LIFE IS UNIVERSAL!

Robert T. Wainwright

CIBA-GEIGY Corporation

Abstract

The game of Life[1] involves forms built out of simple birth and death rules which a computer puts through a series of rapid transformations. This game was invented by John Horton Conway and recently introduced in Scientific American by Martin Gardner. Many computers have been programmed to play the game of Life. In this paper we shall show how to return the compliment by making Life forms that can imitate computers. Then we shall see that many remarkable consequences follow from the existence of such constructions. Further we shall see that in Life there exists the possibility of organisms with the ability to duplicate themselves, to reproduce. It has even been suggested that the universe itself is space-time granular and that the future although completely deterministic is unpredictable, being its own fastest simulation.

1. Introduction

Life[1] is based on cellular automata theory, a relatively unknown branch of mathematics, which is fast gaining exposure and prominence. Although this subject has been known for about 20 years it was only just recently, when John Horton Conway invented the game of Life based on this theory that cellular automata really gained popularity. In fact, extensions to this theory and also information theory were possible as a result of Life. Interestingly, these developments were made not on a theoretical basis but from empirical observations.

Unfortunately Conway has never published anything about the game of Life. The only references to his work were those originally given in Martin Gardner's Mathematical Games Section of the October 1970 and February 1971 issues of Scientific American. Readers interested in further reading on cellular automata theory and related subjects may find the bibliography given at the end of this paper helpful.

Cellular automata theory defines a universal cellular space in which an n-dimensional space is filled with regular n-dimensional polyhedrons. Sometimes this is referred to as a 'tesselation'. This simply means that our universe of concern is divided into unit cells which are identical. For instance, in three dimensions, the space may be divided into cubical cells. In two dimensions, the space is a plane and may be divided in any of the ways shown in Figure 1. Each cell within this space can itself have any number of different states including an empty (or quiescent) state. In two dimensions, four different states, for example, would be represented as shown in Figure 2.

Each cell also has a set of neighbor cells that can influence its state. Any pattern of neighbors may be defined such as shown in Figure 3.

Having chosen the basic parameters defining the physical cellular space we now introduce the element of time and the concept of change or evolution.

A cell's state will change according to a set of transition rules that apply simultaneously to every cell in the space. The transition rules are a function of both the current state of the cell and also the collective state of its neighbors. Any given pattern of cells will therefore change its state in discrete steps by recursively applying the transition rules.

Conway's game of Life is based on a space made up of squares just like an infinite sheet of graph paper. Each square or cell, as we shall call it, has only two possible states: either empty or full (occupied with a 'bit'). These two states can be thought of as off and on or as zero and one. The

[1] Throughout this paper, the game of Life will simply be referred to as Life. Association (of Life) with life in the sense of the real universe is not meant to be implied.

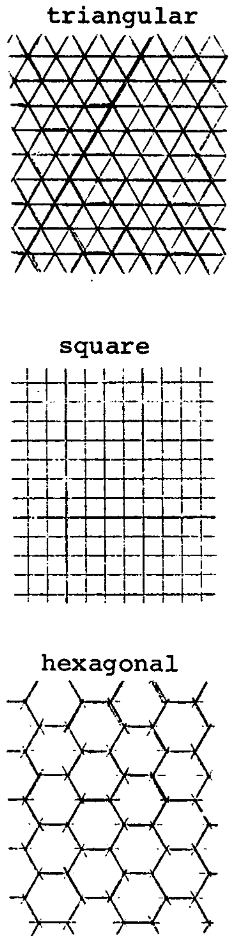

Figure 1.
Several planer tesselations.

Figure 2. A four-state system.

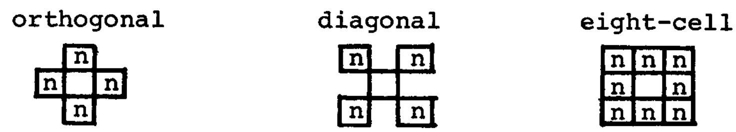

Figure 3. Several types of neighborhoods.

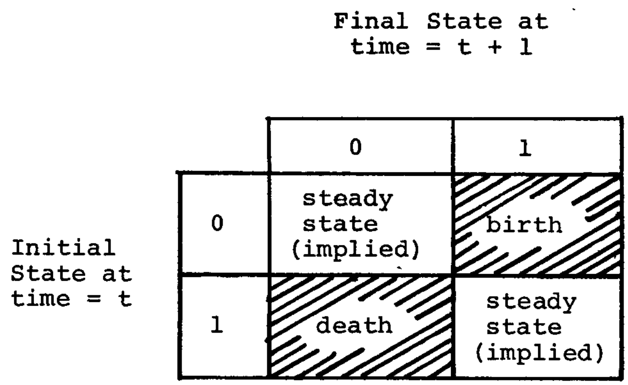

Figure 4. The possible transitions in a two-state system.

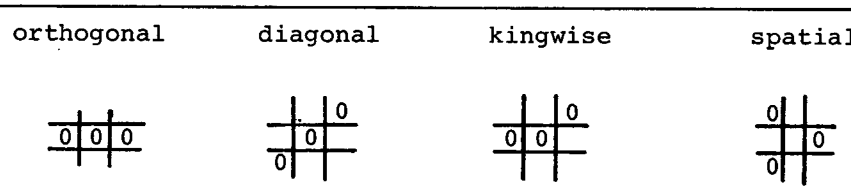

Figure 5. Types of initial pattern structure.

neighborhood consists of the eight squares immediately surrounding a cell (See Figure 3). With only two possible states, the rules for transition need only consider and be defined for a cell going from empty to occupied (birth) or from occupied to empty (death). These possible combinations of state change are shown in Figure 4. Notice that we need only define the combinations involving a change of state (shaded areas) and that the others are implied by whatever transition rules we establish for the former.

