of different graphs, and self-replication of the given graph is not their concern.

In this paper, we propose a generic model for a GRS that is self-replicable. In contrast to the Southwell model, our model considers vertices as well as edges as valid parameters for the evolution of graphs. All the models developed by Southwell [19] are just the specific instances of our model. This paper identifies a specific GRS that is self-replicable. Further, we analyze some of the existing reproduction models through our system and identify the models that are self-replicable. We consider self-replication as an algorithmic process to investigate the graphs that are self-replicable. Our graph self-replicating model through a GRS will fit into an altogether different approach that does not fall into any of the four categories described by Sipper in [2].

In this paper, we abstract the self-replication concept using graphs. The paper is organized as follows. Section 2 proposes a generic model of GRS $\boldsymbol{\rho}$. Section 3 discusses the graph self-replication system (GSS) along with illustrations. Section 4 discusses a rigorous mathematical proof for the self-replicability of the GRS $\boldsymbol{\rho}$ along with the results obtained through simulation. Section 5 investigates some of the existing graph reproduction models for self-replicability. The last section concludes with the description of future enhancements for our model.

2. Graph Reproduction System

We present a GRS that has the ability to produce copies of the graph given as an input to the system. Though biological organisms are the most familiar examples of a self-replicating system, this paper explores the self-replication system with graphs as the basic unit. This artificial GSS is motivated by the desire to understand the fundamental information processing principles and algorithms involved in self-replication, independent of their physical realization. A better theoretical understanding of the GRS could be useful in a number of ways, from a computational as well as an engineering perspective.

As mentioned earlier, the main aim is to study the feasibility of any graph-generating system becoming a self-replication system. For this purpose, we should have a generating mechanism through which graphs get generated from the given graph through some well-defined process.

In that sense, we consider a GRS that generates graphs through the process of evolution/reproduction over discrete time steps t. This paper assumes asexual reproduction with a single parent, in the sense that offspring are born with the parent's strategy (potential) and link up to the surroundings in a similar way to their parent. This will simulate the natural process of children inheriting the parent's genes and getting connected to the environment of their parents.

In our view, any asexual reproduction system in the social environment should not omit the following items.

- Offspring are born based on the parent's strategy (potential) and get connected to the environment of their parent.
- Individual organisms lose their reproductive potential over a period of time due to various reasons. That is, individual organisms may become infertile.
- Infertile organisms may gain fertility due to some medical treatment.
- Organisms lose their connectivity with other organisms over a period of time.
- Individual organisms die due to the aging process.

- As a social constraint, organisms might not produce more offspring, though they are capable. In other words, there may be a cap on the number of offspring that can be produced by an organism; for example, organisms are allowed to reproduce only once during their lifetime.

Considering all these items, we propose a graph-generating mechanism called a graph reproduction system (GRS), starting from an initial graph. Our main goal is to check the potential of this GRS in becoming a graph self-replication system (GSS), thereby possessing the ability to generate copies of the initial graph.

Definition 1. A GRS, $\boldsymbol{\rho} = (\text{RA, RI, CA, LCA, DA})$, where

- RA is the reproduction rule—the set of rules with which the vertices of a graph reproduce.
- RI is the reproductive index—a positive integer, the maximum number of offspring that can be produced by an organism throughout its life.
- CA is the connectivity rule—the set of rules with which the offspring are connected to their parent and the other members of the society.
- LCA is the loss of connectivity rule—the set of rules by which the connectivity between the different vertices of the graph is lost.
- DA is the dying rule—the set of rules by which the vertices die.

Definition 2. Language generated by a GRS: Let $\boldsymbol{\rho}$ be any GRS. Let G be any graph. $\boldsymbol{\rho}(G)$ is the set of graphs reproduced by G; that is, $\boldsymbol{\rho}(G) = U_{i=1}^{\infty} \boldsymbol{\rho}^i(G)$, where $\boldsymbol{\rho}^i(G)$ is the graph produced in the i^{th} generation.

Note 1. The initial graph G is not included in $\boldsymbol{\rho}(G)$, just to observe whether $\boldsymbol{\rho}(G)$ produces G or not.

Since the vertices and edges are the only two parameters of a graph, the preceding five-tuple GRS represents a comprehensive model of any GRS in the sense that the system includes the birth and death of vertices along with the birth and death of edges. Any graph-generating reproducing mechanism will be a specific instance of our GRS. The generating reproducing system will differ only in the description of rules.

3. Graph Self-Replication System

The preceding GRS $\boldsymbol{\rho}$ has five components; by describing each of the five components, we will have a specific GRS. Any component of $\boldsymbol{\rho}$ can be described in a number of ways; thereby, we will have an exponential number of GRS examples.

Definition 3. Graph self-replication system (GSS): A GRS $\boldsymbol{\rho}$ is said to be a GSS system of order k if there exists at least one graph G and a $k \geq 1$ such that $\boldsymbol{\rho}^k(G) \cap \{G\} \neq \Phi$. If for any k, $\boldsymbol{\rho}^k(G)$ has disconnected components, the individual components of $\boldsymbol{\rho}^k(G)$ will be considered as separate graphs for the computation of $\boldsymbol{\rho}^k(G) \cap \{G\}$.

