An Implementation of von Neumann's Self-Reproducing Machine

Umberto Pesavento
Princeton University
Class of 2000
Princeton, NJ 08544
pesavent@intercity.shiny.it

Keywords
cellular automata, self-reproduction, universal constructor

Abstract This article describes in detail an implementation of John von Neumann's *self-reproducing machine.* Self-reproduction is achieved as a special case of construction by a *universal constructor.* The theoretical proof of the existence of such machines was given by John von Neumann in the early 1950s [6], but was first implemented in 1994, by the author in collaboration with R. Nobili. Our implementation relies on an extension of the state-transition rule of von Neumann's original cellular automaton. This extension was introduced to simplify the design of the constructor. The main operations in our constructor can be mapped into operations of von Neumann's machine.

1 Introduction

Von Neumann [6] introduced *constructive universality* in cellular automata to study the implementability of *self-reproducing machines* and to extend the concept of *computational universality,* introduced by A. Turing [5]. A computing machine is said to be computationally universal if it is capable of simulating any other computing machine; von Neumann's universal constructor is a machine capable of generating any other machine that can be embedded in its cellular automaton and described in a finite but arbitrarily long *tape.* Computational universality and constructive universality are conceptually related properties, but a machine does not need to possess a universal computer to be a universal constructor. Von Neumann, in fact, demonstrated that it is possible to implement a Turing machine in his cellular automaton, but pointed out that a Turing machine is not a necessary component of the universal constructor.

Von Neumann's conceptual extension is relevant from a bio-theoretical standpoint as it afforded for the first time the conditions necessary for a system to be capable of self-reproduction. However, due to the rigid determinism governing von Neumann's machines as well as their lack of fault tolerance, they are not good models of living beings.

Moreover, von Neumann left open the question of determining the minimal logical organization necessary for a system to exhibit self-reproduction properties. Langton [2,3] showed that self-reproduction is not necessarily a particular case of universal constructive capabilities.

Other works aimed to improve the results of von Neumann. Codd [1] reduced the number of cell states of the transition rule, but that resulted in an increase of the structural complexity of the constructor. Thatcher [4] completed the parts of the universal constructor that von Neumann had not finished in detail, simplifying some of the components outlined in the fifth chapter of von Neumann's original manuscript, summarized by Burks in [6].

Von Neumann defined a universal constructor to be a machine capable of (a) reading the description of an arbitrary quiescent cell assembly from a tape, (b) producing excitations in a field of cells in the blank state so as to build the described quiescent cell assembly, and finally, (c) activating that cell assembly by a starting excitation. The self-reproducing machine described in this article is obtained from a universal constructor by providing it with a description of the constructor itself and adding to it the capability of copying the description provided.

To obtain complete self-reproduction, the tape is interpreted in two different ways. In the first interpretation, the tape is simply copied, while in the second the tape is decoded to produce excitations corresponding to the cell assembly to be built. To implement these interpretations, our universal constructor possesses an organ for the representation of its internal state. This organ governs the changing interconnections between the other organs of the constructor.

Our constructor obeys an extension of von Neumann's original transition rule. The extended rule differs from the original only in the behavior of the so-called confluent element. This element has been redefined to direct signal crossing and provide a one-bit memory unit. Three additional excited states of the confluent element are introduced to direct signal crossing. However, as explained in Section 11, a different extension of von Neumann's transition rule can provide signal crossing without increasing the cell state number.

The input-output devices of our constructor are essentially the same as those used in von Neumann's original design and in Thatcher's work [4]. A more detailed description of them can be found in [6] in Sections 5.2.1 and 5.2.2.[1] Almost all the other areas and organs of the constructor, including the controls of the input-output devices, have been reimplemented or replaced by a more efficient and smaller apparatus. Nonetheless, the *encoders*, the *decoders*, and the *discriminator* follow von Neumann's original design. We have introduced new organs for the local direction of information traffic, for the memory operations and for the representation of the internal state of the constructor. These make the processing of the information collected from the tape much simpler than in von Neumann's original design. The main improvements of our design are (a) the substitution of the *coded-channel*, the device used by von Neumann to solve the problem of crossing transmission lines, with structures that use the new crossing cell states of the extended transition rule; (b) the substitution of the *static-dynamic converter* with five confluent elements that, exploiting the extended transition rule, work as memory units; and (c) the substitution of the state organs SO that formed the finite automaton FA and of the control organs CO with counters and other traffic control organs. As a consequence, the size of the universal constructor presented here is substantially smaller than that described in [6].

This article is organized as follows. After this introductory section, the major areas of the machine and their interactions are described. The details of the structure of these areas are presented in the next sections [Sections 5–8]. These details assembled, the dynamics of the construction process [section 10] are examined. The transition rule underlying this process is presented in Appendix A. As an aid to understanding the dynamics of the constructor, a sequence of interactive demonstrations has been made available at the Alife Online web site,[2] as described in Appendix B.

[1] Some minor errors in the patterns of excitations described by von Neumann have been corrected.

[2] URL: http://alife.santafe.edu/refereed/96/jvn

2 Areas and Organs of the Universal Constructor

The quiescent structure of the constructor is made of *transmission elements* and *confluent elements*. Each element can assume one or more excited states (*excitations*) that represent the information traveling in the constructor. Transmission elements are used to carry patterns of excitations in the constructor, while the confluent elements are used to process patterns of excitations.

The constructor is divided into four main areas, each with a distinct functional role, and containing several organs in close communication. These areas are (a) the *reading loop area*, (b) the *memory area*, (c) the *writing loop area*, and (d) the *state control area*. These four areas taken together form the universal constructor. These areas will be introduced, and then, in the next sections, their action described in full detail.

2.1 The Reading Loop Area

The task of the reading loop area is to read the genetic information stored in the tape, and then send the collected bits of information to the memory area. The reading loop area contains (a) an extensible reading loop, which enables the constructor to scan the tape; (b) encoders, which direct the action of the loop; and (c) a discriminator, which enables the constructor to distinguish between 0 and 1 bits.

2.2 The Memory Area

The memory area contains five confluent elements with a neighbor configuration such that these cells work as memory units. The memory area collects bits from the reading loop area and releases them in quintuplets of excitations. According to the constructor state, it may also interpret each single bit, converting it to a writing arm action special quintuplet.

2.3 The Writing Loop Area

The writing loop area directs the action of the *writing arm*, that is, a pair of extensible transmission lines that convey patterns of excitations to the cells in the blank area of the automaton. The cell assembly described in the tape is thus built in the blank area. The writing loop area decodes the quintuplets of excitations arriving from the memory area into writing arm movements, cell state transitions in the blank area of the automaton (*constructive processes*), or the release of a starting excitation. To achieve this, it contains a control apparatus that, in turn, is made up of two sets of decoders De_1 and De_2, and corresponding sets of encoders. The first decoder of De_1 is used to detect the 11011 state change quintuplets. All of the other decoders of De_1 and De_2, along with their associated encoders, direct the operations of the *writing head* (the terminal part of the writing arm).

2.4 The State Control Area

The state control area contains the *constructor state multiple counter* C_S. When a state change quintuplet is detected by the first decoder of the writing loop control, a single excitation is conveyed to the input point of this area. Such excitations increment the multiple counter C_S, which, in turn, sends various excitations to the traffic control organs in the other areas. This causes changes in the cabling of the constructor.

3 The Interactions Among the Four Main Areas

The reading loop area is the first to be activated by the starting excitation. This area, from the beginning to the end of the construction, continues working almost independently from the other areas of the constructor. It sequentially scans the tape and provides a constant stream of information to the rest of the constructor.