Initial patterns may be constructed with any desired arrangement of bits (i.e., occupied cells). The bits in the initial pattern may be connected in a rookwise fashion, a bishopwise fashion, kingwise fashion (combination), or not even connected at all. Examples of these are shown in Figure 5.

By applying Conway's special transition rules (which we have not yet defined) to an initial pattern of bits we will observe a change in the state of the entire pattern. This can be done recursively to determine the evolution of the population of bits.

2. Life's Rules

We have now briefly discussed cellular automata theory in general and specified the concepts of a square grid, two states, and the eight-cell neighborhood for Life. Still to be defined are the special set of recursive rules which make Life so interesting and unique.

Conway took several years to select his rules with great care to avoid two extremes: patterns that grow too quickly without limit and patterns where many would fade away. By striking a delicate balance he constructed a model of surprising unpredictability and one that produces an incredible variety of activity.

Conway chose his transition rules to meet these general desiderata:

i. There should be no initial pattern for which there is a simple proof that the population can grow without limit.

ii. There should be initial patterns that apparently do grow without limit.

iii. There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in three possible ways: fading away completely (from overcrowding or from becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering into an oscillating phase which repeats an endless cycle of two or more periods.

In brief, the rules should be such as to make the behavior of the population unpredictable. Conway's genetic laws are delightfully simple. Recall that each cell has exactly eight neighboring cells; four adjacent orthogonally and four adjacent diagonally.

The rules are:

BIRTHS: each empty cell with exactly 3 neighbors whose cells are full (contain a bit) is a birth cell. A bit is placed in it for the next move.

DEATHS: each full cell (containing a bit) with 4 or more neighbors dies from overpopulation. Every full cell with 1 or no neighbors dies from isolation. The bit is removed from it for the next move. In other words, every bit with 2 or 3 neighbors survives (remains) for the next move.

When Conway first stated his rules, he presented a third law for survivals but here we have included it with the death rule since survivals are implied. It is very important to understand that all births and deaths occur simultaneously. Together they constitute a single move, or as we shall call it, a 'generation', in the complete Life history of an initial configuration.

Note that we are free to chose any pattern desired for this beginning or zero generation. After this, the pattern is governed by Conway's rules. We will find the population constantly undergoing unusual, sometimes beautiful and always unexpected change. In a few cases the 'society' eventually dies out (all bits vanishing), although this may not happen until after many generations. Most starting patterns do not die out but become either stable figures - Conway calls them 'still lifes' - that cannot change or patterns that oscillate forever. Patterns with no initial symmetry tend to become symmetrical. Once this happens the symmetry can never be lost, although it may increase in richness.

3. Life's Creations

Let us see what happens to a variety of simple initial patterns.

A single bit or any pair of bits, wherever placed, will obviously die on the first move.

A beginning pattern of three bits also dies unless at least one bit has two neighbors. Figure 6 shows the triplets that do not fade on the first move. Their orientation is of course irrelevant. The first three (a,b,c) die on the second move. In connection with (c) it is worth noting that single diagonal chain of bits, however long, loses its two end bits on each move until the chain finally disappears.

The last two (d and e) are the trominoes (three rookwise-connected bits). Pattern (d) becomes a stable figure which Conway calls a 'block' on the second move. Notice that there are no empty cells with three full neighbors for birth and that all of the occupied cells each have three neighbors and hence survive. The initial pattern (e) is unusual for it evolves into a figure somewhat different than the others- it forever changes between two individual phases! This particular pattern will oscillate with a period of two alternating between horizontal and vertical rows of three. Conway calls this a 'blinker'.

Figure 7 shows the Life histories of the five tetrominoes (four rookwise-connected bits). The block (a) is, as we have seen,

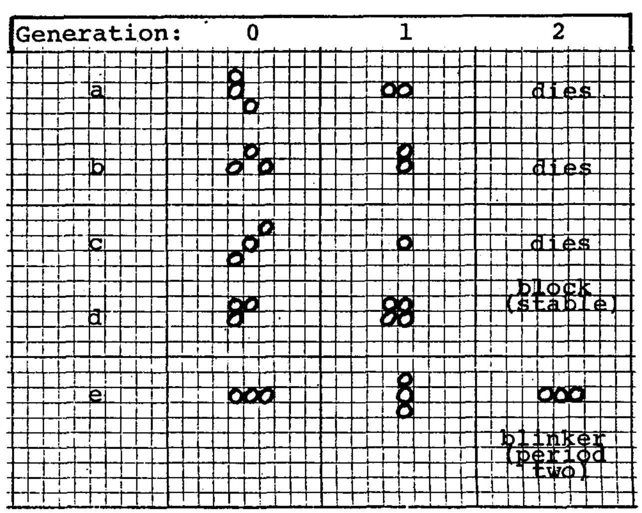

Figure 6. Fates of the triplet patterns.

a still life figure. Tetrominoes (b,c and d) all eventually reach a stable figure called the 'beehive' due to its hexagonal resemblance to the familiar honeycell. Conway noted that these two still lifes were the most frequently produced Life objects. Figure 8 shows some more common forms of still life. Tetromino (e) is the most interesting of the group. After nine moves it becomes four separate blinkers, an oscillator called 'traffic lights'. It too, is a fairly common configuration.

One of the most remarkable of Conway's discoveries is the five-bit 'glider' (his term) shown in Figure 9. After two moves it has shifted and been reflected in a diagonal line. Geometers call this a glide reflection, hence the object's name. After two more moves the glider has righted itself and moved one cell down and to the right from its initial position!

The speed a chess king moves in any direction is called by Conway the 'speed of light'. Conway chose that phrase because it is the maximum speed at which any kind of movement can occur on the grid. We say, therefore, that a diagonal chain decays at each end with the speed of light. Since the glider replicates itself in the same orientation after four moves, and has traveled one cell diagonally, one says that it glides across the field at a fourth the speed of light. Conway has proved that the maximum speed is a fourth the speed of light for a finite object moving diagonally. Movement of a finite object orthogonally (horizontally or vertically) into empty space, Conway has also shown, cannot exceed half the speed of light. Objects that move in any direction at any speed are extremely hard to find.