Definition 4. Inherent GSS: A GRS $\boldsymbol{\rho}$ is said to be an inherent GSS if $\boldsymbol{\rho}^k(G) \cap \{G\} \neq \Phi$ for any graph G with a nonempty set of vertices and a $k \geq 1$.

As already mentioned, the graph reproduction models of Southwell [22] discuss primarily the generation of different graphs, whereas the main focus of our GRS is to check self-replicability of the reproduction system, in the sense of producing copies of the initial graph. For all the rules like RA, CA, LCA and DA, we have given the description without attaching any specific physical meaning to that. We can think about the exhaustive list of descriptions for each of the rules, which will bring out many comparative models. Every GRS $\boldsymbol{\rho}$ will differ in the description of the different rules. Now, we describe each rule in a specific way, to propose a GRS.

3.1 Graph Self-Replication System

Now we describe a specific GRS that is self-replicable, where each component is described one by one as follows.

I: Reproduction rule (RA^1): In the sense of reproduction, we categorize the vertices of a graph as: reproduction-capable vertices (RC), nonreproduction-capable vertices (NRC) and newborn vertices (NB). Only RC vertices can produce offspring, one at a time. In nature, some living organisms lose fertility/gain fertility/retain fertility due to various reasons. In that sense, an RC vertex can become an NRC vertex based on the following conditions.

RC to NRC conversion rules:

1. Every RC vertex with either two or three RC neighbors will remain an RC vertex in the next generation.
2. Every NRC vertex that is adjacent to exactly three RC neighbors will become an RC vertex in the next generation.
3. Every RC vertex with four or more RC neighbors will become an NRC vertex in the next generation.
4. Every RC vertex with one or fewer RC neighbors will become an NRC vertex in the next generation.

Every vertex in the graph is labeled with a three-tuple (name of the node, nature of the node RC/NRC/NB, index of the generation). For example, in generation zero, consider an initial graph G. Let v_i be a vertex in G. Since all vertices in the initial graph are RC, v_i is referred to as (v_i, RC, 0) or in short, $v^0_{i,\mathrm{RC}}$. The index of generation is a positive number indicating the generation in which the vertex was born. A vertex $v^2_{i,\mathrm{RC}}$ means an RC vertex with label v_i, born in the second generation.

For describing the RC to NRC conversion rules, we have just replicated analogously Conway's Game of Life rule [23], wherein birth and death are the two states of the cell. We have just simulated the birth and death of the cell with RC vertices and NRC vertices, respectively. We do not attach any physical significance for choosing Conway's Game of Life for the RC to NRC conversion rule. We want to have some constraints by which an RC vertex becomes an NRC vertex and vice versa. Since Conway's Game of Life suits the purpose in the analogous sense, we have chosen that. A deep study in this direction may result in better conditions that could have some real-world meaning.

Now we state the reproduction rule.

Reproduction rule (RA^1):

1. All the vertices in the initial graph are RC vertices.
2. Any NB vertex will become an RC vertex in the next generation. That is, the generation in which the vertex is born is the first generation.
3. All RC vertices produce one vertex in a generation.
4. All NRC vertices are treated as infertile and hence will not produce any vertex in that generation.
5. RC vertices may become NRC vertices and vice versa based on the RC to NRC conversion rules described in (I).

II: Reproduction index (RI^1): The reproduction index denoted by k is a positive integer ($k \neq 0$) that constrains the number of offspring that can be produced by an organism. For example, if $\mathrm{RI} = 1$, that means that the organisms can produce only one offspring throughout their lifetime. Here $k \neq 0$, for the reason that our generative mechanism depends on reproduction alone.

III: Connectivity rule (CA^1): Connectivity rules are the rules that prescribe the edges connecting the NB vertices with those of the other vertices in the graph.

Connectivity rule: All the newborn vertices are connected to their respective parents and the offspring of their parent's neighbors.

Southwell and Cannings [22] described a graph-generating mechanism using the reproduction process. The authors detailed the eight models, with each model describing a specific mechanism of connectivity between the newborn vertices and the other vertices of the graph.

https://doi.org/10.25088/ComplexSystems.28.3.313

As mentioned in the RA^1 rule, we do not attach any specific meaning to the choice of model described as in Southwell's generating mechanism [19] for our connectivity rule.

IV: Loss-of-connectivity rule (LCA^1): This LCA rule describes the loss of connectivity of a vertex with others in due course of time. This loss of connectivity (due to various reasons) is an existing phenomenon in society.

Loss-of-connectivity rule:

1. There shall be no edge connecting an RC vertex with an NRC vertex and vice versa; that is, there will not be an edge of the form $(v^i_{1,\mathrm{RC}}, v^j_{2,\mathrm{NRC}})$ for any i, j.
2. There shall be no edge connecting two RC vertices that belong to different generations; that is, there will not be any edge of the form $(x^i_{\mathrm{RC}}, y^j_{\mathrm{RC}})$, $i \neq j$.

V: Dying rule (DA^1): This dying rule describes when the vertices die in the due course of time. This is also an existing phenomenon in society.

The dying rule: The vertices of degree zero (i.e., isolated vertices) that are NRC will die from one generation to the next generation if they become isolated.

With the rules described, we propose a GRS,

$$\rho = \left(\mathrm{RA}^1, \mathrm{RI}^1 = 1, \mathrm{CA}^1, \mathrm{LCA}^1, \mathrm{DA}^1\right),$$

where RA^1, RI^1, CA^1, LCA^1, DA^1 are the ones described earlier.

