2D Hexagonal Cellular Automata: The Complexity of Forms

Vural Erdogan
ve17aae@herts.ac.uk

Abstract — **We created two-dimensional hexagonal cellular automata to obtain complexity by using simple rules same as Conway's game of life. Considering the game of life rules, Wolfram's works about life-like structures and John von Neumann's self-replication, self-maintenance, self-reproduction problems, we developed 2-states and 3-states hexagonal growing algorithms that reach large populations through random initial states. Unlike the game of life, we used six neighbourhoods cellular automata instead of eight or four neighbourhoods. First simulations explained that whether we are able to obtain sort of oscillators, blinkers and gliders. Inspired by Wolfram's 1D cellular automata complexity and life-like structures, we simulated 2D synchronous, discrete, deterministic cellular automata to reach life-like forms with 2-states cells. The life-like formations and the oscillators have been explained how they contribute to initiating self-maintenance together with self-reproduction and self-replication. After comparing simulation results, we decided to develop the algorithm for another step. Appending a new state to the same algorithm, which we used for reaching life-like structures, led us to experiment new branching and fractal forms. All these studies tried to demonstrate that complex life forms might come from uncomplicated rules.**

Keywords: **Hexagonal cellular automata, self-replication, self-maintenance, self-reproduction**

I. Introduction

Artificial Life attracts scientists' attention since studying life-as-it-might-be approach instead of life-as-we-know-it approach [1]. One of the most crucial point that contributes to Artificial Life (Alife) being attractive investigate simple rules in the background of the complex systems. Therefore, Alife scientists endeavour to simulate and analyse of simple principles in order to achieve a complex system such as life-like.

Cellular Automata (CA), which is a subtopic of Alife, have a shared history with Alife. Cellular automata accompany simulations of life-like systems by systematising simple mathematical equations. One of the famous examples of CA is "Conway's Game of Life" algorithm (Martin Gardner, 1970) [2], a 2D cellular automaton with 2-states, rectangular and Moore neighbourhood basis. Each rectangular cell depends on a rule that allows only two states "Alive" or "Dead". With this algorithm, Richard K. Guy has discovered a spaceship which is named "glider" in 1970 that oscillates by moving to other grids [2]. The gliders could contribute to creating a virtue world and new formations in cellular automata. Also, the other findings such as blinkers, an oscillator that returns its original state, have an essential role in building new structures. The gliders, the blinkers and the other oscillators such these might enhance the complexity of patterns.

Another famous rule of CA is Stephen Wolfram's "elementary cellular automata" [3] that inspired Alife scientists remarkably due to the complexity of rule 30 that was created in a one-dimensional environment. After his systematic studies about complex cellular automata, he published "A New Kind of Science" [4] in 2002 that demonstrates cellular automata can be related to many fields.

Besides these, one of the pioneers of cellular automata, John von Neumann, investigated self-replication and self-reproduction problems to build a "universal constructor" that is a self-replicating machine building own copies by using artefacts of an environment and was designed as a draft plan without a computer in 1940 by him. He tried to explain the self-reproduction problem, that is more complicated than self-replication because of creating a new structure that must be at least as complex as itself, in his studies. However, he could not share vital details of self-reproduction throughout his life. Instead of him, Arthur Walter Burks published the theory of self-reproduction [5] by completing his thesis after his death. These problems are still being part of the discussions today, especially self-reproduction is the focus of the researchers' due to its complexity.

This paper illustrates an example of how to obtain a life-like structure complexity in cellular automata by using simple equations with the hexagonal neighbourhood. In the A section, we indicate how to obtain hexagonal logic environment with rectangular grids. We determine some notations to clarify features of the rules in part B. Afterward, we compare the algorithms by terming them as "growing", "swinging" and "melting" and simulate them by explaining their details in the C section. The paper includes three experiment sections about the gliders and the blinkers similarly Richard's glider oscillators, the complexity of structures that inspired Wolfram's life-like structures and John von Neumann's self-replication, self-reproduction and self-maintenance problems [6]. The first experiments of the first proposed rule in section D show the feasibility of oscillators in the hexagonal environment. The second simulation examples illustrate attempting of self-reproduction of gliders and developing fractal-like and branching structures which aimed to increase the complexity for the forms in chapter E; the branching rule includes three-states that has been modified from another proposed rule. Chapter F explains an example of self-maintenance, and chapter E indicate a self-replication instance by using different rule.