The two output lines of this area (b_0 for the excitations representing zeros and b_1 for those representing ones) are input lines of the memory area. This area, after having preprocessed the inputs (by quintuplets or by single bits depending on the constructor state), passes them to the control apparatus of the writing loop area. Here, the first decoder discriminates the 11011 state change quintuplets from those governing the writing arm. The detection of a 11011 quintuplet causes a single excitation to be conveyed to the input line of the state control area, while the other quintuplets are further processed by the same control apparatus and will result in writing arm actions.

The 11011 quintuplets recognized by the first decoder of the writing loop control and passed to the state control area have a direct feedback on all the other areas of the constructor. In fact, a number of transmission lines connect the state control area to the traffic control organs of the constructor. Through these lines the state control area directs the behavior of the others, switching, for example, the direction of the tape scanning, the modality of interpretation of the tape information, and the action of the writing arm.

Thus, there are two main information flows in the constructor: one goes from the reading loop area to the writing loop area and translates the information of the tape into the excitations of the blank area of the automaton corresponding to the cell assembly to be built; the other goes from the reading loop area to all the others, passing through the state control area, and directs the changes in the cabling of the constructor.

4 The Organization of the Tape Information

The tape is a particular kind of cell assembly, consisting of a sequence of cells in the quiescent ordinary downward transmission state (representing the 1s) and cells in the blank state (representing the 0s). The tape encodes the cell assembly to be built and its position. It also encodes the positions of the tape copy and of the starting excitation. It would be possible to design the universal constructor so that the cell assembly to be built and the tape copy could be located at any position in the cellular automaton. However, to keep the design of the universal constructor as simple as possible, the locations of the cell assembly to be built and of the tape copy will be restricted to the top right of the constructor.

The tape is a read-only device whose general organization reflects the various phases of the construction. It contains two main areas, separated from each other by a 11011 quintuplet: the *tape position area* and the *cell assembly position and structure area*. The starting excitation, if present, is contained at the end of the tape. Two 11011 quintuplets delimit the beginning and the end of the tape.

During the construction process, the tape will be scanned several times by the reading head: (a) to get the tape position; (b) to read through the whole tape to determine its size; (c) to copy the tape; and (d) to produce the cell assembly at the right position and to activate it with the starting excitation. These phases of tape scanning correspond to the main phases of the construction of the universal constructor.

5 The Reading Loop Area

The reading loop area is the first to be activated by the starting excitation, which arrives from point S. This excitation reaches directly point R and causes the first reading quintuplet 10101 to be generated and conveyed to the *reading head* H_R. When the reading quintuplet 10101 passes through the reading head, it activates the reading loop area and reads the first bit of the tape. The reading head communicates with the *discriminator* that receives either a 10101 quintuplet, when the reading head reads a 1 from the tape, or a single excitation, when it reads a 0. In turn, the discriminator sends out a single excitation to line b_0 or b_1 according to the pattern received (respectively, 1 or 10101). Because bit reading is a destructive operation (as it implies that the bit read is always replaced by a 1) the information coming from the discriminator is used also to restore the tape information. After restoring the bit read, the reading head always performs a one-step motion and reads another bit from the tape, until the end of the construction.

The reading loop area can perform five operations on the reading head H_R. Each operation is activated by a single excitation, coming at a different input point: at point R for the reading operation; at point L_1 for moving the head leftward and restoring a 1 on the current position; at L_0 for moving in the same direction and restoring a 0; and at R_1 and R_0 for moving rightward and restoring, respectively, a 1 and a 0 in the tape. While the tape is scanned bits are restored to the tape immediately after they are read. Effectively, the tape is a read-only device. Unlike the tape in a Turing machine, this tape is not used as a support for computation. The direction of motion of the reading head is determined by the constructor state. The transmission elements T_0 and T_1 will be oriented rightward during the advancement of the reading head and upward during its retreat. The orientation change of such transmission elements is achieved using a pair of switches governed by the state control area.

Motion of the reading head along the tape is achieved exploiting the constructive and destructive capabilities of the transition rule. Motions in different directions are obtained by the use of different sets of encoders that send several properly timed patterns of excitations to the two transmission lines forming the upper part of the reading loop. The interactions of these patterns of excitations result in the desired reading head motion. The reading loop area is represented in Example 10 of the interactive demo.

6 The Memory Area

To temporarily store the bits coming from the reading loop area and release them in quintuplets of excitations, the memory area possesses five confluent elements working as memory units. The transition rule ensures that an excited confluent element that does not have outgoing transmission elements in its neighborhood remains in its excited state. Therefore, the confluent elements $M_1, \ldots, M_5$, when excited by the bits coming from the reading loop area, remain in the excited state until the five cells on their right assume the quiescent ordinary rightward transmission state, thus working as one-bit memory units. These state transitions occur when all the bits of a quintuplet have been read and the counter C_R has received five inputs. Multiple counter C_Q is used to convey each bit of a quintuplet to its corresponding memory unit.

The operations performed by the memory area on the bits coming from the reading loop area depend on the state of the constructor and in particular on the states of the gate G_0, of the double gates $G_1, \ldots, G_4$ and of the interruption I_1. All these traffic control organs are directed by the state control area.

6.1 Gate G_0

Gate G_0 controls the passage of the *default excitation* released by counter C_R when each quintuplet has been read. Such an excitation (if not stopped by interruption I_2) will reach the input point W_3 of the writing loop area and cause the writing head to retreat leftward. This is the writing arm motion associated with each cell state transition that sets the state of a cell of the assembly under construction in the blank area of the automaton.

6.2 Double Gates G_1 and G_2

Double gates G_1 and G_2, are always opened and closed in pairs. When they are open, they send a single excitation to the input point W_3 of the writing loop area for each bit received from the reading loop area. This causes the writing head to retreat leftward. In addition, if the value of the incoming bit is 1, they send the pattern 101 to the point Q of the writing loop area. This pattern corresponds to a state transition from the blank state to the quiescent ordinary downward transmission state. This kind of processing of the tape information is used in the tape copy phase of the construction.

6.3 Double Gates G_3 and G_4

Double gates G_3 and G_4 work as a pair during the tape size evaluation process. If open, they send a 11001 rightward advancing quintuplet to the writing loop control for each bit coming from the reading loop. So, at the end of the tape size evaluation process, when all the tape has been scanned, the writing head has advanced a number of steps equal to the size of the tape.

During the processes of size evaluation and copy of the tape, the bits are also grouped and interpreted in quintuplets to detect 11011 state change quintuplets. However, a set of gates prevents all the other quintuplets and the default excitations released by counter C_R from reaching the writing loop control. During these processes, in fact, the action of the writing arm is directed only according to the value of each single bit.

Interruption I_1 is used to prevent a spurious bit from reaching the memory area when the reading head movement is inverted. This area is represented in Example 11 of the interactive demo.

7 The Writing Loop Area

The writing loop control can perform five operations on the writing head H_W: the movements in the four directions right, up, left, and down and the release of a starting excitation, to activate the cell assembly after its construction. These operations are respectively activated by an excitation coming at input points W_1, W_2, W_3, W_4, and W_5. During the leftward movement, a properly timed quintuplet coming from point Q will cause the state transition of the cell at the current writing head position from the blank state to its corresponding quiescent state (*constructive process*).

All the operations performed by the writing head are obtained by exploiting the constructive and destructive capabilities of the transition rule. Each input excitation activates its corresponding set of encoders sending several properly timed patterns of excitations to the two transmission lines forming the writing arm, whose interactions result in the desired arm motion. The state transitions from the blank state to the other quiescent states that set the quiescent structure of the assembly under construction always occur during the leftward retraction of the writing arm, so as not to interfere with the fragment of the assembly already built.

7.1 Special Quintuplets

The writing loop control is able to recognize five special quintuplets and to decode them into writing arm actions or constructor state changes. Any other quintuplet is interpreted as encoding a constructive process in the blank area of the automaton that sets the state of a cell of the assembly under construction, with a consequent leftward arm movement. The five special quintuplets are the *state change quintuplet* 11011 (the bit pattern of this quintuplet is symmetric, in order to make it possible to read it scanning the tape in both directions); the *starting excitation quintuplet* 11101; the *rightward writing arm movement quintuplet* 11001; the *upward writing arm movement quintuplet* 10101; and the *downward writing arm movement quintuplet* 10011.