3.2 Illustrations

Illustrations for $\rho = \left(\mathrm{RA}^1, \mathrm{RI}^1 = 1, \mathrm{CA}^1, \mathrm{LCA}^1, \mathrm{DA}^1\right)$.

In the following figures, the vertices are labeled with the nature of the vertex as RC (reproduction-capable vertex), NRC (nonreproduction-capable vertex) and NB (newborn vertex) instead of the actual vertex label.

Example 1. Consider a complete graph G with three vertices (Figure 1). At the zeroth generation, by the RA^1 rule, the initial vertices will be the reproduction-capable (RC) vertices.

At the first generation, each RC vertex will produce an offspring called the newborn (NB) vertex by using the RA^1 rule. As per the connectivity rule (CA^1), all the NB vertices will get connected with the respective parent RC vertices. Further, all NB vertices will get connected with the offspring of neighbors of the parent RC vertices. As each of the initial RC vertices has two RC neighbors, RC vertices will remain RC vertices.

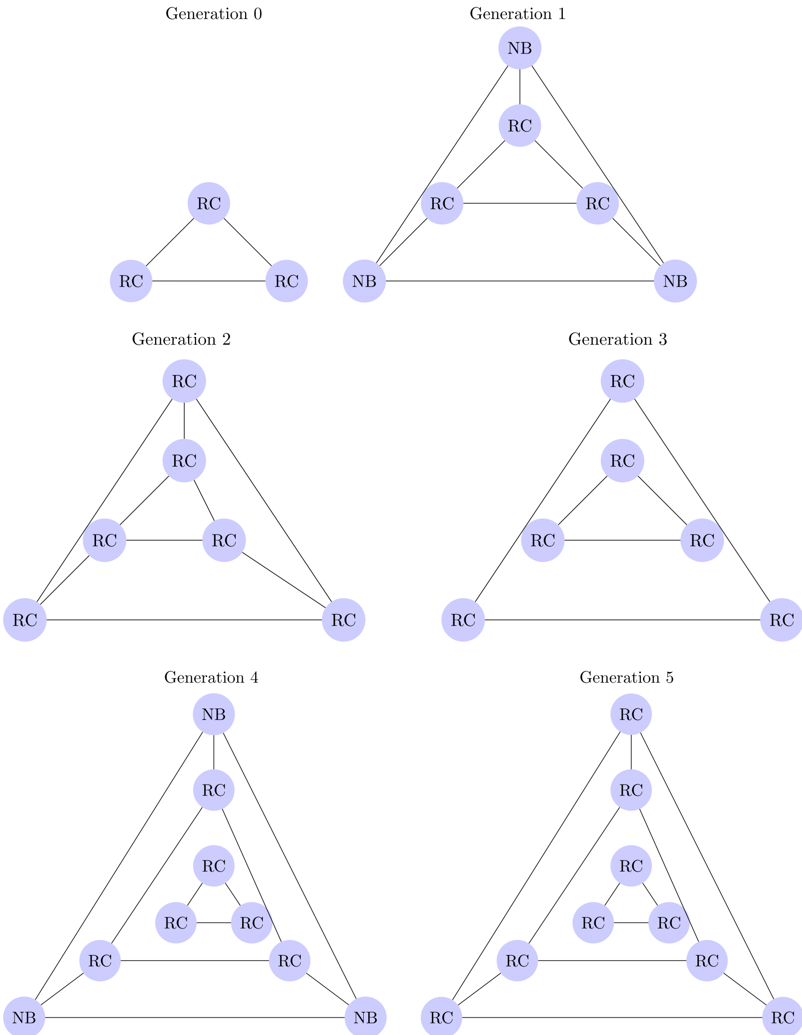


Figure 1. Illustration for complete graph on three vertices.

At the second generation, the three NB vertices will become RC vertices. As the reproducing index is 1, the initial RC vertices will not produce any NB vertices.

At the third generation, by the LCA[1] rule, the edges between RC and RC at different levels are removed and the original graph is reproduced. In fact, two isomorphic original graphs are obtained at this stage. That means self-replication can be seen at this stage.

In the fourth generation, each of the RC vertices produced at generation 1 will produce an NB vertex. The same process is continued, and we obtain the original graph at time step 6, time step 9 and so forth.

Example 2. Consider a path graph G with three vertices (Figure 2). At the zeroth generation, by the RA^1 rule, the initial vertices will be the reproduction-capable (RC) vertices.

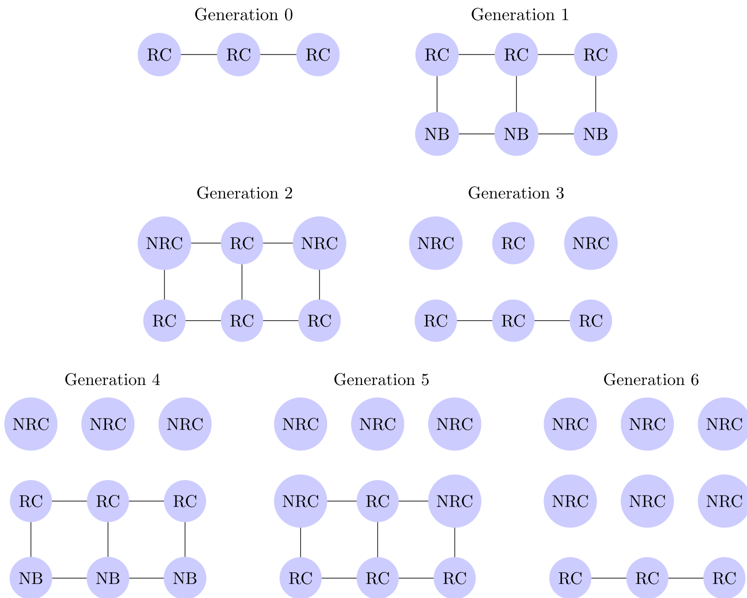


Figure 2. Illustration for path graph.