In examining the twelve pentominoes (all patterns of five rookwise-connected bits) we find that five die before the fifth move, two quickly reach a stable pattern of seven bits (Figure 8) and four in a short time become traffic lights. The only pentomino that does not end quickly (by vanishing, becoming stable or oscillating) is the R-pentomino (Figure 10). When Martin Gardner first presented Life in the October 1970 issue of Scientific American, the fate of this particular object was not known - even to Conway.

Gardner asked for help from his readers to settle this unsolved problem. Many readers set about writing computer programs to simulate Life's rules using this seemingly unobtrusive five-bit object as the initial pattern. Even with this additional help, only a few readers were able to successfully track the R-pentomino to its eventual fate. The confirmation, made by several readers, established that the R-pentomino reaches

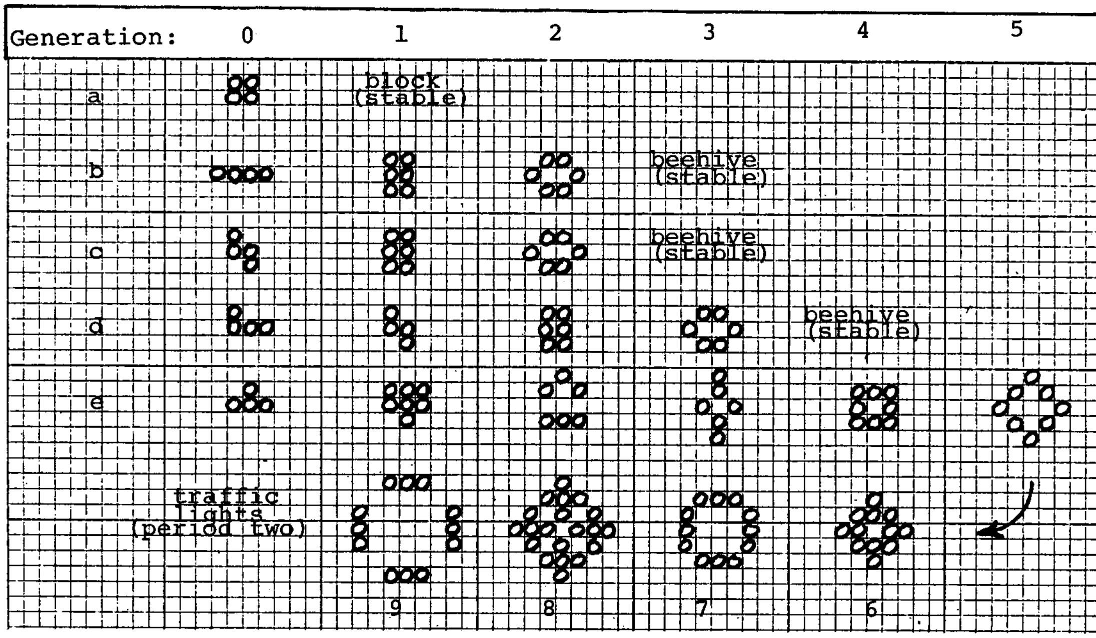

Figure 7. Life histories of the five tetrominoes

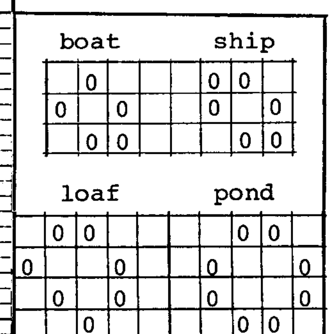

Figure 8. Some more common still life forms.

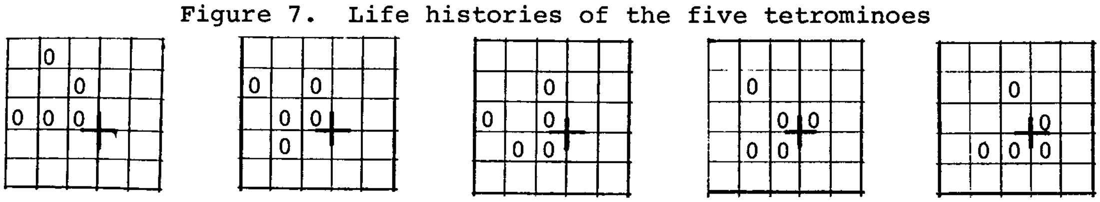

Figure 9. The remarkable glider

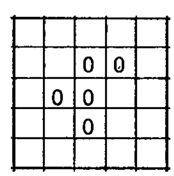

Figure 10. The R-pentomino

a 'steady state' only after an astonishing 1,103 generations! By then six gliders have been produced and are traveling outward at a fourth the speed of light in thee different directions. The constellatin (stable debris left in the center) consists of eight blocks, one boat, four beehives, one ship, one loaf, and four blinkers.

4. Conway's Conjecture

It thus appears that Conway did indeed establish a set of rules that meet his original desiderata. The R-pentomino was certainly unpredictable in its outcome, relative to the activity of other similar sized objects and especially to many who have attempted tracking it including Conway and also, it apparently did seemingly show unlimited growth. Conway felt so strongly about his first desiderata that he put forth a conjecture which Martin Gardner presented in October 1970. In addition, he offered a $50 prize to anyone who, by the end of that year, could either prove or disprove it. More will now be said about this original conjecture.

Conway was fully aware of earlier games and it was with them in mind when he selected his recursive rules. By striking a delicate balance he designed a game of surprising unpredictability and one that produced such remarkable figures as oscillators and moving gliders. He conjectured that no finite population could grow (in number of members) without limit. This was the deepest and most difficult question posed by the game when first presented in Gardner's first column. One way to disprove it would be to discover a configuration that continuously adds bits to the field. For instance, a 'gun' that repeatedly shoots out moving objects such as the glider or a 'puffer train' that moves while leaving behind a trail of 'smoke'.