Because a decoder releases a single excitation if provided with a pattern of 1s and 0s such that each of the 1s of the decoder pattern match a 1 of the incoming pattern (ignoring the presence of 1s in the incoming pattern matching with 0s of the decoder), it is necessary to first check the two patterns 11011 and 11101 (this operation is performed by De_1) and, if found, to stop them before they reach the decoders of De_2. Otherwise, these patterns will be recognized also by such decoders. So one of the operations activated by the decoders of De_1 is to close the gate G_6 for just the time necessary to stop the quintuplet from being recognized. De_1 can also send a single excitation to the state control area (pattern 11011) or send a single excitation to the input point W_5 of the writing loop area through the *starting excitation transmission line p* (pattern 11101). In the same way, any other special quintuplet detected by the second set of decoders De_2 is prevented from reaching point Q. Otherwise, it would be interpreted as encoding the quiescent state of a cell of the assembly under construction. Another operation always performed when a special quintuplet is detected is the interruption of the transmission line r at point I_2. This prevents the activation of the leftward writing arm movement, which is otherwise the default operation (See Example 12 of the interactive demo).

8 The State Control Area

The main organ of this area is the state multiple counter C_S, which keeps track of the constructor state. This multiple counter exploits the constructive and destructive capabilities of the transition rule to set a rightward-oriented transmission element in the place corresponding to the current constructor state (0 to 12 in Figure 1). When the constructor is in one of the states 1, 3, 6, 9, 11, 12 (i.e., a line from one of the confluent elements of the counter to another is filled), the next state change will cause some of the cabling of the other areas of the constructor to be changed. The most important traffic control organs set by such state changes are those governing the direction of the movement of the reading head (advancing or retreating) and the interpretation of the tape bits provided in the memory area and those preventing default excitations and quintuplets to cause unwanted writing arm actions during the processes of size evaluation and copy of the tape.

The only input line of this area is line m. A single excitation arrives from this line whenever a 11011 state change quintuplet is released by the memory area and recognized by the writing loop control.

9 The Travel of a Bit of Information

The following is a description of the travel of a bit of information from the tape to the writing head, during the cell assembly construction and activation phase of the construction.

When a bit of information stored in the tape is read by the reading head H_R, either a 1 or a 10101 pattern of excitations is produced in the *return line t*, according to the value of the bit read (respectively 0 or 1). This pattern of excitations travels along line t until the discriminator D. The discriminator conveys a single excitation either to line b_0 or to line b_1 according to the pattern received (respectively 1 or 10101). Then, the excitation is duplicated. One copy is fed to the input points of the reading loop area (in order to restore the information read, move the reading head, and perform another reading); the other is sent to the memory area. During the cell assembly construction and activation phase, the bits sent to the memory area are grouped into quintuplets and sent to the writing loop control. A single excitation coming from line b_0 increments

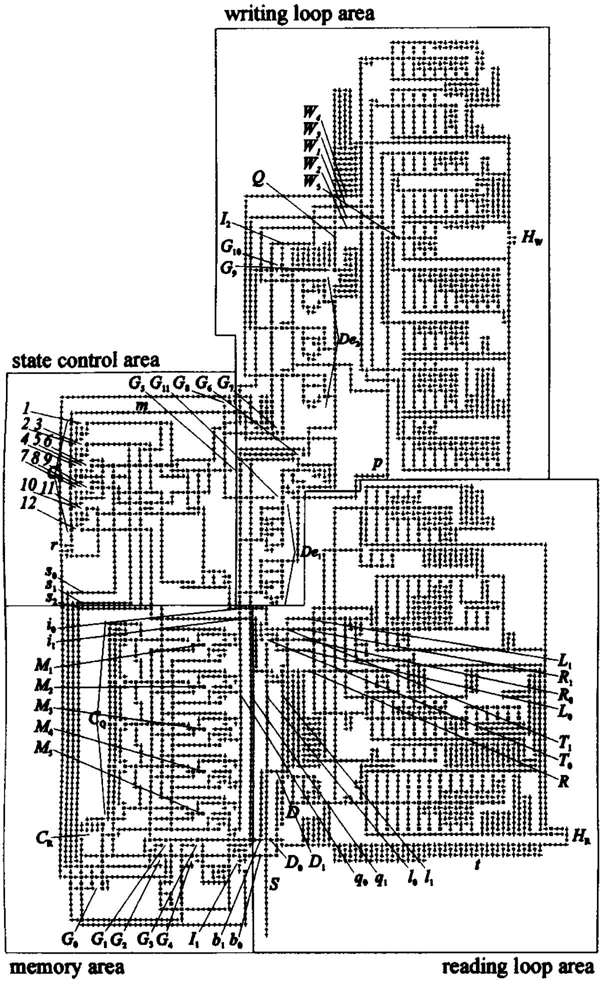

Figure 1. The universal constructor. See Appendix C.

counter C_R, without storing anything in the memory units. A single excitation coming from line b_1 increments counter C_R and stores a 1 in the memory unit indicated by multiple counter C_Q. When counter C_R reaches its threshold, all the bits stored in the memory area are released together as a quintuplet of excitations. In addition, a default excitation is sent to line r. From now on, we will follow the travel of a whole quintuplet of tape bits. This quintuplet, traveling along line q_0, enters the control of the writing loop area and is conveyed to all the decoders of sets De_1 and De_2. One of these decoders may recognize the quintuplet. If not, it will reach point Q and will be interpreted as encoding a constructive process that sets the state of a cell of the assembly under construction. This constructive process is associated with a leftward retraction of the writing head. This is the operation that is activated by the arrival of the default excitation released by counter C_R at the input point W_3 of the writing loop area. On the contrary, if one of the decoders recognizes the quintuplet, it stops the quintuplet along with the default excitation of line r and activates one of the special operations of the constructor. The two decoders of De_1 feed the state control area and the starting excitation transmission line p going to input point W_5; the decoders of De_2

feed the input points W_1, W_2, and W_4 of the writing loop area, corresponding to the rightward, upward, and downward motions of the writing head.

10 The Construction

The construction can be divided in several phases, each of which is characterized by one of the following groups of constructor states: [0,1], [2,3], [4,5,6], [7,8,9], [10,11], [12]. Note that I consider as phase transitions only those state transitions that entail a cabling change in an area of the constructor different from the state control area. In the constructor described in the file `tape.evn` on Alife Online, the six phases of the construction indicated here start respectively at Steps 0, 14033, 28270, 168036, 308223, and 322390. The construction ends at Step 449127.

10.1 The Tape Positioning Phase [0,1]

The construction begins when the starting excitation coming from point S starts the reading loop action. The first 11011 quintuplet encountered by the reading head at the beginning of the tape makes the constructor enter state 1. In this state, the bits coming from the reading loop are interpreted as quintuplets and decoded into writing arm movements. In this phase, the *tapeposition area* of the tape is read and the writing head is positioned at the starting point of the tape copy.

10.2 The First Reading Head Repositioning Phase [2,3]

At the end of the *tape position area* the reading head finds a 11011 quintuplet and the state multiple counter C_S is incremented to 2. This phase transition entails that the movement direction of the reading head is inverted. At this point a number of gates are closed to prevent the writing head from being affected by any of the bits read by the reading head. Then, the reading head will again detect the same 11011 quintuplet (so that the constructor state is set to 3) and will continue to retreat until it finds the 11011 quintuplet at the beginning of the tape, at which point the multiple counter C_S is incremented to 4.