One of the reasons why hexagonal cells used for this experiment is to increase the possibility of growing when gliders interact each other. The game of life oscillates with a “swinging rule”, explained in the C section, which does not tend to grow up. With new hexagonal rules, we are able to interact other cells conveniently with fewer neighbours. Another reason is that hexagonal grids have only a few examples in literature, e.g. Paul Callahan proposed one of them in 1997 which indicates diverse glider oscillators and complexity [7]. Another instance is 3-states glider-based cellular automata explaining self-reproduction, self-destruction, gliders and glider-guns (a glider producer) proposed by Andrew Wuensche in 2005 [8].

II. Hexagonal Cellular Automata

A. Hexagonal grids

TABLE I
ABBREVIATIONS OF DIRECTIONS

Abbr.	Directions
NW	North West
NE	North East
SW	South West
SE	South East
N	North
W	West
E	East
S	South
C	Centre

Hexagonal grid rule is proposed by Ken Preston Jr, first described in in 1971. The rule explains how to transform rectangular environment to hexagonal environment. Each cell represents a direction of the cell, and we used the abbreviation of direction for the rule such as Table I. A rectangular centre has eight neighbours; however, a hexagonal centre must have six neighbours. Therefore, the hexagonal grid rule proposes neglecting “South West” and “North East” directions (Fig. 1).

We decided to simulate our experiments with hexagonal rules as it eases interacting with the cells. If we look left at 45° degrees horizontally, we can see a hexagonal neighbourhood (Fig. 2).

NW	N	NE
W	C	E
SW	S	SE

→

NW	N	
W	C	E
	S	SE

Fig. 1. Transformation of the hexagonal neighbourhood.

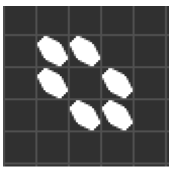

Fig. 2. A sample of six hexagonal neighbours of the death cell with a hexagon icon.

For the simulations, “Golly” software is used for all experiments. The software has rule generator with transition function that converts python codes to rule automatically. That contributed us to experiment several variations in a short time.

B. Notations

We determined notations for the rules just as the other proposed rules. For instance, the game of life rule equates to “B3S23” [2]:

- “Birth” happens in the centre if there are three neighbours.
- The centre cell “Survives” if there are two or three neighbours.
- Otherwise, all the exceptions result in death.

For hexagonal cells, we called them “ortho”, “meta”, “para” just as Paul Callahan [7] and added a new term “meta-para” which equates to “o”, “m”, “p” and “mp” respectively. Fig. 3 illustrates ortho, meta, para and meta-para positions of neighbours across the reference point that is shown as “R”. For mp position, two cells must be adjacent, and one cell must be separated from those two.

We named the beginning of the name of rules as “hex” to clarify that is a hexagonal rule. The other notation is about multiple states rules. We preferred brackets to indicate this, e.g. “(number)”. If there are more than two states in the algorithm, we can use brackets to notify it (the notations are explained in chapter C together with details).

Fig. 3. Notations of hexagonal neighbours: 1-) o, m, p positions, 2-) mp position.

C. Features of rules and simulations

Before starting experiments, we divided rules into three; “melting”, “swinging” and “growing”. For instance, after five experiments “Hex-B3S23”, is similar to the game of life but with hexagonal grids, classified as a melting algorithm unlike the game of life (Fig. 4). The initial states are filled randomly by using software’s feature and chosen in 6 squares each time.

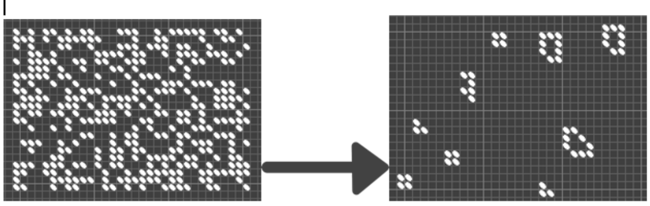

Fig. 4. The initial state of “Hex-B3S23” and after 30 generation.