The prize was won in November 1970 by a group headed up by R. Wm. Gosper, Jr. in the Artificial Intelligence Lab at M.I.T. Gosper made a truly astounding discovery; he was able to construct an oscillating configuration that every 30 generations would create a glider! (See Figure 11). Gosper's 'glider gun' forms the basis of much of what will now be discussed.

5. Glider Guns and Engagements

With gliders being so plentiful via the gun it was only natural that 'Lifenthusiasts' would place gliders on various collision paths to see what happens. As it turned out, experimentors found that two gliders oriented in different ways would, upon colliding, create different objects (stable and oscillating).

Additionally, in many cases it was possible to destroy an object by firing a prepositioned glider at it. (See Figure 12). An exception involves the stable 'eater' shown in Figure 13. This uncommon object has the unique ability of destroying (i.e., eating) gliders fired at it from several different directions. This unusual property will be useful in some constructions to be described later. Furthermore in a few cases it was found possible to actually shift or move an object that had previously been formed by striking it with a well aimed glider. With these glider mechanics now available as building blocks we can construct just about anything, even the gun itself! The sequence of steps involved in creating the gun is illustrated in Figure 14.

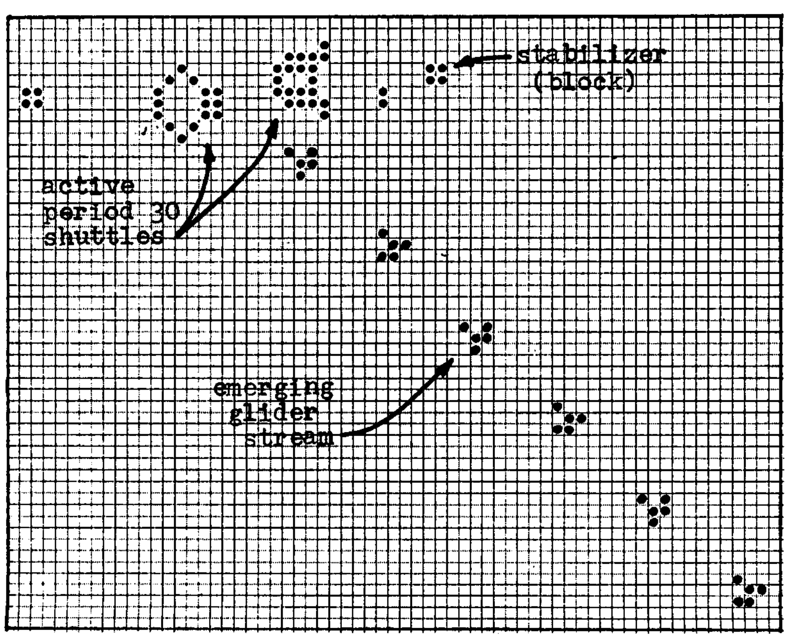

Figure 11. The glider gun.

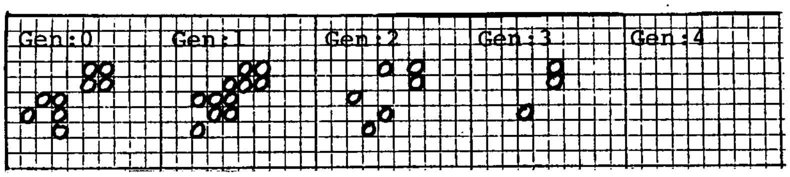

Figure 12. Annihilation of a block.

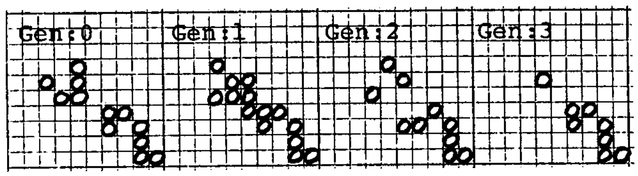

Figure 13. The curious 'eater'

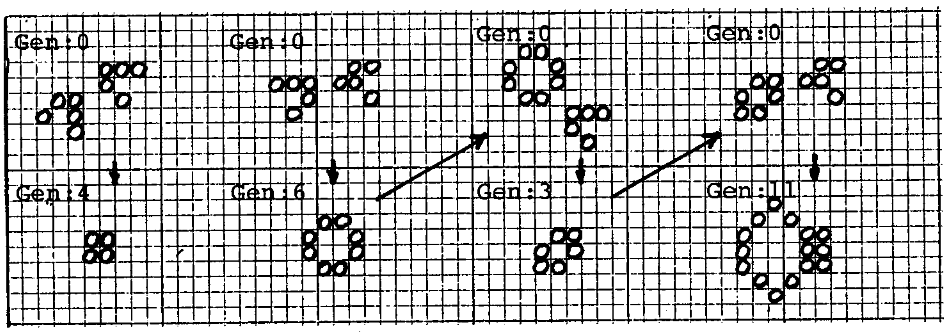

Figure 14. Construction of a glider gun

6. Making a Life Computer

Many computers have been programmed to play the game of Life. We shall now show how to return the compliment by making Life patterns that can imitate computers. Then we shall see that many remarkable consequences follow from the existence of such constructions.

a. components

Good old-fashioned computers are made of pieces of wire, along which pulses of electricity travel. Somewhere in the computer is a mechanism called a clock pulse generator, which generates pulses at regular intervals, and most of the working parts of the machine are made up of logical gates, like those drawn in Figure 15.

These gates act as follows. The OR gate has two input wires (A,B) and one output wire, which emits a pulse only when the gate has just received a pulse from either the A or the B input. The AND gate is similar except that a pulse is emitted only when the gate has just received pulses at both the A and B inputs. A NOT gate has just one input, and emits a pulse only if it has not received one. We shall assume that a competent engineer can build a computer if he is given enough of these pieces.