10.3 The Tape Size Evaluation Phase [4,5,6]

When the constructor passes from State 3 to State 4, the direction of the reading head movement is inverted again and the states of the traffic control organs are changed to make the writing head advance for each bit read. This process will continue until the detection of the 11011 quintuplet at the end of the tape. At that point, the writing arm will be advanced from the starting point of the tape copy a number of steps equal to the tape length. During this phase, the reading head will scan all the tape, encountering again the 11011 quintuplet at the beginning of the tape (Constructor State 5), the 11011 quintuplet that divides the two areas of the tape (Constructor State 6), and that at the end of the tape.

10.4 The Tape Copy Phase [7,8,9]

When the reading head encounters the 11011 quintuplet at the end of the tape (Constructor State 7), it will begin to scan the tape retreating and the cabling of the constructor will be changed to allow a different interpretation of each single bit. A 0 bit will cause the writing head simply to retreat leftward and a 1 bit will cause it to retreat in the same direction and inject into the cell at its right the pattern of excitations corresponding to the state transition to the quiescent ordinary downward transmission state. In this way, the whole tape is copied at the end of the phase. That is, when the reading head has encountered again the 11011 quintuplet at the end of the tape (Constructor

State 8) and those situated at the middle (Constructor State 9) and at the beginning of the tape. When the reading head reads the 11011 quintuplet at the beginning of the tape, the constructor enters State 10 and the tape copy phase is over.

10.5 The Second Reading Head Repositioning Phase [10,11]

In this phase, without any actions of the writing arm, the tape position area of the tape is scanned until the *cell assembly position and structure area*, reading again the 11011 quintuplets at the beginning (Constructor State 11) and at the middle of the tape.

10.6 The Cell Assembly Construction and Activation Phase [12]

When the reading head enters the *cell assembly position and structure area* (Constructor State 12), the constructor begins again to interpret the tape information in quintuplets. The information read in this phase directs the writing arm movements in the blank area of the automaton toward the cell assembly position, the construction of the cell assembly to be built, the release of one or more starting excitations, and the movements of the writing arm to its original position. After this phase, the constructor ceases its activity and goes back to its quiescent state.

11 The Transition Rule

Below is an intuitive presentation of the transition rule governing the cellular automaton in which our constructor is embedded; see Appendix A for a more formal description.

The domain of the transition rule is the set of all the possible ordered quintuplets of cell states (those of the cell and of its four neighbors). Its range is the set of the possible cell states.

11.1 The Cell States

In this transition rule each cell can assume one of 32 possible states. The state set is divided in *quiescent states* and *excited states* and the transition rule ensures that any assembly made of cells in quiescent states (i.e., in its quiescent state) remains unaltered until acted upon by a cell in an excited state.

This transition rule possesses 10 quiescent states: the fundamental or blank state, four oriented quiescent ordinary transmission states (rightward, upward, downward, and leftward), four oriented quiescent special transmission states, and the quiescent confluent state. An *element* is defined as a cell in a state belonging to a certain class of states. The classes of states considered are composed by a quiescent state and all the states that can be obtained from it without a previous transition to the blank state. So all the quiescent states with the exception of the blank state characterize an element.

The transmission elements can assume one excited state. The confluent element can be excited, next-excited, excited-next-excited, in a horizontal conduction state, in a vertical conduction state, or in a double conduction state. Note that von Neumann's confluent element had only the first three of these excited states. The first three excited states are used to implement delay, logical AND, signal duplication, and memory operations; the last three are crossing states. Logical OR is achieved when two transmission elements of the same type join directly together. The remaining eight states are called *sensitized states* and are used to direct the constructive process, that is, the transition from the blank state to a quiescent state.

11.2 Transmission of Data

The basic data used in the constructor are patterns of excitations, which travel in lines of oriented transmission elements called *transmission lines* and are processed by two basic organs: the *encoder*, which transforms a single excitation into a given pattern of excitations, and the *decoder*, which outputs a single excitation if provided with a pattern that matches a target pattern. Moreover, using a confluent element connected with one input and two output transmission lines it is possible to duplicate patterns of excitations (See Example 1 of the interactive demo).

11.3 Constructive and Destructive Processes

To implement constructive and destructive properties, the state set possesses two types of transmission states, the *ordinary transmission states* and the *special transmission states*. A pattern of five bits (excitations) injected into a cell in the blank state results in a transition to one of the other nine quiescent states. On the other hand, an excitation injected by a special transmission element into an ordinary transmission element or into a confluent element, or an excitation injected by an ordinary transmission element into a special transmission element results in a transition to the blank state (See Example 2 of the interactive demo). In all the other cases, an excitation is propagated along the orientation of the transmission elements and does not permanently alter the quiescent structure of the automaton. By contrast, constructive and destructive processes result in a permanent alteration of the quiescent structure of the automaton.

In the universal constructor presented here, the alteration of quiescent structures caused by constructive and destructive processes are mainly used to deviate the traffic of information (patterns of excitations) by changing the orientation of particular transmission elements that work as switches, and to keep track of the constructor state. Thus, the constructor is made of transmission lines, encoders, decoders, and some other special organs that provide internal memory, keep track of the constructor state, and control the information traffic.

11.4 The Confluent Element

As mentioned above, the only difference between this transition rule and that described in [6] is the behavior of the confluent element, which now possesses three more excited states. In this rule the confluent element acts (a) as a memory unit (when none of its four neighbors is an outgoing transmission element); (b) as a one-step delay (when one of its neighbors is an incoming transmission element and another is an outgoing transmission element); (c) as a logical AND (when among its neighbors there are one outgoing and two or three incoming transmission elements); (d) as a signal duplication unit (when among its neighbors there are one incoming and two or three outgoing transmission elements); (e) as an ordinary-to-special signal converter (when its input transmission elements are of the ordinary type and its output transmission elements are of the special type); and (f) as a crossing unit (when its neighborhood consists of two pairs of transmission elements, with each pair containing an incoming and an outgoing transmission element).

Note that with a slightly more elaborate change in von Neumann's transition rule, it is possible to achieve direct signal crossing without increasing the number of cell states. Consider Figure 2, which represents a crossing configuration, that is, the intersection of two transmission lines. If each cell of the configuration could, by analyzing the state of its neighbors, recognize the crossing configuration, then the three additional excited states of the confluent element of our extended transition rule could be replaced by the three excited states of the confluent element already present in von Neumann's original transition rule. The central confluent element can recognize the crossing configuration just as the extended transition rule of this article does (Function f of the preceding paragraph). The transmission elements adjacent to the central confluent element can recognize the crossing configuration because there are three confluent elements among their neighbors. Because none of the cell assemblies required to implement the other

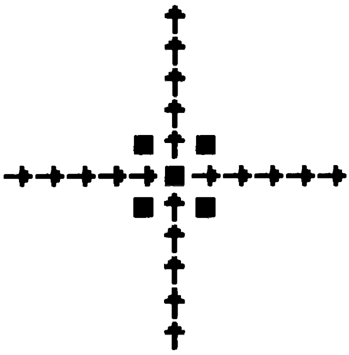

Figure 2. An alternative implementation of signal crossing.

operations of the universal constructor needs a configuration in which a transmission element has three confluent elements among its neighbors, this configuration could be used to identify direct signal crossing.

12 Organs and Transmission Lines

Patterns of excitations traveling in cell assemblies are the information processed by the universal constructor. *Organs* are cell assemblies capable of deterministically producing certain outputs, when presented with properly timed inputs. Each organ has a simple functional interpretation. Organs are connected together by *transmission lines*, sequences of oriented transmission elements of the same type, and work as parallel processors on the information flowing in the constructor. Thus, we may think of the tape as the *program* to be executed and the cell assembly being built as the output of the *computation*.