At the first generation, each RC vertex will produce an offspring called the newborn (NB) vertex by using RA^1 as described in I. As per the connectivity rule (CA^1), all the NB vertices will get connected with the respective parent RC vertices. Further, all NB vertices will get connected with the offspring of neighbors of the parent RC vertices.

At the second generation, as two of the initial RC vertices have one RC neighbor, they will become nonreproduction-capable vertices. One RC vertex has two RC neighbors, so it will remain an RC vertex. The three NB vertices will become RC vertices. As the reproducing index is 1, the initial RC vertices will not produce any NB vertices.

At the third generation, by the LCA^1 rule, the edges between RC and RC at different levels and edges between RC and NRC vertices are removed, and the original graph is reproduced. In fact, two isomorphic original graphs are obtained at this stage. That means self-replication can be seen at this stage.

In the fourth generation, each of the RC vertices produced at generation 1 will produce an NB vertex. The same process is continued, and we obtain the original graph at time step 6, time step 9 and so forth.

4. Self-Replicability of ρ

Now, we prove that the GRS ρ turns out to be a GSS.

Theorem 1. $\rho = \left(\mathrm{RA}^1, 1, \mathrm{CA}^1, \mathrm{LCA}^1, \mathrm{DA}^1\right)$ is a GSS of order 3. In fact, ρ is an inherent GSS of order 3.

Proof. Part I:

Consider a graph $G = (V, E)$. Let $V = \{v_1, v_2, \ldots, v_n\}$ and E be the edge set of the graph G.

We describe the proof in stages.

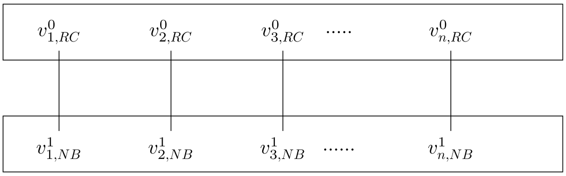
Stage 1: Generation 0: In the initial graph G, let the vertices be $v_1^0, v_2^0, \ldots, v_n^0$. All the v_i are RC vertices.

Stage 2: Generation 1: Now, all the $v_{i,\mathrm{RC}}^0$ will produce an offspring. Let the offspring of the v_i^0 be called v_i^1, respectively.

As per the connectivity rule CA^1, all the $v_{i,\mathrm{NB}}^1$ vertices will get connected with the respective $v_{i,\mathrm{RC}}^0$. Further, $v_{i,\mathrm{NB}}^1$ will get connected with the offspring of neighbors of $v_{i,\mathrm{RC}}^0$ (as in the graph G).

In other words, there will be some edges connecting $v_{i,\mathrm{NB}}^1$ with $v_{j,\mathrm{NB}}^1$.

By the CA^1 rule, all the offspring are connected to the offspring of the neighbors of the parents; that is, if there is an edge between the parents $(v_{i,\mathrm{RC}}^0, v_{j,\mathrm{RC}}^0)$, then there will be an edge between $(v_{i,\mathrm{NB}}^1, v_{j,\mathrm{NB}}^1)$. Due to this, the adjacency maintained among the $v_{i,\mathrm{RC}}^0$ vertices is preserved among the $v_{i,\mathrm{NB}}^1$ vertices; that is, the block diagram of $\rho^1(G)$ will look like:



Here, all the vertices that are adjacent in the top rectangle will have the adjacencies preserved with the respective vertices in the bottom rectangle.

In fact, the graph in the top rectangle is the original graph and is isomorphic to the graph in the bottom rectangle.

It is clear that $\rho^1(G) \bigcap \{G\} \neq \Phi$.

Stage 3: Generation 2: By the RA[1] rule, all the $v^0_{i,\,\mathrm{RC}}$ vertices will become either $v^0_{i,\,\mathrm{RC}}$ (retaining RC) or $v^0_{i,\,\mathrm{NRC}}$, depending on the nature of the respective neighbors.

All the $v^1_{i,\,\mathrm{NB}}$ vertices of $\rho^1(G)$ will become $v^1_{i,\,\mathrm{RC}}$.