The game of life is classified as a swinging algorithm as it still oscillates after 1000 generations, but neither it tends to grow up nor it vanishes (Fig. 5).

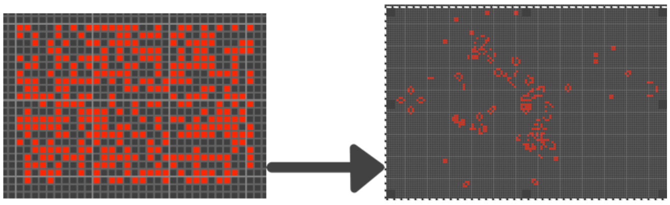

Fig. 5. The initial state of the "B3/S23" and after 1000 generation.

One of the examples of the proposed rules is "Hex-B2S2o3m4", which is named "the honeycomb" by the author due to its stable structure, classified as growing algorithm since it tends to grow up rapidly (Fig. 6).

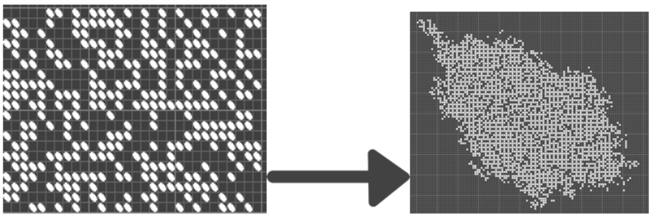

Fig. 2. The initial State of "the honeycomb" and after 289 generation.

The rules of the honeycomb consist of these statements:

- If there are two any neighbours, and centre is a dead cell, birth happens.
- If there is a centre cell that has two ortho neighbours, it survives.
- If there is a centre cell that has three meta neighbours, it survives.
- If there is a centre cell that has four neighbours, it survives.
- Otherwise, all the exceptions result in the death.

D. The oscillators: Blinkers and Gliders

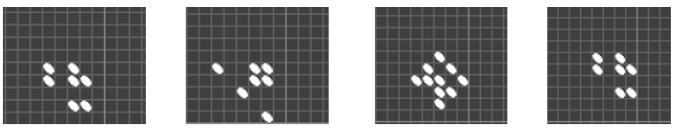

Fig. 7. A glider. Generations: 1, 2, 3, 4

After classification, studying the growing algorithms could contribute to our experiments since the gliders need to interact other cells, blinkers and other gliders by colliding. In order to produce gliders, random initial states were chosen for the experiment and those growing structures were observed during the experiments. Even though the software applies a hash life algorithm, was proposed by Bill Gosper in 1984, that processes small area firstly then it enhances processing area according to coordination of cells in brief to increase the speed of simulating generations, it still slows down after reaching 1 million population [9]. If the rule does not show any oscillator in approximately 10 minutes, it should be altered with another rule since increasing population causes decreasing the rapidity of simulations. In that way, we could save our time. Random and manual initial states, the simulations illustrated some glider structures and blinkers for rule "Hex-B2S23mp" (Fig. 7).

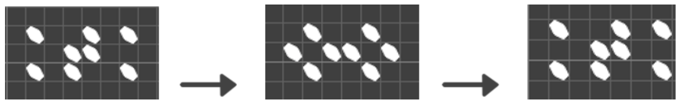

Fig. 8. The conveyor blinker. Generations: 1, 2, 3.

The experiment result has an adjacent reference grey line to clarify that the glider is shifting. The second rule Hex-B2S23mp has these features:

- If there are two any neighbours, and centre is a dead cell, birth happens.
- If there is a centre cell that has any two neighbours, it survives.
- If there is a centre cell that has three meta-para neighbours, it survives.
- Otherwise, all the exceptions result in the death.

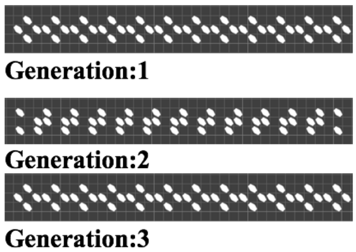

Fig. 9. Generation of the conveyor blinker

Fig. 8 indicates a blinker form of rule Hex-B2S23mp. We termed it "the conveyor blinker". Its structure is similar to two-sided conveyor band. Moreover, it is not affected when another conveyor blinker is appended it (Fig. 9).