This interpretation ceases to be correct if we consider that the output of our computation (that, indeed, is a *construction*) is an object of the same kind as the machine that produced it. The constructed object can feedback on the constructor. For example, the cell assembly to be built can be a Turing machine that is able to work successively in connection with the constructor itself to produce any recursive structure. We will not go further into this matter here; we will consider the cell assembly produced only as the output of a construction that, once activated by the starting excitation, begins to behave independently. However, the intrinsic homogeneity of cellular automata plays a fundamental role in the dynamics of self-reproduction. Note that in this model, (a) the static data describing the self-reproducing machine (the *tape*), (b) the dynamic data flowing in the machine itself (the *excitations*), (c) the *organs* of the machine, (d) their cabling (the *transmission lines*), and (e) the surrounding space in which self-reproduction takes place (the field of cells in the blank state to be excited) are made of objects of the same kind. Each cell of the cellular automaton can be at the same time an operand and an operator.

Organs are cell assemblies that retain a constant functional interpretation. This does not necessarily mean that they maintain a stable quiescent state at the end of each computation. Some organs, such as the *signal processing organs*, do maintain a constant state. The signal processing organs deterministically output certain patterns of excitations upon being fed suitable inputs. Another class of organs, the *traffic control organs*, are dynamic. Such organs may permanently alter their quiescent structure in order to deviate signal traffic. These alterations may produce stable changes in the cabling of the constructor. Because we define the *constructor state* as the state of the cabling of the constructor, only such stable changes will be considered as *constructor state changes.*

12.1 Traffic Control Organs

The traffic control organs used in this design are *gates, double gates, switches, interruptions, counters,* and *multiple counters.*

A *gate* is a simple cell assembly made of a confluent element in a transmission line and of an apparatus that creates and destroys a transmission element pointing toward it. If such a transmission element is present, the transition rule ensures that no pattern of excitations will pass through the confluent element; if absent, patterns will pass unaltered (See Example 4 of the interactive demo).

A *double gate* consists of a confluent element with one input transmission line and two output transmission lines; one of the output lines is always open while the other can be switched on or off (See Example 5 of the interactive demo).

A *switch* is an organ placed at the intersection of two transmission lines. The switch orients the transmission element at the intersection point so as to direct traffic onto one or the other of the lines. (See Example 7 of the interactive demo.)

An *interruption* is just a rightward transmission element of a transmission line together with the apparatus to destroy such an element, so that the next excitation traveling along the transmission line will be lost to restore it (See Example 6 of the interactive demo). The interruption can be considered a particular kind of traffic control organ because the cabling change it produces is not stable.

A *counter* (for example, the counter C_R in the memory area) performs multiple interruptions. The number of interruptions performed by the counter represents its threshold. Each excitation entering the counter fills one of its interruptions until it is completely filled. A subsequent excitation resets the counter, again performing the interruptions, and produces an excitation at the counter output.

Linking several counters, so that the output excitation of each counter (among the other operations) activates the following one, we can obtain *multiple counters.* (See Example 9 of the interactive demo). Multiple counters C_Q and C_S are used, respectively, in the memory area and in the state control area.

12.2 Signal Processing Organs

The two main *signalprocessing organs* are the *encoder* and the *decoder.* Both of these organs were used in von Neumann's original design. Basically, an encoder is an organ that receives a single excitation as input and produces a certain pattern of excitations as output. The decoder has the opposite behavior: upon receiving a pattern of excitations that matches a particular target pattern, it outputs a single excitation.

13 Conclusions

We have shown how to implement a machine performing a complex algorithm in a cellular automaton. This machine is the first example of a working universal constructor. For historical reasons this implementation was carried out using a transition rule very similar to that proposed by von Neumann. However, to use cellular automata as models for parallel processing, more complex transition rules should be considered in the future. In fact, von Neumann's rule was conceived to achieve constructive processes and its computational capabilities were introduced only to direct them. Moreover, the proof of the universal computational capabilities of such a transition rule was carried out by demonstrating the implementability of a Turing machine. A cellular automaton is an intrinsically parallel computer; to use it to simulate a strictly sequential Turing machine is not efficient, thus other strategies should be examined. One approach, which will be the subject of a subsequent article, is to introduce more algorithmically expressive transition rules. One of the main shortcomings of von Neumann's cellular automata, with respect to parallel processing, is the lack of signal crossing. This shortcoming makes the structures implemented in such automata cumbersome and very weakly parallel. In this article we have seen how to solve this problem, even without increasing the cell state number. With more complex transition rules, one can design more efficient signal-processing organs, capable of performing all the operations of logic and arithmetic calculus. In cellular automata used as models for parallel processing, universal constructive capabilities can be used to set up the quiescent structures underlying each particular computation.

Acknowledgments

I would like to thank Howard Gutowitz of ESPCI for the encouragements and the suggestions he gave me in writing this paper. I am also grateful to Renato Nobili of the University of Padova for many stimulating discussions about cellular automata.

References

1. Codd, E. (1968). *Cellular automata.* New York: Academic Press.
2. Langton, C. G. (1984). *Self-reproduction in cellular automata. Physica 10D,* 135–144.
3. Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica 22D,* 120–149.
4. Thatcher, J. (1970). Universality in the von Neumann cellular model. In A. Burks (Ed.), *Essays in cellular automata* (pp. 132–186). Urbana: University of Illinois Press.
5. Turing, A. (1936). On computable numbers, with an application to the Entscheidungproblem. *Proceedings of the London Mathematical Society,* Series 2, 42, 230–265.
6. von Neumann, J. (1966). In A. Burks (Ed.), *The theory of self-reproducing automata.* Urbana: University of Illinois Press.

Appendix A: A Formal Description of the Transition Rule

In the following, the blank state is indicated by V. The transmission states are indicated by $T^{c}_{\tau,\mathbf{u}}$. Variable τ takes value 1 or 0 according to whether the transmission state type is special or ordinary; variable ϵ takes the value 1 or 0 according to whether the state is excited or not; unit vector $\mathbf{u}$ points at one of the following directions: rightward: $= \mathbf{i}$; upward: $= \mathbf{j}$; leftward: $= -\mathbf{i}$; downward: $= -\mathbf{j}$ (unit vectors $\mathbf{i}$, $\mathbf{j}$ are, as usual, the positive directions of the cell-lattice Cartesian reference frame).

The confluent elements bear two excitation modes and three additional excited states for signal crossing: *excitation* and *next-excitation*, that are indicated by $C^{\epsilon,\epsilon'}$ and C_i (the *left-right conduction* state), C_j (*up-down conduction* state), C_k (*double conduction* state). Variable ϵ takes the value 1 or 0 according to whether the confluent element is *excited* or *not excited*, whereas ϵ' takes the value 1 or 0 to indicate that the confluent element is respectively *next excited* or *not next excited.*

The transition from the blank state to another quiescent state occurs through a short chain of intermediate states called *sensitized states.* The first sensitized state is the state produced by a single excitation of the blank state. A sensitized state always evolves toward a quiescent state either spontaneously or under the action of a pattern of excitations. The resulting quiescent state depends upon the pattern of excitations injected. The sensitized states can be thought of as the nodes of a bifurcation tree taking root from the first sensitized state and leading to the transmission and confluent quiescent states. Further excitation of a quiescent state results in an excited state. This excited state often decays to a quiescent state at the next step.