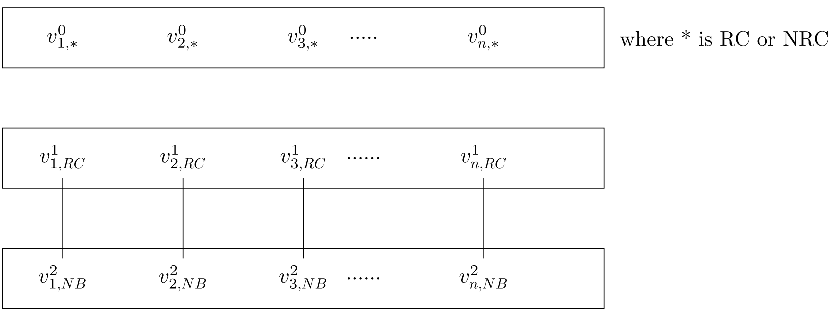
All the $v^1_{i,\,\mathrm{RC}}$ vertices produce offspring $v^2_{i,\,\mathrm{NB}}$ vertices.

As per the CA[1] rule, all the $v^2_{i,\,\mathrm{NB}}$ vertices will have adjacency with the offspring of the neighbors of $v^1_{i,\,\mathrm{RC}}$ vertices. That is, now we will have three rectangles (as in the next figure), preserving the adjacency among the respective vertices.

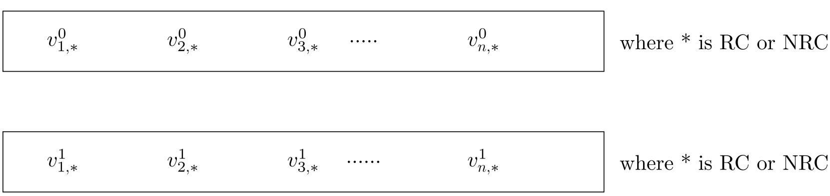
Now, because the RI value is 1, $v^0_{i,\,\mathrm{RC}}$ vertices will not produce any offspring.

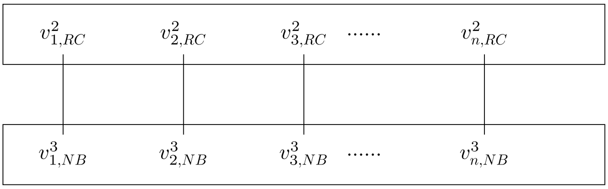
By the LCA[1] rule, the edges connecting RC vertices of different levels as well as vertices connecting RC vertices and NRC vertices will be lost.

In other words, the edges connecting the top and middle block will be lost.



Now all three blocks are isomorphic. Continuing in the same way, we have:





Stage 4: Generation 3: Now, we have, $\rho^3(G) \bigcap \{G\} \neq \Phi$.

In fact, we have two copies of G replicated. That is, the top two blocks individually are copies of the initial graph G; we can continue this process. Since we have proved $\rho^3(G) \bigcap \{G\} \neq \Phi$, ρ is self-replicable of order 3.

Part II:

G is any arbitrary graph and $\rho^3(G) \bigcap \{G\} \neq \Phi$, $\forall\, G$. This implies that ρ is inherently a GSS of order 3. □

4.1 Simulation of Graph Reproduction System

Thus, we have proved the self-replicability of our GRS $\boldsymbol{\rho}$ as an inherent GSS of order 3, with full rigor, beyond any ambiguity. With the aim of bringing out any hidden property of our GRS that is not normally observed, we captured the dynamic behavior of our GRS $\boldsymbol{\rho}$ by simulating with MATLAB, using appropriate code. An adjacency matrix that corresponds to the initial graph in our GRS formed input for our model, and the output is also obtained as an adjacency matrix. The MATLAB code used for the purpose is available at [24].

We simulated our GRS with various initial graphs as input and studied the behavior of our GRS by simulating to a maximum of 100 generations. The evolution of the complete graph on three vertices described in Section 3, Example 1, obtained through our simulation procedure is provided at [24].

As expected, results of our simulation concurred with the theoretical results of $\boldsymbol{\rho}$, proved in this section. Besides, as an outcome of our analysis of $\boldsymbol{\rho}$ through simulation, we observed the following properties of $\boldsymbol{\rho}$.

Result 1. Let $\boldsymbol{\rho}$ be a GSS of order k. Given an initial graph G, $\boldsymbol{\rho}$ will produce j copies of G; at the end of the jk^{th} generation, $j \neq 0$. Including the parent graph, there will be a total of $j+1$ copies of G. Further, none of the reproduced copies of G will have an edge connecting them.

Result 2. Let G be a disconnected graph with n disconnected components, say G_1, G_2, $\ldots G_n$. Let $\boldsymbol{\rho}$ be a GSS of order k. Then at the end of the jk^{th} generation, we will have j copies of the individual components G_1, G_2, $\ldots G_n$.

Result 3. The process of self-replication by ρ ensures that all the characteristics of the parent graph are inherited by the offspring graph.

Result 4. The process of self-replication produces a disconnected graph in which the parent and the offspring vertices form the disconnected components of the graph.

5. Existing Graph Reproduction Models

Southwell and Cannings explored different graph-reproducing models [19] that produce different types of graphs from the given initial graph. In their models, every vertex produces a new vertex that is connected to the existing vertices based on different constraints, thereby generating eight different models of the graph-generating mechanism. In their models, edges get eliminated because of the death of vertices. The models concentrate on the evolution of different types of graphs, and self-replication of the given graph is not the authors' concern. The models proposed in [19] are the specific instances of our GRS. The eight models are represented through our GRS as $\rho_i = (\text{RA}_i, \text{RI}_i, \text{CA}_i, \text{LCA}_i, \text{DA}_i)$, where $i = 0, 1, 2, 3, 4, 5, 6, 7$.