E. Self-reproduction and the complexity of the branching forms

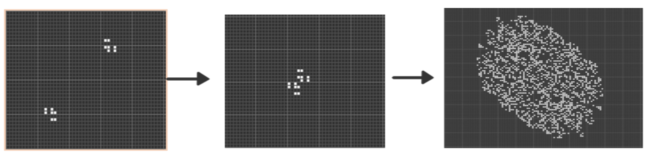

Fig. 10. Glider explosion, Generations: 1, 24, 276

Discovering the gliders and the blinkers led us to do a study in range area. For example, glider-blinker interactions or glider-glider collisions could be observed. If colliding gliders could grow up, and if this was being swift, self-production and self-replication would be possible (Fig. 10). Another point is that growing structures had to distribute gliders to outside. Symmetrical glider spreading increases the possibility of glider meeting. As a result, selecting manually symmetrical initial states saved more time when we simulated experiments for self-reproduction or self-replication. We triggered with symmetrical initial states to produce a glider spreading structure. After experiments, one of them sprat the gliders symmetrically (Fig. 11). The giant structure is named "amoeboid" due to similarity of its temporary opening. To show self-production, we used a different layer to prevent collision of the amoeboid and the new structure. The copy of the new structure in the new layer illustrated that the new structure was as complex as the amoeboid (Fig. 12).

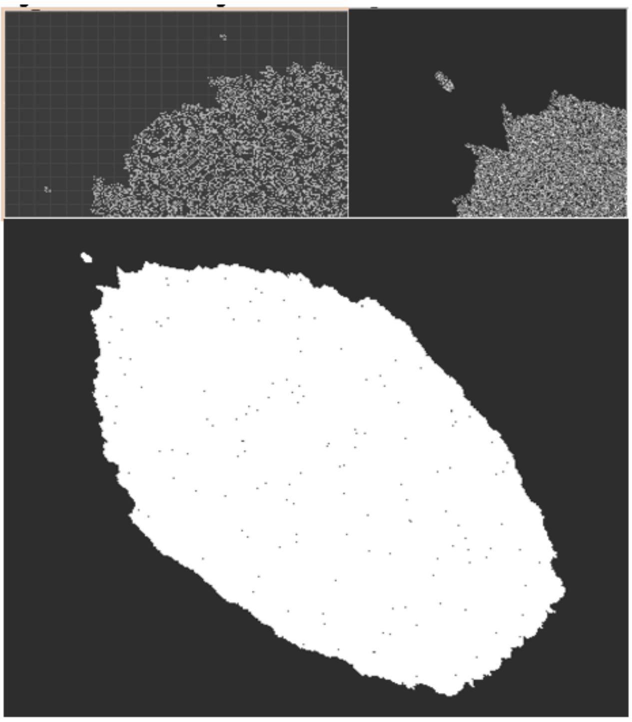

Fig. 11. The meeting of gliders and the amoeboid, Generation: 2057, 2420, 2420. Population: 320.430, 430.729, 430.729

While watching the large populations' simulations of Hex-B2S23mp, we realised that the amoeboid or the other similar forms were attempting branching, trying to produce long pointed roots or arms; nevertheless, the branches were collapsing immediately due to lack of rule's feature. The branches or the roots, named by the author, were not sufficient for initiating. Producing the branches or the fractals were attractive to discover that pushed us to alter the rule with a new rule. On the other hand, the gliders were not supposed to be disappeared. For this reason, we preferred to add only new states instead of changing the entire rule. Nearly six rules later, the third rule has been founded surprisingly. The rule is named "Hex-B2S23mp(2)" much similar to the second rule. However, it has three states; "death", "sick" and "alive" that equal to 0, 1 and 2 points respectively.

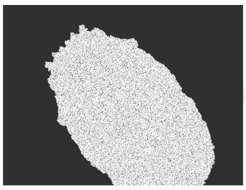

Fig. 12. The new structure, produced by the amoeboid. Generation: 2057 Population: 278,017

In other words, if there is an alive neighbour, we consider that we have two neighbours:

- If there are two any neighbours or points, and centre is a dead cell, a sick cell occurs.
- If there is a centre cell that has any two neighbours or points, it survives as a sick cell.
- If there is a centre cell that has three meta-para neighbours or points, it survives as an alive cell.
- Otherwise, all the exceptions result in the death.