The sensitized states are symbolically represented by $\Sigma(n, \beta)$, where n ($n = 1, 2, 3, 4, 5$) represents the *depth* of the bifurcation tree and β the pattern of excitations, which generates the node, in binary code. In other terms, $\Sigma(1, 1)$ represents the root; $\Sigma(2, 10)$, $\Sigma(2, 11)$ the first-order nodes; $\Sigma(3, 100)$, $\Sigma(3, 101)$, $\Sigma(3, 110)$, $\Sigma(3, 111)$ the second-order nodes; and so on. The terminal nodes are

$$\Sigma(5, 10000) = T^0_{0,\mathbf{i}}, \quad \Sigma(5, 10001) = T^0_{0,\mathbf{j}}, \quad \Sigma(4, 1001) = T^0_{0,-\mathbf{i}},$$

$$\Sigma(4, 1010) = T^0_{0,-\mathbf{j}}, \quad \Sigma(4, 1011) = T^0_{1,\mathbf{i}}, \quad \Sigma(4, 1100) = T^0_{1,\mathbf{j}},$$

$$\Sigma(4, 1101) = T^0_{1,-\mathbf{i}}, \quad \Sigma(4, 1110) = T^0_{1,-\mathbf{j}}, \quad \Sigma(4, 1111) = C^{0,0}.$$

Let $\mathbf{v}$ indicate cell positions within the lattice and t be a discrete time variable, and let $S(\mathbf{v}, t)$ represent the state of the cell at position $\mathbf{v}$ at time t.

The transition rule is expressed by the following logical conditions:

$\mathbf{L_1}$ Let $S(\mathbf{v}, t-1) = V$ then

(*a*) $S(\mathbf{v}, t) = \Sigma(1, 1)$ iff $S(\mathbf{v}+\mathbf{u}, t-1) = T^1_{\tau,-\mathbf{u}}$ for some $\mathbf{u}$;

(*b*) $S(\mathbf{v}, t) = V$ iff (*a*) does not hold.

Meaning: any excitation carried by a transmission element, whether special or ordinary, changes the state of a cell in the blank state to the first sensitized state.

$\mathbf{L_2}$ Let $S(\mathbf{v}, t-1) = \Sigma(n, \beta)$ with either $n < 4$ or $n = 4$ and $\beta = 1000$ then

(*a*) $S(\mathbf{v}, t) = \Sigma(n+1, \beta+\beta+1)$ iff $S(\mathbf{v}+\mathbf{u}, t-1) = T^1_{\tau,-\mathbf{u}}$ for some $\mathbf{u}$;

(*b*) $S(\mathbf{v}, t) = \Sigma(n+1, \beta+\beta)$ iff (*a*) does not hold.

Meaning: governing sensitized state dynamics.

$\mathbf{L_3}$ Let $S(\mathbf{v}, t-1) = T^{\epsilon}_{\tau,\mathbf{u}}$ then

(*a*) $S(\mathbf{v}, t) = V$ iff $S(\mathbf{v}+\mathbf{u}', t-1) = T^1_{\tau',-\mathbf{u}'}$ for some $\mathbf{u}'$ and also $\tau \neq \tau'$;

(*b*) $S(\mathbf{v}, t) = T^1_{\tau,\mathbf{u}}$ iff (*a*) does not hold and (b_1), (b_2), (b_3), (b_4), or (b_5) holds:

(b_1)	$S(\mathbf{v}+\mathbf{u}', t-1) = T^1_{\tau,-\mathbf{u}'}$	for some	$\mathbf{u}' \neq \mathbf{u}$;
(b_2)	$S(\mathbf{v}+\mathbf{u}', t-1) = C^{1,\epsilon'}$	for some	$\mathbf{u}' = \pm\mathbf{i}, \pm\mathbf{j} \neq \mathbf{u}$;
(b_3)	$S(\mathbf{v}+\mathbf{u}', t-1) = C_k$	for some	$\mathbf{u}' = \pm\mathbf{i}, \pm\mathbf{j} \neq \mathbf{u}$;
(b_4)	$S(\mathbf{v}+\mathbf{u}', t-1) = C_i$	for some	$\mathbf{u}' = \pm\mathbf{i} \neq \mathbf{u}$;
(b_5)	$S(\mathbf{v}+\mathbf{u}', t-1) = C_j$	for some	$\mathbf{u}' = \pm\mathbf{j} \neq \mathbf{u}$.

(*c*) $S(\mathbf{v}, t) = T^0_{\tau,\mathbf{u}}$ iff neither (*a*) nor (*b*) holds.

Meaning: (*a*) guarantees mutual destruction capability between special and ordinary transmission elements while (*b*) and (*c*) govern excitation propagation.

$\mathbf{L_4}$ Let $S(\mathbf{v}, t-1) = C^{\epsilon,\epsilon'}$ or $S(\mathbf{v}, t-1) = C_z$ ($z = i, j, k$) then

(*a*) $S(\mathbf{v}, t) = V$ iff $S(\mathbf{v}+\mathbf{x}, t-1) = T^1_{1,-\mathbf{x}}$ for some $\mathbf{x}$;

(b_1) $S(\mathbf{v}, t) = C_i$ iff $S(\mathbf{v}-\mathbf{x}, t-1) = T^1_{0,\mathbf{x}}$, $S(\mathbf{v}+\mathbf{x}, t-1) = T^{\epsilon''}_{\tau,\mathbf{x}'}$ with $\mathbf{x}' \neq -\mathbf{x}$, $S(\mathbf{v}-\mathbf{y}, t-1) = T^0_{0,\mathbf{y}}$ and $S(\mathbf{v}+\mathbf{y}, t-1) = T^{\epsilon'''}_{\tau',\mathbf{y}'}$ with $\mathbf{y}' \neq -\mathbf{y}$, for some $\mathbf{x} = \pm\mathbf{i}$ and $\mathbf{y} = \pm\mathbf{j}$;

(b_2) $S(\mathbf{v}, t) = C_j$ iff $S(\mathbf{v} - \mathbf{x}, t - 1) = T^1_{0,\mathbf{x}}$, $S(\mathbf{v} + \mathbf{x}, t - 1) = T^{\epsilon''}_{\tau,\mathbf{x}'}$ with $\mathbf{x}' \neq -\mathbf{x}$, $S(\mathbf{v} - \mathbf{y}, t - 1) = T^0_{0,\mathbf{y}}$ and $S(\mathbf{v} + \mathbf{y}, t - 1) = T^{\epsilon'''}_{\tau',\mathbf{y}'}$ with $\mathbf{y}' \neq -\mathbf{y}$, for some $\mathbf{x} = \pm\mathbf{j}$ and $\mathbf{y} = \pm\mathbf{i}$;

(b_3) $S(\mathbf{v}, t) = C_k$ iff $S(\mathbf{v} - \mathbf{x}, t - 1) = T^1_{0,\mathbf{x}}$, $S(\mathbf{v} + \mathbf{x}, t - 1) = T^{\epsilon''}_{\tau,\mathbf{x}'}$ with $\mathbf{x}' \neq -\mathbf{x}$, $S(\mathbf{v} - \mathbf{y}, t - 1) = T^1_{0,\mathbf{y}}$ and $S(\mathbf{v} + \mathbf{y}, t - 1) = T^{\epsilon'''}_{\tau',-\mathbf{y}'}$ with $\mathbf{y}' \neq -\mathbf{y}$, for some $\mathbf{x} = \pm\mathbf{i}$ and $\mathbf{y} = \pm\mathbf{j}$.

Meaning: neighborhood conditions that enable a confluent element to work as a crossing unit (b_1) in the horizontal direction, (b_2) in the vertical direction, and (b_3) in the horizontal and vertical directions at the same time.

(c) $S(\mathbf{v}, t) = C^{\epsilon',1}$ (with $\epsilon' = 0$ if $S(\mathbf{v}, t - 1) = C_z$, $z = i, j, k$) iff (a), (b_1), (b_2), and (b_3) do not hold and both (c_1) and (c_2) hold:

(c_1) $S(\mathbf{v} + \mathbf{x}, t - 1) = T^1_{0,-\mathbf{x}}$ for some $\mathbf{x}$;
(c_2) never $S(\mathbf{v} + \mathbf{x}', t - 1) = T^0_{0,-\mathbf{x}'}$ for any $\mathbf{x}'$.

