Self-Reproduction in Asynchronous Cellular Automata

Chrystopher L. Nehaniv

Adaptive Systems Research Group
Faculty of Engineering & Information Sciences
University of Hertfordshire
Hatfield, Hertfordshire AL10 9AB
United Kingdom

E-MAIL: `C.L.Nehaniv@herts.ac.uk`
WWW: `http://homepages.feis.herts.ac.uk/~nehaniv/welcome.html`

Abstract

Building on the work of Von Neumann, Burks, Codd, and Langton, among others, we introduce the first examples of asynchronous self-reproduction in cellular automata. Reliance on a global synchronous update signal has been a limitation of all solutions since the problem of achieving self-production in cellular automata was first attacked by Von Neumann half a century ago. Our results obviate the need for this restriction.

We introduce a simple constructive mechanism to transform any cellular automata network with synchronous update into one with the same behavior but whose cells may be updated randomly and asynchronously. This is achieved by introduction of a synchronization substratum which locally keeps track of the passage of time in a local neighborhood in a manner that keeps all cells locally in-step.

The generality of this mechanism is guaranteed by a general mathematical theorem (due to the author) that allows any synchronous cellular automata configuration and rule to be realized asynchronously in such a way the the behavior of the original synchronous cellular automata can be recovered from that of the corresponding asynchronous cellular automaton. Thus all important results on self-reproduction, universal computation, and universal construction, and evolution in populations of self-reproducing configurations in cellular automata that have been obtained in the past carry over to the asynchronous domain.

1. Introduction and Motivations

From the beginnings of the study of self-reproduction in artificial systems initiated by John Von Neumann in 1948, the primary formal model has been synchronous cellular automata in which configurations develop that eventually may include an unbounded number of copies of the original. The models constructed by Von Neumman and his successors have amply demonstrated that self-reproduction is indeed possible in artificial systems. Furthermore, Von Neumann's work on his automata models even anticipated the important transcription ("blind copying") and translation ("executability") properties of genetic material later found for DNA.

The different possibilities for achieving self-reproduction have implications for our understanding of the origin of life, the nature of organic life, and for the possibilities of life as it may exist elsewhere in the universe. Szathmáry (1999) offers a first classification of replicators applicable to natural and artificial systems. Moreover, self-reproduction is a prerequisite for any independent evolutionary process. Self-reproduction and self-repair (or self-maintenance) are often closely related in biology, and an understanding of self-reproduction can thus contribute to our ability to create self-repairing, self-maintaining hardware and software. Sending information, instructions on how to build copies of desired structures using local materials, into an environment rather than sending all necessary materials into that environment represents more economical methods of space exploration and colonization. See the NASA report edited by Freitas and Gilbreath (1980) for further potential examples and applications of self-reproduction to space science, e.g. self-replicating and self-maintaining lunar factories.

However the need for an assumption of synchronous update in these spaces may be questioned. In building an artificial self-reproducing entity is it really necessary to have a single global synchronization signal that reaches all parts of the entity simultaneously (or at least within a well-defined

Proceedings of the 2002 NASA/DOD Conference on Evolvable Hardware (EH'02)
0-7695-1718-8/02 $17.00 © 2002 **IEEE**

tolerance)? If local parts of the configuration are ready to change their state, is it realistic and practical to assume that they must wait until all other parts of the cellular space are also ready to update their states?

In this paper we free all cellular automata models of self-reproduction (as well as all cellular automata models of evolution, universal computation, and universal construction) from the need for synchronous update. This is accomplished by an elegant simple mechanism that allows one to construct an asynchronous cellular automaton that is capable of emulating the behavior of a given synchronous cellular automaton. State updates in the asynchronous model may be produced by any asynchronous update mechanism whatsoever (e.g. updates may be random, sequential, locally Poisson-distributed, partially simultaneous, etc., or even synchronous). This result for cellular automata is a special case of a more general theorem for automata networks with inputs due to the author (Nehaniv, 2002 (accepted)).

Here we describe the construction for making any cellular automata computation asynchronously realizable, give examples that illustrate how the use of "local time" frees the cellular automata from the need for global synchronization, and display the first examples of self-reproduction in cellular automata.

2. Preliminaries on Cellular Automata

Cellular automata were introduced by J. Von Neumman and S. Ulam to model natural physical and biological phenomena, in particular, for Von Neumann's pioneering studies of self-reproduction.

A *graph* Γ is a set of *nodes* V and a set of *edges* E. An edge $e \in E$ is an unordered pair of nodes $\{v, v'\} \subseteq V$. The *neighborhood* of a node is the set of nodes directly connected to it in the graph by an edge. That is, node v is a neighbor of v' if $\{v, v'\}$ is an edge in the graph. We shall write $v' \in nbhd(v)$ if node v' is a neighbor of node v.

A deterministic *finite state automaton* $\mathcal{A}$ is finite set of states Q, a finite set of inputs X and a transition function $\delta : Q \times X \to Q$.