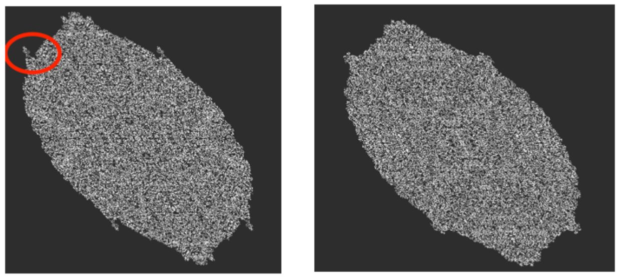

Fig. 13. a) The branching attempts of the structure Generation: 700, Population: 47.184
b) Failing of the branching attempts Generation: 719, Population: 50,072.

The red circle shows a branching attempts of the rule Hex-B2S23mp; there are four of them symmetrically in Fig 13. We tried to compare Fig. 13 with a similar form such as Fig. 14 that illustrates the roots, in other words, a fractal formation unlike the Hex-B2S23mp(2).

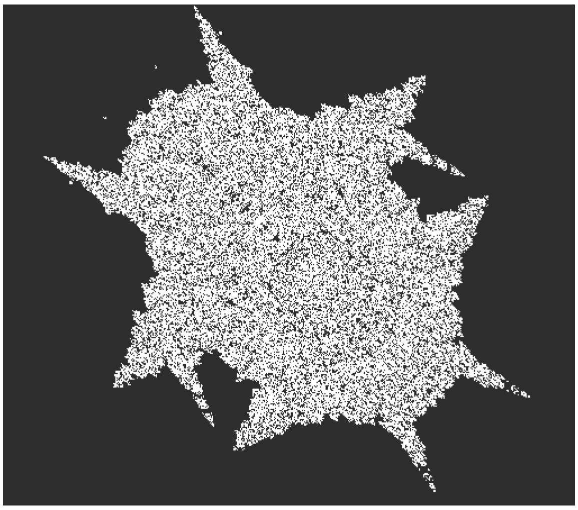

Fig. 14. The branching attempts of the rule Hex-B2S23mp(2). Generation:1843 Population: 94.032

The simulations indicated when a branch tried to expand in a direction, it created new branches same as itself. Also, the rule does not affect the glider and the blinker forms. Discrete white points from the form represent producing the gliders. Moreover, it created more complex glider oscillators than before (Fig. 15).

The grey line in the black frame has been selected as a reference point to show the moving oscillator.

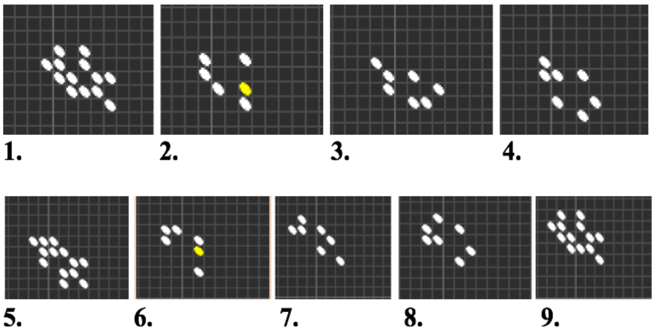

Fig. 15. A new glider.

In Fig. 15, the white cells represent alive cell due to 2-states cellular automata. Conversely, Fig. symbolise yellow cells as “alive cells” which has 2 points, and the white cells as “sick cells” owing to 3-states. The names’ purpose is to determine that the states are in the range of 0-2. The cells had to be less than alive and more than death. By selecting this name, we wanted to clarify relations between the three states CA. Consequently, sick cells are just symbolic names, not related to any sickness.