Meaning: governing the confluent element dynamics when it is not working as a crossing unit. These are the same as (b_1) and (b_2) of the original rule (see the description of von Neumann's transition rule in [6]).

(d) $S(\mathbf{v}, t) = C^{\epsilon',0}$ (with $c' = 0$ if $S(\mathbf{v}, t - 1) = C_z$, $z = i, j, k$) iff (a), (b_1), (b_2), (b_3), and (c) do not hold and $S(\mathbf{v} + \mathbf{x}, t - 1) = T^{\epsilon''}_{\tau,\mathbf{x}'}$ for some $\mathbf{x} \neq -\mathbf{x}'$;

(e) $S(\mathbf{v}, t) = C^{\epsilon,\epsilon'}$ (with $\epsilon = \epsilon' = 0$ if $S(\mathbf{v}, t - 1) = C_z$, $z = i, j, k$) iff (a), (b_1), (b_2), (b_3), (c), and (d) do not hold.

Meaning: the confluent element passes from an excited state to the quiescent state only if one of its four neighbors is an outgoing transmission element.

Appendix B: The web demonstrations

The interactive demonstrations referred to in the text, as well as several others, are found at the Alife Online web site, URL:`http://alife.santafe.edu/refereed/96/jvn`. The constructor described in the body of this article is represented in the file `tape.evn` of the interactive demo. Note, however, that the universal constructor of `tape.evn` is not provided with a tape containing its own description. In fact, the tape required to describe a cell assembly is five times bigger than the cell assembly described, and the whole process of self-reproduction would be too long to be actually simulated in this demo, as the time to read a bit increases with its distance from the origin of the tape. The tape of `tape.evn` contains the description of a loop made of transmission elements, to be activated with a single starting excitation.

Alife Online also contains two more examples of universal constructors: `constructor.jvn`, implemented in von Neumann's original automaton; and `constructor.evn`, implemented in our extended automaton. Neither of these constructors would be able to achieve complete self-reproduction as they are not able to produce a copy of the tape.

The main differences between `constructor.jvn` and `tape.evn` concern the memory area and the state control area. The memory area of `constructor.jvn` is implemented using von Neumann's *static-dynamic converter*; the state control area is absent. Recall that the state control area is needed to implement the two different interpretations of the tape used to achieve cell assembly production and tape copy. Constructor `constructor.jvn` shows how to substitute or reimplement more efficiently most of the apparatus described in [6], using structures that can be embedded in von Neumann's original cellular automaton. As an example, the problems of crossing transmission lines and signal interference are solved more elegantly than by using von Neumann's *coded-channel*, by the use of local traffic control organs. This constructor, furthermore, is much smaller than von Neumann's original universal constructor.

Constructor `constructor.evn` exploits the new crossing states of the extended transition rule to read nine tapes at the same time. It possesses only the reading loop area and the writing loop area. Because it simultaneously reads all the bits of information that are necessary to direct one step of the writing arm action, it does not need to store temporarily the single bits collected from the tape. The input-output operations of this constructor are organized so as to keep all its areas permanently active during the construction. Thus, this constructor is an example of a parallel algorithm achieved by the use of the new crossing states. This constructor is also the most structurally simple of the universal constructors in the Alife Online interactive demo.

The other demonstrations represent in detail the most important apparatus of the constructors: the dynamics of an encoder feeding two different decoders (Example 1); some cases of constructive and destructive processes (Example 2); all the possible functions of the confluent element (Example 3); the traffic control organs used in constructor `tape.evn`, the gate (Example 4), the double gate (Example 5), the interruption (Example 6), the switch (Example 7), the counter (Example 8), and the multiple counter (Example 9); a bit reading by the reading loop of `tape.evn` (Example 10); the arrival of the fifth bit of a quintuplet at the memory area of `tape.evn`, with the consequent release of the whole quintuplet (Example 11); and the writing arm of `tape.evn` setting the state of a cell in the blank area of the automaton (Example 12).

Appendix C: Notes to Figures

The reading loop area. Cells in quiescent ordinary transmission states are represented with blue arrows, cells in quiescent special transmission states with red arrows, and cells in the quiescent confluent state with violet squares. All the structures represented in the figures are in their quiescent state. The reading loop area receives inputs from the state control area through input lines i_0 and i_1 and sends its outputs to the memory area through output lines b_0 and b_1. At the beginning of the construction, the starting excitation of the constructor, conveyed from point S, reaches point R and generates the first reading pattern 10101. Any reading pattern passes through the reading head H_R and is changed to a single excitation or remains unaltered according to whether it finds a 0 or a 1 bit at its current position on the tape. Bit reading is always a destructive operation as it always replaces the value of the bit read with a 1. The pattern generated by the reading head (1 or 10101) is subsequently recognized by the discriminator D, which sends a single excitation respectively either in line b_0 or line b_1 according to the pattern received. Such a single excitation is then duplicated and sent to the memory area and, through line l_0 or l_1, to the input points of the reading loop area, to restore tape information, move the reading head, and release another 10101 reading quintuplet. Each single excitation arriving at one of the input points R, R_0, R_1, L_0 and L_1 activates in turn all the encoders connected to it, generating all the properly timed patterns of excitations required to perform the desired operation on head H_R (respectively releasing a reading quintuplet, moving right and restoring a 0 at the current position, moving right and restoring a 1, moving left and restoring a 0, moving left and restoring a 1). Input excitations coming from the state control area through line i_0 invert the direction of the tape scanning, changing the orientation of transmission elements T_0 and T_1; line i_1 carries the excitation that stops the reading loop action at the end of the construction.

The memory area. This area receives inputs from the reading loop area through lines b_0 and b_1 and from the state control area through lines s_0, s_1, and s_2. All the outputs of this area are sent to the control of the writing loop area through lines r, q_0, and q_1. The bits coming from lines b_0 and b_1 are temporarily stored in the memory units $M_1, \ldots, M_5$ and released in quintuplets when counter C_R reaches its threshold. Multiple counter C_Q is used to convey each bit of a quintuplet to its corresponding memory unit. If one of the gates pairs G_1-G_2 and G_3-G_4 is open, additional preprocessing of each single bit will be activated. The first pair is open during the tape copy phase of the construction, the second during the tape-size-evaluation phase. The quintuplets stored and released by the memory units are always sent to the writing loop area through line q_0, while those created after the analysis of single bits are conveyed through line q_1. The default excitation released by counter C_R is conveyed to line r.

The writing loop area. The control apparatus of this area consists of two sets of decoders: the first (De_1) to recognize the state change and the starting excitation quintuplets, the second (De_2) to recognize writing arm movement quintuplets. If all the gates of this area are open, the quintuplets conveyed by the line q_0 will pass through both sets, while those carried by q_1 will pass only through the second set. However, during the reading head repositioning, the tape size evaluation, and the tape copy phases of the construction the gates of this area will be closed to prevent unwanted writing arm actions. The first decoder of this area is used to catch the 11011 state change quintuplets and to send its output to the state control area. The second decoder and all those of the second set recognize the writing arm action special quintuplets and feed one of the input points of the writing loop. The leftward movement input point is fed by the default excitation line r. The arrival of such a single excitation is often associated with that of a properly timed quintuplet at point Q, resulting in a constructive process in the cell at the current writing head position.

The state control area. This area consists of multiple counter C_S, fed by line m, and of a number of transmission lines departing from it and leading to the other areas of the constructor. Such transmission lines feed all the traffic control organs of the constructor, determining its state and dynamics. All the positions of multiple counter C_S correspond to a constructor state and are indicated by a progressive number from 1 to 12. Any state change that determines the current counter position to be moved to a new line (i.e., those 1-2, 3-4, 6-7, 9-10, 11-12, and 12-end states) causes a stable change in the cabling of one or more of the three other areas of the constructor, a change we call a phase transition. The different states of the constructor determine the different interpretations of the tape information that are necessary to achieve complete self-reproduction.