Our problem is to design a Life configuration which will act just like a computer. Certain ideas occur immediately. Rather than using electrical pulses, we can use gliders. Then we can do without wires - in effect the wires are the particular paths in the plane along which we intend to send the gliders. For the clock pulse generator we will use a glider gun. The remaining problems are simply to design the various pieces we shall need, and work out ways in which they can be 'wired together'.

With these ideas in mind, it is natural to study all the possible interactions of two gliders which meet at right angles to see

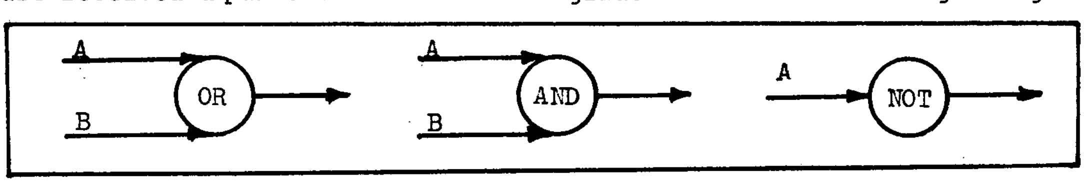

Figure 15. Logical gates.

what use we can make of them. As we have already mentioned, there are many possibilities according to the exact arrangement and timing of the gliders.

In many cases, the reaction is at first sight rather disappointing - the gliders collide to form an unstable pattern which then fades away completely. These are the vanish reactions, and they turn out to be surprisingly useful. We shall need to know only that there are several such reactions with differing positions and timings, and in which the decay is so fast that later gliders in the same oncoming stream will not be affected. In our figures, a star (see Figure 16) indicates a vanish reaction, and an arrow denotes a glider or glider-stream.

Another useful reaction is the kick-back reaction, in which the decay product is another glider travelling backwards in a direction parallel to one of the two original gliders, which we think of as having been kicked back by the other. This is indicated as in Figure 16.

Almost all the working parts of our computer are made up by combining glider-streams with vanish and kick-back reactions! The only static parts will be glider guns and eaters, represented in the figures by the letters G and E. Remember that instead of wires we have paths along which gliders may or may not be travelling in certain places - when one of these places does not contain a glider we call it a hole. In this way we can represent an information stream in binary form.

b. time and spatial control

Glider-streams as they emerge from normal guns are so dense that it is impossible for two to interpenetrate without interfering. If our computer were to use streams at this density, we would not be able to draw its wiring-diagram in a plane with no wires ever being allowed to cross. Therefore we had better find some way to reduce the pulse-rate. A neat way is to use two kick-back reactions as in Figure 17. For clarity, the rest of these figures have been rotated 45 degrees to show the streams in horizontal and vertical positions.

Here guns G_1 and G_2 fire in parallel but opposite directions, producing normal glider-streams. But there is a glider g which will travel west until at a it is kicked east by a glider from the G_1 stream. The timing and phasing is such that at b it will be kicked back towards a again, so it repeatedly "loops the loop", removing one glider from each of the two streams per cycle. After this, every Nth glider (say) is missing from each of these streams. We don't want the G_1-stream, and so feed it into an eater, but we feed the G_2-stream into a vanish reaction with the stream from a third gun G_3. Every glider from G_2 now dies, but every Nth glider from G_3 escapes through one of the holes in the G_2-stream! So the whole pattern acts as a "thin gun", producing just one Nth as many gliders as the normal gun.

It turns out that to get the phasing right N must divide exactly by 4, but it can be arbitrarily large, and so we can make an arbitrarily thin stream. It is possible for two such streams to cross without interacting, as in the right hand part of the figure, provided things are properly timed. So from now on, we can use the work 'gun' to mean an arbitrarily thin guns - perhaps we would need a thinning factor of 1000 for all of our design to work.

Figure 18 shows how a glider stream can be repositioned accurately for its next intended use. We simply feed it through a number of guns (here 4), which fire at the same rate as the stream. Of course not every place in the input stream will be filled by a glider - it really consists of gliders mingled with holes. Any glider in the input kills a glider from the first gun G_1, yielding a hole in the emerging stream, while each hole in the original stream permits a glider from G_1 to pass unmolested. The effect is that the stream is turned through a right angle and complemented - gliders being interchanged with holes.

After an even number of such reactions, the glider stream emerges unharmed, but delayed,

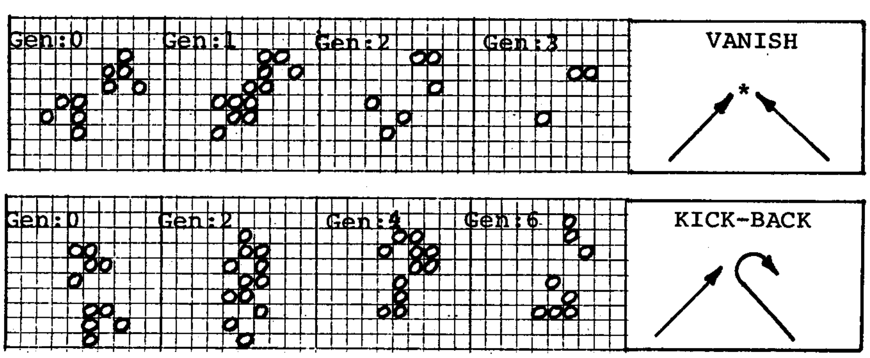

Figure 16. The 'vanish' and 'kick-back' reactions.