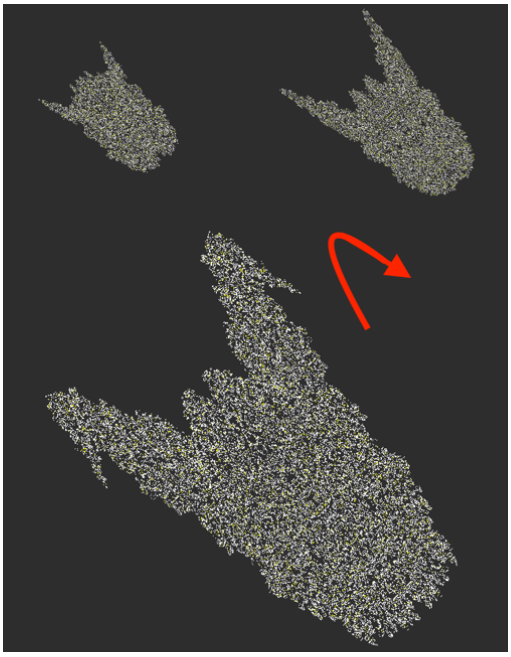

Fig. 16. Turning an arm branching. Generation: 517, 758, 968. Population: 12.237, 18.345, 29.596

While observing simulations, some branches were applying turning another side. The branching was growing up regularly subsequently changed its rotation and turned right (Fig. 16). The red line in Fig. 16 symbolises the branch turning and its direction. Due to the similarity of a “mantis arm” and the branching form, we labelled it as a “mantis”.

There are many life-like forms belong this rule. For example, Fig. 17 is one of them that might be called a six-legged starfish. Another one is Fig. 18 that can be compared a butterfly because of the similarity.

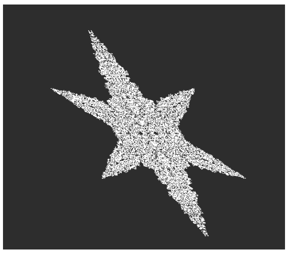

Fig. 17. A six-legged starfish life-like form.

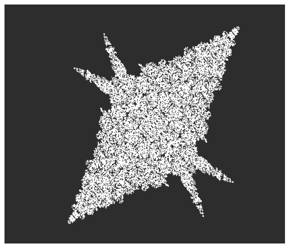

Fig. 18. A butterfly life-like form.

The branching attempts shaped to fractals forms that create similar branches same as its shape with a symmetry. We encountered often with these forms. In Fig. 19, we indicated a simple fractal form and its initial state consists of six sick and one alive cells. The large fractal structure grew up to south east that was similar to its initial state.

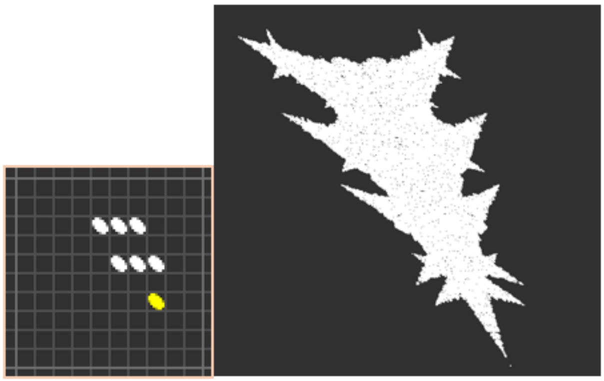

Fig. 19. A simple fractal form’s initial state and form. Generations: 1, 2057 Populations: 7, 278.193

F. Self-maintenance

Besides all these life-like forms, we decided to indicate self-maintenance in cellular automata. To show this, we removed random 112-square cells; each square includes 100 rectangular grids, from structures for repairing itself. The patterns that belong to these three rules were tried to repair the gap of this 112-square.

The honeycomb was chosen to show this (Fig. 20), and it was a clear example to show self-maintenance attempt for the pattern has a large “still life” structure in the middle [10], that means the pattern does not change from one generation to next, but unlike the middle of the pattern, the surface has been oscillating and producing cells. Therefore, when we removed cells out from the pattern, the simulation has been appearing more visible. The other rules, Hex-B2S23mp and Hex-B2S23mp(2), had similar results.