This article has been cited by:

1. Edward B. Clark, Simon J. Hickinbotham, Susan Stepney. 2017. Semantic closure demonstrated by the evolution of a universal constructor architecture in an artificial chemistry. *Journal of The Royal Society Interface* **14**:130, 20161033. [Crossref]
2. Alex Ellery. John von Neumann's self-replicating machine — Critical components required 000314-000319. [Crossref]
3. Matthew S. Moses, Hans Ma, Kevin C. Wolfe, Gregory S. Chirikjian. 2014. An architecture for universal construction via modular robotic components. *Robotics and Autonomous Systems* **62**:7, 945-965. [Crossref]
4. . References 271-298. [Crossref]
5. Gianluca Tempesti. 2012. Self-replicating loops: a survey. *International Journal of General Systems* **41**:6, 633-643. [Crossref]
6. Ayoola R. Yinusa, Chrystopher L. Nehaniv. Study of inheritable mutations in von Neumann self-reproducing automata using the GOLLY simulator 211-217. [Crossref]
7. Pierre T. Kabamba, Patrick D. Owens, A. Galip Ulsoy. 2011. The von Neumann threshold of self-reproducing systems: theory and application. *Robotica* **29**:01, 123-135. [Crossref]
8. William M Stevens. 2011. A self-replicating programmable constructor in a kinematic simulation environment. *Robotica* **29**:01, 153-176. [Crossref]
9. Manuel Rivas-Perez, A. Linares-Barranco, Francisco Gomez-Rodriguez, A. Morgado, A. Civit, G. Jimenez. An AER Spike-Processing Filter Simulator and Automatic VHDL Generator Based on Cellular Automata 157-165. [Crossref]
10. M. Rivas-Perez, A. Linares-Barranco, J. Cerda, N. Ferrando, G. Jimenez, A. Civit. Visual spike-based convolution processing with a Cellular Automata architecture 1-7. [Crossref]
11. Tim J. Hutton. 2010. Codd's Self-Replicating Computer. *Artificial Life* **16**:2, 99-117. [Abstract] [PDF] [PDF Plus]
12. Zhijian Pan, James A. Reggia. 2010. Computational Discovery of Instructionless Self-Replicating Structures in Cellular Automata. *Artificial Life* **16**:1, 39-63. [Abstract] [PDF] [PDF Plus]
13. A. Linares-Barranco, J.L. Sevillano, M. S. Obaidat, N. Ferrando, J. Cerda, D. Cascado, G. Jimenez, A. Civit. AER filtering using GLIDER: VHDL cellular automata description 614-617. [Crossref]
14. Yousuke Takada, Teijiro Isokawa, Ferdinand Peper, Nobuyuki Matsui. 2007. Asynchronous self-reproducing loops with arbitration capability. *Physica D: Nonlinear Phenomena* **227**:1, 26-35. [Crossref]
15. Tim J. Hutton. 2007. Evolvable Self-Reproducing Cells in a Two-Dimensional Artificial Chemistry. *Artificial Life* **13**:1, 11-30. [Abstract] [PDF] [PDF Plus]
16. Yousuke Takada, Teijiro Isokawa, Ferdinand Peper, Nobuyuki Matsui. 2006. Construction universality in purely asynchronous cellular automata. *Journal of Computer and System Sciences* **72**:8, 1368-1385. [Crossref]
17. Stephen Kercel. The von Neumann's Self-Replicator and a Critique of Its Misconceptions 179-206. [Crossref]
18. YOUSUKE TAKADA, TEIJIRO ISOKAWA, FERDINAND PEPER, NOBUYUKI MATSUI. 2006. UNIVERSAL CONSTRUCTION AND SELF-REPRODUCTION ON

SELF-TIMED CELLULAR AUTOMATA. *International Journal of Modern Physics C* **17**:07, 985-1007. [Crossref]

19. Adzni Abdul Rahim, Jason Teo, Azali Saudi. An Empirical Comparison of Code Size Limit in Auto-Constructive Artificial Life 1-6. [Crossref]
20. D. Mange, A. Stauffer, L. Peparaolo, G. Tempesti. 2004. A macroscopic view of self-replication. *Proceedings of the IEEE* **92**:12, 1929-1945. [Crossref]
21. Chris Salzberg, Antony Antony, Hiroki Sayama. 2004. Evolutionary dynamics of cellular automata-based self-replicators in hostile environments. *Biosystems* **78**:1-3, 119-134. [Crossref]
22. Daniel Mange, André Stauffer, Enrico Petraglio, Gianluca Tempesti. 2004. Artificial cell division. *Biosystems* **76**:1-3, 157-167. [Crossref]
23. Arnold Smith, Peter Turney, Robert Ewaschuk. 2003. Self-Replicating Machines in Continuous Space with Virtual Physics. *Artificial Life* **9**:1, 21-40. [Abstract] [PDF] [PDF Plus]
24. Kohji Tomita, Haruhisa Kurokawa, Satoshi Murata. 2002. Graph automata: natural expression of self-reproduction. *Physica D: Nonlinear Phenomena* **171**:4, 197-210. [Crossref]
25. Tim J. Hutton. 2002. Evolvable Self-Replicating Molecules in an Artificial Chemistry. *Artificial Life* **8**:4, 341-356. [Abstract] [PDF] [PDF Plus]
26. Kim Laurio, Fredrik Linåker, Ajit Narayanan. 2002. Regular biosequence pattern matching with cellular automata. *Information Sciences* **146**:1-4, 89-101. [Crossref]
27. Tim Taylor. Creativity in Evolution 79-108. [Crossref]
28. Barry McMullin. 2000. John von Neumann and the Evolutionary Growth of Complexity: Looking Backward, Looking Forward. *Artificial Life* **6**:4, 347-361. [Abstract] [PDF] [PDF Plus]
29. J.-L. Beuchat, J.O. Haenni. 2000. Von Neumann's 29-state cellular automaton: a hardware implementation. *IEEE Transactions on Education* **43**:3, 300-308. [Crossref]
30. Hiroki Sayama. 1999. A New Structurally Dissolvable Self-Reproducing Loop Evolving in a Simple Cellular Automata Space. *Artificial Life* **5**:4, 343-365. [Abstract] [PDF] [PDF Plus]
31. Tomoyuki Yamamoto, Kunihiko Kaneko. 1999. Tile Automaton: A Model for an Architecture of a Living System. *Artificial Life* **5**:1, 37-76. [Abstract] [PDF] [PDF Plus]
32. Gianluca Tempesti, Daniel Mange, André Stauffer. 1998. Self-Replicating and Self-Repairing Multicellular Automata. *Artificial Life* **4**:3, 259-282. [Abstract] [PDF] [PDF Plus]
33. Moshe Sipper. 1998. Fifty Years of Research on Self-Replication: An Overview. *Artificial Life* **4**:3, 237-257. [Abstract] [PDF] [PDF Plus]
34. James A. Reggia, Jason D. Lohn, Hui-Hsien Chou. 1998. Self-Replicating Structures: Evolution, Emergence, and Computation. *Artificial Life* **4**:3, 283-302. [Abstract] [PDF] [PDF Plus]
35. James A. Reggia, Hui-Hsien Chou, Jason D. Lohn. Cellular Automata Models of Self-replicating Systems 141-183. [Crossref]
36. Moshe Sipper, Daniel Mange, André Stauffer. 1997. Ontogenetic hardware. *Biosystems* **44**:3, 193-207. [Crossref]
37. J.D. Lohn, J.A. Reggia. 1997. Automatic discovery of self-replicating structures in cellular automata. *IEEE Transactions on Evolutionary Computation* **1**:3, 165-178. [Crossref]
38. M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Perez-Uribe, A. Stauffer. 1997. A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. *IEEE Transactions on Evolutionary Computation* **1**:1, 83-97. [Crossref]