In all eight models, the rules other than the connectivity rule remain the same. They differ only in the connectivity aspect; that means the models can be differentiated only with respect to the connectivity rule.

The components in the preceding five-tuple are described as:

- RA_i: All vertices produce offspring vertices one at a time $\forall\ i$.
- $\text{RI}_i = 1 \forall\ i$; that means all the vertices produce only one offspring vertex throughout their lifetime.
- $\text{LCA}_i = \Phi$ (empty set) $\forall\ i$.
- DA_i: Q, a positive integer, which means that every vertex of degree greater than Q will die.

The connectivity axioms are described as follows:

- CA_0: No offspring are connected among themselves and no offspring are connected to their parent.
- CA_1: Offspring are connected to their parent's neighbors.
- CA_2: Offspring are connected to their parent.
- CA_3: Offspring are connected to their parent and their parent's neighbors.
- CA_4: Offspring are connected to the offspring of their parent's neighbors.

- CA_5: Offspring are connected to their parent's neighbors and their parent's neighbors' offspring.
- CA_6: Offspring are connected to their parent and the offspring of their parent's neighbors.
- CA_7: Offspring are connected to their parent, their parent's neighbors and the offspring of their parent's neighbors.

The GRS $\boldsymbol{\rho}_i$ corresponds to the i^{th} model described in [19].

Apart from the respective descriptions of RA_i, CA_i, $\boldsymbol{\rho}_i = (RA_i, 1, CA_i, \Phi, Q)$ conveys that LCA_i is empty, RI_i is 1 and $DA_i = Q$.

We now investigate Southwell's models for self-replicability.

Theorem 2. Theorem:

1. $\boldsymbol{\rho}_0 = (RA_0, 1, CA_0, \Phi, Q)$ is a GSS of order 1. Further, $\boldsymbol{\rho}_0$ will replicate all graphs $G = (V, E)$ such that $\deg(v_i) \le Q$ for at least one $v_i \in V$ and will not replicate any graph G such that $\deg(v_i) > Q$.
2. $\boldsymbol{\rho}_4 = (RA_4, 1, CA_4, \Phi, Q)$ is a GSS of order 1. Further, $\boldsymbol{\rho}_4$ will replicate all graphs $G = (V, E)$ such that $\deg(v_i) \le Q \forall\, v_i \in V$. $\boldsymbol{\rho}_4$ will not replicate any graph G such that $\deg(v_i) > Q$ for at least one $v_i \in V$.
3. $\boldsymbol{\rho}_6 = (RA_6, 1, CA_6, \Phi, Q)$ is a GSS of order 3. Further, $\boldsymbol{\rho}_6$ will replicate all graphs $G = (V, E)$ such that $\deg(v_i) \le Q \forall\, v_i \in V$. $\boldsymbol{\rho}_6$ will not replicate any graph G such that $\deg(v_i) > Q$ for at least one $v_i \in V$.

Proof. Without loss of generality, we prove the theorem for one model, $\boldsymbol{\rho}_4$. The other proofs can be done analogously.

Given $\boldsymbol{\rho}_4 = (RA_4, 1, CA_4, \Phi, Q)$, let $G = (V, E)$ be a graph where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices of G and E is the set of edges of G.

We describe the proof in stages.

Stage 1: Generation 0: Let the vertices in the zeroth generation be $v_1^0, v_2^0, \ldots, v_n^0$. All the v_i^0 are parent vertices.

Stage 2: Generation 1: Now, all the v_i^0 will produce an offspring vertex. Let the offspring of the v_i^0 be denoted as v_i^1, respectively.

As per the connectivity rule CA_4, all the offspring (v_i^1) vertices will get connected with the offspring of neighbors of parent (v_i^0) vertices. That is, if (v_i^0, v_j^0) is an edge in generation 0, there will be a new edge (v_i^1, v_j^1) in generation 1 and there will be no other edge connecting any of the vertices of generation 0 with any of the vertices of generation 1.

In other words, (1) adjacency among the parent vertices is preserved among the offspring vertices. Further, (2) $\deg(v_i^0) = \deg(v_i^1) = \deg(v_i)$.

Case 1: Suppose $\deg(v_i) \le Q \forall\, v_i \in V$. By (1) $\deg(v_i^0) = \deg(v_i^1) \le Q$. No vertices will die because of the dying rule. By (1) and (2) we will have a copy of the original graph reproduced at the end of generation 1. Hence $\boldsymbol{\rho}_4$ is a GSS of order 1.

Case 2: Suppose $\deg(v_i) > Q$ for at least one $v_i \in V$.

Let v_k be a vertex in G such that $\deg(v_k) > Q$. This implies $\deg(v_k^0) > Q$. By (1), $\deg(v_k^0) = \deg(v_k^1) > Q$. By DA_4, v_k^0 and v_k^1 will die and all the edges incident on v_k^0 and v_k^1 will get eliminated.

Vertices in generation 1 are v_1^0, $v_2^0, \ldots v_{k-1}^0$, $v_{k+1}^0, \ldots v_n^0$; v_1^1, $v_2^1, \ldots v_{k-1}^1, v_{k+1}^1, \ldots v_n^1$.