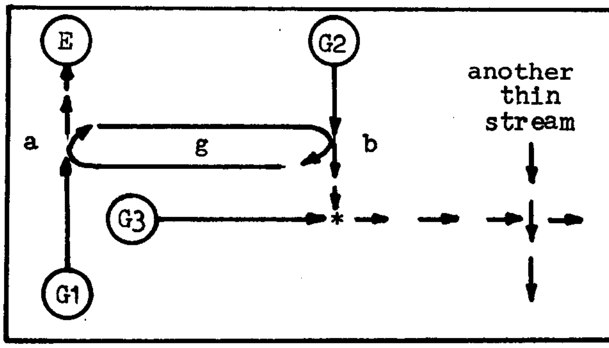

Figure 17. Thinning a glider stream.

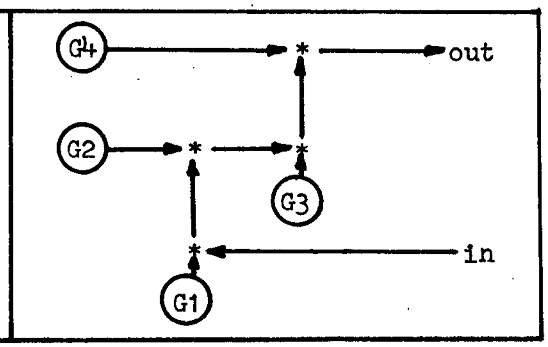

Figure 18. Delaying a glider stream.

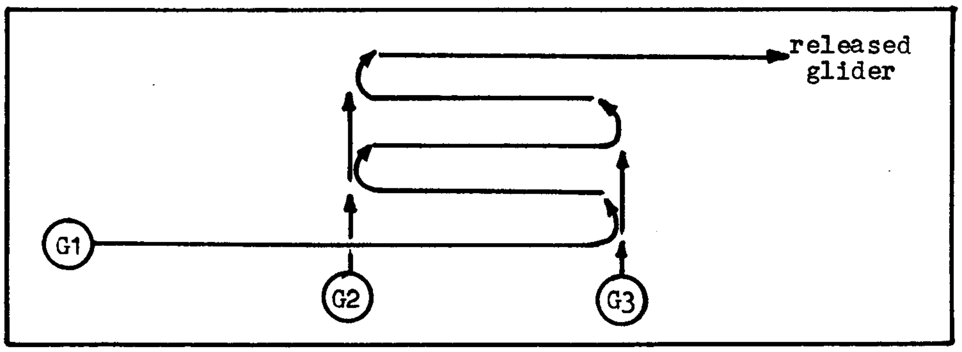

Figure 19. Moving a glider stream.

and travelling in direction either parallel or opposite to its original one. Since there are plenty of available vanish reactions, we can control the delay. Although some delay is inevitable, we can adjust it arbitrarily. In the same way, we can use the various reactions to locate the output stream where we want it.

How exactly do we control the spatial location of all the gliders for these manipulations? A mechanism can be designed, so as to produce gliders where and when we want them. The only restrictions are that they be produced far apart. (See Figure 19). A glider is initially shot from a gun G_1. At the right moment, a second gun G_2 is ordered to fire, followed some time later by another one G_3, and then G_2 and G_3 fire alternately for some time.

The glider from G_1 is kicked back by the first glider from G_3, then kicked again by the first glider from G_2, back again by the second from G_3, and so on. In the process, it is gradually 'blown' away. Just when it gets to the right place, G_2 and G_3 stop firing, and the glider from G_1 escapes.

Many gliders can be sent along closely parallel paths in the same sort of way. We can even arrange to use the same three guns G_1, G_2, G_3 (controlled differently) to send all these gliders. Later on, we shall need this fact that many gliders can be sent to controlled positions in space using only a fixed number of guns, in our carefully designed computer.

c. logical gates and stream duplication

In Figure 20 we show how to build the logical gates using only vanish reactions. In the case of AND and OR gates, we are forced to make the output streams parallel to the input streams, while for the NOT gate the output is necessarily at right angles to the input. If we had an easy way to turn streams around corners without complementing them, or complement them without turning around corners, all would be well. Fortunately, the solution to another problem automatically solves this one.

If we want to use the information carried by a glider stream in several different places, we shall need several copies of the stream. This is the hardest problem which we still have to solve. To get some clues, we examine what happens when we use one glider to kick back the first glider from a gun-stream.

Suppose that this gun-stream produces a glider every N generations, and we call it the full stream. Now it turns out that if N is 120 (¼ gun density), then when we kick

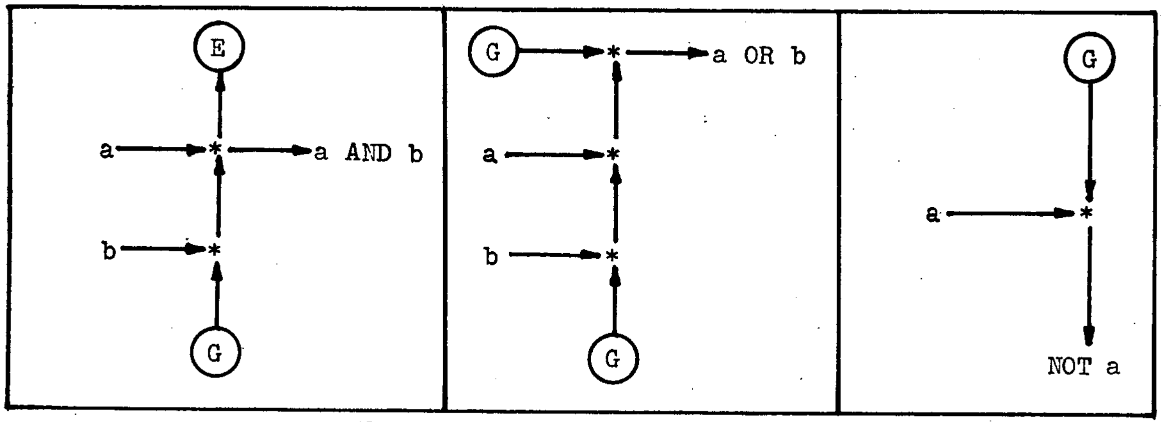

Figure 20. Life logical gates.

it with another glider, we remove just three gliders from the stream! This happens as follows:

1. The first glider (1) is kicked back along the full stream.
ii. The second glider (2) then fuses with (1) to form a block.
iii. The third glider (3) then annihilates this block.
iv. All subsequent gliders from the full stream escape unharmed.