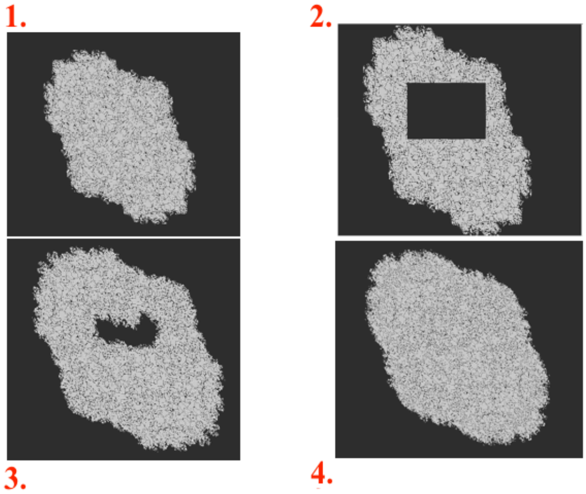

Fig. 20. An example of self-maintenance: The honeycomb.

G. *Self-replication*

For discussing self-replication, we needed to change the rule over again after many experiments as the previous rules did not copy itself exactly. After examined several various rules, one of them attempted to copy itself (Fig. 21). The hexagons in the Fig. 21 are not similar to the previous forms; they resemble a better-organised structure. The fourth rule is "Hex-B2(2)S3mp":

- If there are two any neighbours, and centre is a dead cell, an alive cell occurs.
- If there is a sick cell that has three meta-para neighbours or points, it survives as a sick cell.

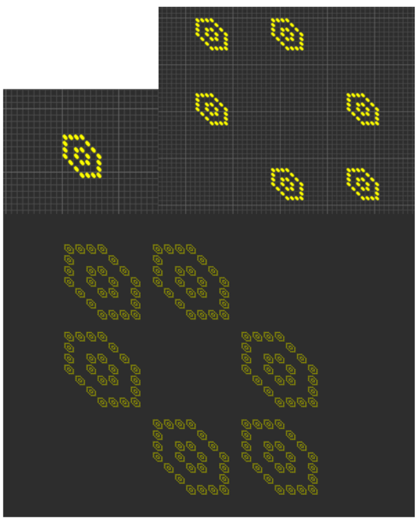

Fig. 21. An example of self-replication attempt. Generations: 3, 11, 365

- Otherwise, all the exceptions result in the death.

The rule could be called "Hex-B2(2)" or "Hex-B2" without mp neighbours or 3-states; however, we preferred to show this as "Hex-B2(2)S3mp". In this manner, it contributed us to illustrate with different colour and slow down the simulation speed somewhat that enabled it more observable in the first generations. Consequently, the meta-para neighbourhood condition is non-functional.

Unlike the generations in Fig. 21, intermediate generations presented an integration. After integration, they continued to copy itself.

III. Conclusion

In chapter A, we achieved hexagonal neighbourhood in rectangular grids. Using hexagonal neighbourhood enabled us to operate hexagonal cellular automaton. In chapter C, Comparable algorithms in hexagonal CA illustrated that we needed to work with a growing algorithm to reach large populations. While simulating random patterns of growing algorithms, we observed some glider and blinker structures at section D and section E. Therefore, hexagonal cellular automata allowed creating glider and blinker oscillators by confirming the first hypothesis.

The second simulations demonstrated that glider could contribute experiments and lead to a self-reproduction with a glider-glider collision in another layer. The patterns had similar population rates and disappearance at the same generation. However, the main pattern could not achieve self-reproduction in the same layer since the large cell groups were not able to move to other grids that caused a collision between the main pattern and the produced pattern. As a result, the second hypothesis self-reproduction was partially confirmed.

3-states hexagonal cellular automata have been implemented successfully. The third rule enhanced the complexity of the structures by generating fractal-like and branching forms. Some of the branching forms indicated that could be changed its route when growing up. That contributed to increasing the complexity. Also, the patterns were similar to life-like structures. The second hypothesis, increasing complexity, was confirmed.

The first rule the honeycomb realised a self-maintenance evidently. By repairing itself, it confirmed the third hypothesis self-maintenance.

The last rule created the hexagonal fractal-like patterns which grow up instantly. The patterns were able to copy itself; however, occasionally they were gathered and later, they re-copied themselves. Consequently, the last hypothesis self-replication was partially confirmed.

All these four rules emphasised the main hypothesis; the complex structures such as life-like forms not only do consist of complicated rules, but they also can come from simple rules.

IV. Discussion

In literature, we could not see many examples of hexagonal cellular automata with complexity in comparison the game of life rule. With this paper, we could show that the complexity can be reached with hexagonal neighbourhoods. Particularly large populations were not elaborated sufficiently in the literature due to its uncontrollable structure. We tried to encourage other researchers to control patterns further.