Thus, by (1), the graph formed with the vertices v_i^1, $i = 1, 2, \ldots, k-1, k+1, \ldots n$ is the graph G with v_k eliminated. That is, a copy of G could not be obtained in generation 1. Now the degree of all the vertices of generation 1 is less than or equal to Q. If we continue the reproduction process, by case 1, we get a graph formed with the vertices v_i^1, $i = 1, 2, \ldots, k-1, k+1, \ldots n$. That is, a copy of G without the vertex v_k will get generated in generation 3 and also in the subsequent generations. Thus the original graph G cannot be obtained for $\deg(v_i) > Q$ for at least one $v_i \in V$.

Hence in this case, $\boldsymbol{\rho}_4$ is not a GSS. □

As evident from the statement of Theorem 2, $\boldsymbol{\rho}_0$, $\boldsymbol{\rho}_4$ are GSSs of order 1, whereas $\boldsymbol{\rho}_6$ is a GSS of order 3. Further, self-replicability of a graph under $\boldsymbol{\rho}_0$, $\boldsymbol{\rho}_4$ or $\boldsymbol{\rho}_6$ depends upon the degree of the vertices of G.

Theorem 3. $\boldsymbol{\rho}_i$ for $i = 1, 2, 3, 5, 7$ are not a GSS.

Proof. Without loss of generality, we prove the theorem for one model, that is for $\boldsymbol{\rho}_3$. The other proofs can be done analogously.

Given $\boldsymbol{\rho}_3 = (RA_3, 1, CA_3, \Phi, DA_3)$, let $G = (V, E)$ be a graph where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices of G and E is the set of edges of G.

We describe the proof in stages.

Stage 1: Generation 0: Let the vertices in the zeroth generation be $v_1^0, v_2^0, \ldots v_n^0$. All the v_i^0 are parent vertices.

Stage 2: Generation 1: Now, all the v_i^0 will produce an offspring vertex. Let the offspring of the v_i^0 be represented as v_i^1, respectively.

As per the connectivity rule CA_3, all the offspring (v_i^1) vertices will get connected with the respective parent vertices and with the neighbors of respective parent (v_i^0) vertices.

For every adjacent vertex, the degree of v_i^0 will get increased by one (in CA_3 there will be an edge connecting the parent's neighbor and the offspring). There will also be an edge connecting the v_i^0 with its own offspring. Hence if $\deg(v_i^0)$ is k, the $\deg(v_i^0)$ in generation 1 will increase by $2k+1$. The $\deg(v_i^1)$ (v_i^1 are offspring vertices in generation 1) will be $k+1$, where k is the $\deg(v_i^0)$ in generation 0. In CA_3, there will be no edge connecting any two offspring vertices v_i^1 and v_j^1 for any i, j.

Case 1: $\deg(v_i^k) > Q$ for any i, k is either 0 or 1.

If $\deg(v_i^0) > Q$, for any i, then by the dying rule DA_3, v_i^0 vertices will die, along with the incident edges. In such a case, the degree sequence of the graph formed by the vertices v_i^0 (parent vertices) or the degree sequence of the graph formed by the vertices v_i^1 (offspring vertices) or the degree sequence of the vertices v_i^0 and v_i^1 (both the parent vertices and offspring vertices) will not be the same as the degree sequence of the original graph G. Thus we cannot get a copy of G in generation 1.

Case 2: $\deg(v_i^k) < Q$ for all i, k is either 0 or 1.

None of the vertices will die, since there is no edge connecting the offspring vertices, and due to the increase in the degree of the parent vertices (v_i^0), the degree sequence of the graph involving the parent vertices alone will not be the same as that of the degree sequence of the original graph G. The scenarios with the degree sequence of the graph involving the offspring vertices alone and the degree sequence of the graph involving both offspring and the parent vertices are the same. Hence, a copy of G cannot get reproduced in generation 1. We will be encountering a similar situation in the subsequent generations. Hence a copy of G will not get reproduced in any of the generations. So $\boldsymbol{\rho}_3$ is not a GSS. □

Thus, of the eight models, we conclude that three models are self-replicable and the remaining models are not self-replicable.

6. Conclusions and Future Work

In this paper we have described a graph reproduction system (GRS) with five components such as RA, RI, CA, LCA and DA that is capable of self-replication. In fact, our GRS provides an algorithmic process to produce multiple copies of the initial graph. Here we have described the RA, CA, LCA and DA, without attaching much physical significance to the description. We can generate more ρ's by varying all the components of ρ, which may result in an exponential number of GRS. A thorough study of the GRS by varying all the components of ρ will bring out a characterization of a graph self-replication system (GSS). We have investigated the Southwell models through our system and verified which models are self-replicable. In [25], the authors attempted to develop a replication system based on the rolling-circle replication of a circular DNA coupled with recombination using self-encoded phi29 DNA polymerase and externally supplied Cre recombinase. We want to define a composite GRS, wherein more than one GRS is involved with a specific iterative process for the generation of languages. We plan to investigate whether or not composite GRSs are self-replicative, thus introducing an operation among the set of all GRSs, and to check which composite operations are self-replicative or not.

References

[1] R. A. Freitas, Jr. and R. C. Merkle, *Kinematic Self-Replicating Machines*, Georgetown, TX: Landes Bioscience, 2004.