We use this curious behaviour as follows. Suppose that our information carrying stream operates at 1/10 of the density of the full stream (say), so that the last 9 of every 10 places on it will be empty, while the first place might or might not be full. If we use 0 for a hole, and block the places in ten's, our stream therefore looks like:

... (000000000C) (000000000B) (000000000A)

We first feed it into an OR gate with a stream of the type:

... (00000000g0) (00000000g0) (00000000g0)

the g's denoting gliders that are definitely present. The result is a stream:

... (00000000gC) (00000000gB) (00000000gA)

in which every information-carrying place is definitely followed by a glider g.

This stream is used to kick a full stream whose gliders will be numbered for reference:

... (..........) (.........n) (10987654321)

If the glider A (or B or C) is present, it vanquishes full stream gliders 1,2,3, and the following glider g escapes in the confusion. But if A is absent, the full stream glider 1 escapes, and gliders 2,3,4 are removed instead by the following glider g. So the stream which emerges is definitely empty except for the second of every ten places, and these places carry the original version of the input stream (which has not been turned through a right angle). The other stream manages to carry the information from the input stream twice, in its first and fourth digits of each block, the first digit carrying the complement version. By feeding this stream into vanish reactions with other thin gun-streams, we can recover the original input stream either complemented or not, and freed from undesirable accompanying gliders! Figure 21 shows what happens to the four input blocks in such a system.

From here on, it is just a matter of engineering to construct an arbitrarily powerful (albeit slow) computer. Our engineer has been given the tools - let him finish the job! We know that such computers can be programmed to do many jobs. The most important jobs our computer will be required to do all involve emitting sequences of gliders at precisely controlled positions and times.

7. Nontrivial Self-replication

It is known that both eaters and guns can be constructed by crashing suitable patterns of gliders. Using this fact, and also some additional information about the directions in which these gliders travel, we can now show that it is possible to build a computer simply by crashing some enormously large initial pattern of gliders. Moreover, we can design a computer whose sole aim in life is to 'throw' just such a pattern of gliders into the air. In this way, one computer can give birth to another, which can be if we like an exact copy of the first. Alternatively, we could arrange that the first computer eliminate itself after giving birth, then we would regard the second as a reincarnation of the first. We conclude:

i. there are Life-patterns which behave like self-replicating organisms.
ii. there are Life patterns which move steadily in any desired direction, recovering their initial form after a number of generations.

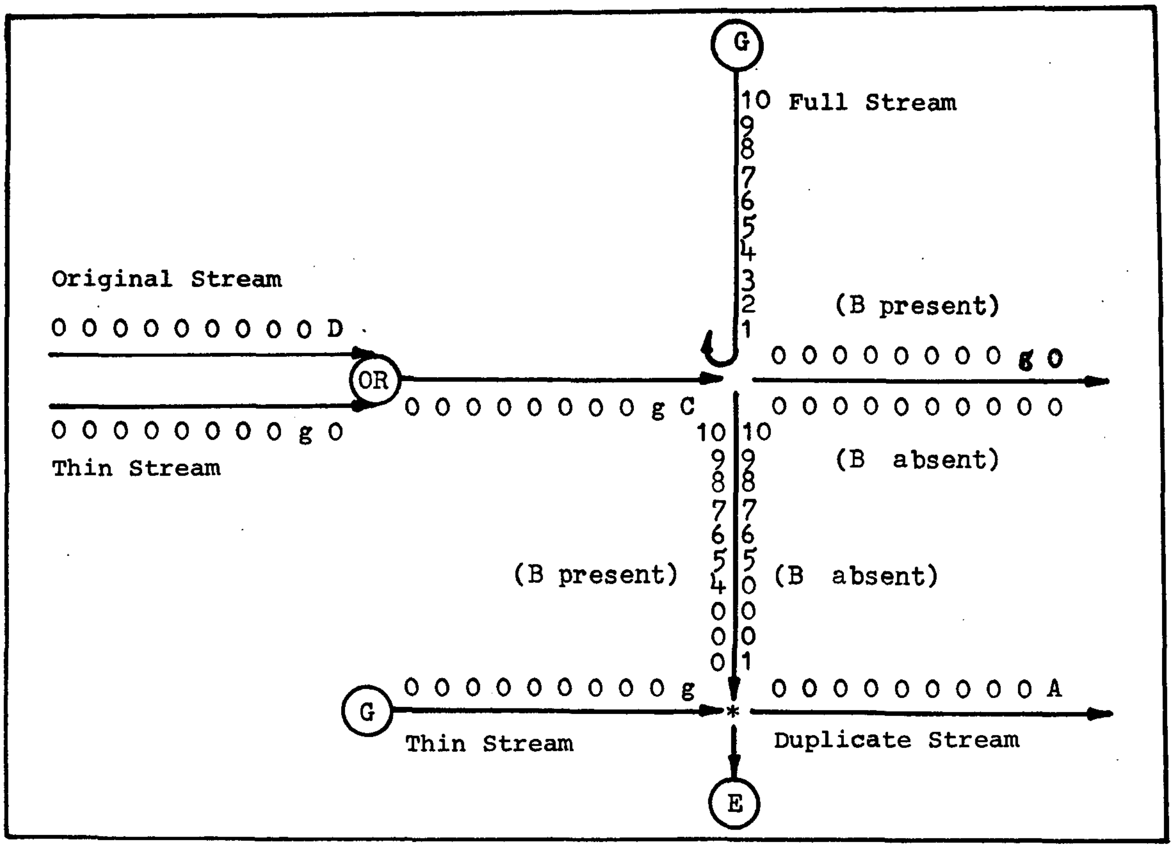

Figure 21. Duplication of an information stream.