With these experiments, we can infer that the complexity of formations in nature may be not as complicated as we thought. Also, it is fair to say that the complexity of shapes can be modelled with a software. Some unusual or symmetrical forms in nature always seem intriguing and attract scientists to research. Therefore, this paper contributes modelling complexity for inspiring other scientists in different departments. Morphology science department that including astronomy, geomorphology and especially biology can be a clear example of this.

The focus of this paper was modelling in a virtue world rather than the feasible examples such as John von Neumann's universal constructor. Because of the large populations in the proposed CA, it does not appear feasible for the universal constructor nowadays. Nevertheless, developing technology surprise us every single day. As a result, it may not seem feasible for today; however, we cannot say that it is going to be infeasible for years later.

The large populations were unpredictable that inconvenienced the experimental conditions. We tried to shaped growth with a full control. Even though the formations might be controlled somewhat such as growth direction, it is difficult to manage large populations. We had to change the initial states for every experiment just as the trial-and-error method to achieve not an uneventful result. An automatic run and control software with an image recognition algorithm can solve this problem. The other limitation was about growing structures which prevent self-production or self-replication owing to the collision. Modifying rules with a "moving cells if conditions", that moves some cells which less than the certain population, can overcome this difficulty.

We were expecting to discover a glider-gun that would bring experiments a further dimension. However, gliders-blinkers collisions did not result as we expected. Some collisions grew up, some of them vanished. Nevertheless, we cannot say that glider-gun cannot be found for the proposed glider producing rules. The other point that we expected is a rule that throws glider frequently. In this way, we would have observed more patterns. In our examples, we could not observe frequent glider throwing instead of rarely spreading.

These algorithms have various way to improve next level. One of them may be a continuous feature instead of discrete CA. That may contribute the other departments to modelling their patterns or experiments for the feasibility. The other method might be a stochastic CA. Even though it slows down the simulation process, it provides many opportunities for new patterns. For instance, if stochastic rules together with moving cells are elaborated enough, a mitotic division can be observed. Also, appending or modifying states may alter experiments' way as we did.

Acknowledgment

We thank Dave Greene for assistance with the usage of Golly software.

REFERENCES

[1] C. G. Langton and S. Wilson, “Artificial Life,” *Artif. Life Proc. an Interdiscip. Work. Synth. Simul. Living Syst. Los Alamos, 1987*, vol. 24, no. 2, p. 244, 1989.

[2] M. Gardner, “Mathematical games: The fantastic combinations of John Conway’s new solitaire game ‘life,’” *Sci. Am.*, vol. 223, no. October, p. 120–123., 1970.

[3] S. Wolfram, “Statistical mechanics of cellular automata,” *Rev. Mod. Phys.*, vol. 55, no. 3, pp. 601–644, Jul. 1983.

[4] G. N. Reeke, “Book Review: A NEW KIND OF SCIENCE BY STEPHEN WOLFRAM,” *Int. J. Comput. Intell. Appl.*, vol. 4, no. 1, pp. 109–114, 2004.

[5] H. P. Edmundson, “Theory of self-reproducing automata,” *Inf. Storage Retr.*, vol. 5, no. 3, p. 151, 1969.

[6] C. L. Nehaniv, N. Oros, and A. R. Yinusa, “Self-Reproducing Automata.", lecture 7COM32 "Artificial life with Robotics", University of Hertfordshire, 2017.

[7] P. Callahan, "Experiments with a somewhat "Life-like" hexagonal CA" (Online). Avaliable: www.radicaleye.com/lifepage/hexrule.txt 2017. Accessed: 23-Nov-2017].

[8] A. Wuensche, “Glider dynamics in 3-value hexagonal cellular automata: the beehive rule,” 2005.

[9] R. W. Gosper, “Exploiting regularities in large cellular spaces,” *Phys. D Nonlinear Phenom.*, vol. 10, no. 1–2, pp. 75–80, Jan. 1984.

[10] “Still Life -- from Eric Weisstein’s Treasure Trove of Life C.A.” [Online]. Available: http://www.ericweisstein.com/encyclopedias/life/StillLife.html. [Accessed: 23-Nov-2017].