[2] M. Sipper, "Fifty Years of Research on Self-Replication: An Overview," *Artificial Life*, **4**(3), 1998 pp. 237–257. doi:10.1162/106454698568576.

[3] J. von Neumann, *Theory of Self-Reproducing Automata* (A. W. Burks, ed.), Urbana, IL: University of Illinois Press, 1966.

[4] J.-L. Beuchat and J. O. Haenni, "Von Neumann's 29-State Cellular Automaton: A Hardware Implementation," *IEEE Transactions on Education*, **43**(3), 2000 pp. 300–308. doi:10.1109/13.865205.

[5] C. G. Langton, "Self-Reproduction in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1–2), 1984 pp. 135–144. doi:10.1016/0167-2789(84)90256-2.

[6] E. Codd, *Cellular Automata*, New York: Academic Press, 1968.

[7] J. D. Lohn and J. A. Reggia, "Discovery of Self-Replicating Structures Using a Genetic Algorithm," in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation (ICEC'95)*, Perth, WA, Australia, 1995, Piscataway, NJ: IEEE, 1995 pp. 678–683. doi:10.1109/ICEC.1995.487466.

[8] J. D. Lohn, "Automated Discovery of Self-Replicating Structures in Cellular Space Automata Models," Department of Computer Science, University of Maryland at College Park, Tech. Rep. CS-TR-3677, 1996. hdl.handle.net/1903/468.

[9] J. D. Lohn and J. A. Reggia, "Automatic Discovery of Self-Replicating Structures in Cellular Automata," *IEEE Transactions on Evolutionary Computation*, **1**(3), 1997 pp. 165–178. doi:10.1109/4235.661547.

[10] D. Emmite, J. A. Reggia and M. Sipper, "Go Forth and Replicate," *SA Special Editions*, **18**(1s), 2008 pp. 48–57. doi:10.1038/scientificamerican0208-48sp.

[11] P. Bratley and J. Millo, "Computer Recreations: Self-Reproducing Programs," *Software Practice and Experience*, **2**, 1972 pp. 397–400. fab.cba.mit.edu/classes/865.18/replication/Bratley.pdf.

[12] J. Burger, D. Brill and F. Machi, "Self-Reproducing Programs," *Byte*, **5**, 1980 pp. 72–74.

[13] T. S. Ray, "An Approach to the Synthesis of Life," *Artificial Life II: Proceedings of the Workshop on Artificial Life* (C. G. Langton, et al., eds.), Santa Fe, NM, 1990, Redwood City, CA: Addison-Wesley, 1992 pp. 371–408.

[14] T. S. Ray, "An Evolutionary Approach to Synthetic Biology: Zen and the Art of Creating Life," *Artificial Life*, **1**(1–2), 1993 pp. 179–209. doi:10.1162/artl.1993.1.1_2.179.

[15] R. Laing, "Automaton Models of Reproduction by Self-Inspection," *Journal of Theoretical Biology*, **66**(3), 1977 pp. 437–456. doi:10.1016/0022-5193(77)90294-6.

[16] L. S. Penrose, "Self-Reproducing Machines," *Scientific American*, **200**(6), 1959 pp. 105–117. www.jstor.org/stable/26309511.

[17] J. Rebek Jr., "Synthetic Self-Replicating Molecules," *Scientific American*, **271**(1), 1994 pp. 48–55. www.jstor.org/stable/24942766.

[18] N. Ichihashi and T. Yomo, "Constructive Approaches for Understanding the Origin of Self-Replication and Evolution," *Life*, **6**(3), 2016 26. doi:10.3390%2 Flife6030026.

[19] R. Southwell and C. Cannings, "Some Models of Reproducing Graphs: 1 Pure Reproduction," *Applied Mathematics*, **1**(3), 2010 pp. 137–145. doi:10.4236/am.2010.13018.

[20] R. Southwell and C. Cannings, "Some Models of Reproducing Graphs: 2 Age Capped Vertices," *Applied Mathematics*, **1**(4), 2010 pp. 251–259. doi:10.4236/am.2010.14031.

[21] R. Southwell and C. Cannings, "Some Models of Reproducing Graphs: 3 Game Based Reproduction," *Applied Mathematics*, **1**(5), 2010 pp. 335–343. doi:10.4236/am.2010.15044.

[22] R. Southwell and C. Cannings, "Games on Graphs That Grow Deterministically," in *International Conference on Game Theory for Networks (Nets'09)*, Istanbul, 2009, Piscataway, NJ: IEEE, 2009 pp. 347–356. doi:10.1109/GAMENETS.2009.5137420.

[23] M. Gardner, “Mathematical Games: The Fantastic Combinations of John H. Conway’s New Solitaire Game ‘Life’,” *Scientific American*, **223**(4), 1970 pp. 120–123. www.jstor.org/stable/24927642.

[24] V. L. Dasari. “MATLAB Code for Simulation of the Graph Reproduction System.” (Oct 4, 2019). in.mathworks.com/matlabcentral/fileexchange/72869-matlab-code-for-simulation-of-the-graph-reproduction-system.

[25] Y. Sakatani, T. Yomo and N. Ichihashi, “Self-Replication of Circular DNA by a Self-Encoded DNA Polymerase through Rolling-Circle Replication and Recombination,” *Scientific Reports*, **8**(1), 2018 13089. doi:10.1038/s41598-018-31585-1.