We have now shown that among the finite Life patterns there is a very small proportion which behave like self-replicating organisms. Moreover, it is presumably possible to design such patterns which will survive in the typical Life environment (a sort of primordial 'broth' made of blocks, blinkers, gliders, etc.). It might for instance do this by shooting out masses of gliders to detect nearby objects and then take appropriate action to eliminate them. So one of these organisms could be more or less adjusted to its environment than another. If both were self-replicating, and shared a common territory, presumably more copies of the better adapted one would survive and replicate.

From here on we read a very familiar story. Inside any sufficiently large random broth, we expect just by chance, that there will be some of these self-replicating organisms or creatures! Any particularly well-adapted ones will gradually come to populate their territory. Sometimes one of the creatures will be accidentally modified by some unusual object which it was not programmed to avoid. Most of these modifications, or mutations, are likely to be harmful, and will adversely affect the organism's chances of survival, but very occasionally, there will be some beneficial mutations. In these cases, the modified organisms will slowly come to predominate in their territory, and so on. There does not seem to be any limit to this process of evolution, and so we conclude:

It is possible indeed, that given a sufficiently large Life space set initially into some random state, that over long periods of time, self-reproducing organisms will emerge and populate some parts of the space!

This is slightly more than just speculation, since the earlier parts are based on precisely proved theorems. Of course, 'sufficiently large' means very large indeed, and we cannot prove that 'living' organisms of any kind are likely to emerge in any Life space which we can construct in practice.

8. Some Speculation

It is remarkable how such a simple system of genetic rules can lead to such complex results. It may be argued that the small configurations so far looked at correspond roughly to the molecular level in the real universe. If a two-state cellular automaton can produce such varied and esoteric phenomena from these simple laws, how much more so our own universe?

Analogies with life processes are impossible to resist. If a primordial broth of amino acids is large enough, and there is sufficient time, self-replicating, moving automata may result from complex transition rules built into the structure of matter and laws of nature. There is even the possibility that space-time itself is granular, composed of discrete units, and that the universe, as Edward Fredkin of M.I.T. and others have suggested, is a cellular automaton run by an enormous computer. If so, what we call motion may be only simulated motion. A moving particle in the ultimate microlevel, may be essentially the same as one of Conway's gliders, appearing to move on the macrolevel whereas actually there is only an alteration of states of basic space-time cells in obedience to transition rules that have not yet been discovered.

Bibliography

Readers interested in further reading on cellular automata theory and related subjects may find these references helpful.

- Abt, Clark C., Serious Games: The Art and Science of Games that Simulate Life, The Viking Press, 1970.
- Arbib, M. A., "A Simple Self-Reproducing Universal Automaton", Information and Control, 1966.
- Banks, E. R., "Information Processing and Transmission in Cellular Automata", PH.D. thesis, Dept. Mech. Eng., MIT (1/15/71).
- Burks, A. W., Essays on Cellular Automata, Univ. of Ill. Press, 1970.
- Codd, E. F., Cellular Automata, Academic Press, Inc. New York and London, 1968.
- Gardner, Martin, "Mathematical Games", Scientific American, October 1970 February 1971.
- Golay, M. J. E., "Hexagonal parallel pattern transformations, IEE Trans. Computers, vol. c-18, Aug. 1969.
- Lee, Chester, "Synthesis of a Cellular Universal Machine using the 29-state Model of von Neumann", Automata Theory Notes, The University of Michigan Engineering Summer Conferences, 1964.
- Minsky, Marvin L., Computation: Finite and Infinite Machines, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.
- Moore, E. F., "Machine Models of Self-Reproduction", Proceedings of Symposia in Applied Mathematics, vol. XIV, American Mathematical Society, 1962, pp. 17-34.
- Shannon, C. E., "A Universal Turing Machine with Two Internal States", in Automata Studies (C. E. Shannon and J. McCarthy, Eds.) Princeton University Press, 1956.
- Smith, Alvy Ray III, "Cellular Automata Theory", Technical Report No. 2, Digital Systems Laboratory, Stanford Electronics Laboratories, Stanford University, 1969.
- Thatcher, J. W., "Universality in the von Neumann Cellular Model", Technical Report 03105-30-T, ORA, The University of Michigan, 1964.
- Turing, A. M., "Computing Machinery and Intelligence", Mind, vol. 59, pp. 433-460, 1950.
- Ulam, S. M., "A Collection of Mathematical Problems", Interscience, New York, 1969. See Sec. II.2: A Problem on Matrices Arising in the Theory of Automata.
- von Neumann, J., The Theory of Self-Reproducing Automata, edited by A. W. Burks, University of Illinois Press, Urbana and London (1966).
- Wainwright, R. T., editor and publisher of LIFELINE, quarterly newsletter (56 Old Highway, Wilton, Conn. 06897).
- Winograd, T. A. "A Simple Algorithm for Self-Replication", A. I. Memo 197, Project MAC, MIT, Cambridge, Mass. 02139 (1970).

ROBERT THOMAS WAINWRIGHT is Director of the CIBA-GEIGY corporate Operations Research Department in Ardsley, New York. CIBA-GEIGY Corporation is a leading producer of pharmaceuticals, agricultural chemicals, dyestuffs, and industrial chemicals.

Formerly with the Mobil Oil Corporation, he was responsible for overall direction and technical consultation to implement computer systems and management sciences techniques. A native of St. Louis, Missouri, Wainwright had previously worked at Standard Oil of Ohio, the National Aeronautics and Space Administration, and the DuPont Company.

He holds a bachelor's degree in chemical engineering from Purdue University, and master's degrees from New York University and Krannert School of Business. He is a member of the American Institute of Chemical Engineers and the Institute of Management Sciences.