A *cellular automaton* is a finite or infinite network of identical deterministic finite state automata and a graph structure such that:
(1) each node has the same (finite) number of neighbors,
(2) at each node we have a fixed ordering on the neigbhor nodes (e.g. north, south, east, and west if there are exactly four neighbors to every node), and
(3) the next state of an automaton at node v is always the same function δ of its current state and the current states of its neighbors. (Thus the ordered list of states at neighbor nodes is the input to the automaton at node v.) We shall denote the ordered list of the states of the neighbors of node v by $(q_w)_{w \in nbhd(v)}$.

Although cellular automata are more general, for our purposes, we may assume that the graph can be realized in some N-dimensional Euclidean space, for example as the set of points with integer coordinates, with edges connecting points that differ by at most exactly 1 in exactly one position ("the Von Neumman neighborhood") or, alternatively by at most 1 in each coordinate ("the Moore neighborhood"). For example, in a two-dimensional case the point (2,3) has neighbors (1,3), (3,3), (2,4), and (3,3) if our cellular automaton uses a Von Neumann neighborhood, and has additional neighbors (1,2), (1,4), (3,2), (3,4) if we instead have a cellular automaton with Moore neighborhoods. We may allow "wrapped around" or a "toroidal topology" by identifying nodes which differ by a fixed vector. For example, in a 25 × 50 node toroidal topology with Von Neumann neighborhood, node (1,1) has neighbors (1,2), (2,1), (25,1) and (1,50) since (25,1) is identified with node (0,1) and (1,50) is identified with (1,0).[1]

Such cellular automata network topologies are specific examples, but the methods presented here and the supporting mathematical results apply to all cellular automata networks regardless of the details of their particular topologies.

In addition there is usually assumed to be a *quiescent state* $q_0 \in Q$ in the local finite state automata such that if the automaton at node v is in state q_0 and all the its neighbors are in state q_0, then in the next time step the automaton at node v will still be in state q_0. (We will not have a strictly quiescent state in our asynchronous cellular automata.)

A *configuration* is any assignment of local state values to the set of automata at nodes in a finite subgraph of Γ.

2.1. Synchronous vs. Asynchronous Update Rules

Usually a cellular automaton $\mathcal{A}$ is required to update the state of the finite automata at all of its nodes simultaneously and in discrete steps. Thus, for all discrete times $t \geq 0$, if at discrete time step t each node v is in some state $q_v(t)$ then at the next discrete time step $t+1$ node v is in its next state $q_v(t+1)$. In the notation introduced above,

$$q_v(t+1) = \delta(q_v(t), (q_w(t))_{w \in nbhd(v)}).$$

Thus the new state $q_v(t+1)$ at node v is given by the local update rule as a function of $q_v(t)$, the current state of v, and the (finite) list $(q_w(t))_{w \in nbhd(v)}$ of all current states $q_w(t)$ of all nodes w in the neigborhood of v. In this case of globally simultaneous update, we say that the cellular automaton is *synchronous*. The *global state* of the cellular automaton $\mathcal{A}$ at time t is comprised of the states $q_v(t)$ of all its nodes

---

[1] We remark that these definitions of Von Neumann and Moore neighborhoods, and that of toroidal topology, are applicable to any dimension N.

at time t and can be regarded as a function from nodes V to local states Q.

If the updates of the local component automata are not required to take place synchronously, but each one will be updated to its next state an unbounded number of times as (locally discrete) time goes on, then we speak of an *asynchronous* automata network. The updates are otherwise unconstrained, e.g. they may be deterministic, non-deterministic, random, sequential, etc., or even synchronous. For further discussion of the relevance of synchronous and asynchronous cellular automata to the modelling of biological systems see for example (Schönfisch and de Roos, 1999). Prior to this paper, all published models of self-reproduction in cellular automata have used only synchronous cellular automata update rules.

Following the general method and theorem of (Nehaniv, 2002 (accepted)), which also applies to more general types of automata networks than cellular automata, for each synchronous cellular automaton $\mathcal{A}$ on graph Γ, we construct another particular cellular automaton $\mathcal{A}'$ on the same graph. Moreover, if the local finite automata in $\mathcal{A}$ have n states then component automaton in $\mathcal{A}'$ at each node v will have $3n^2$ states. The author's theorem guarantees that if $\mathcal{A}'$ is updated by any asynchronous method whatsoever, for each time t in the computation of $\mathcal{A}$, the global state of $\mathcal{A}$ at t is completely determined by a "continuous spatio-temporal section" of the behavior of $\mathcal{A}'$ (see below).

This mathematical theorem implies that all computations that can be carried out on any synchronous automata network can be recovered from the computation of an asynchronous automata network with no constraints on how the updates actually occur in the latter.

3. Background on Self-Reproduction in Cellular Automata

3.1. Models of Self-Reproduction

Von Neumann's original (1966) description of self-reproduction in cellular automata requires that the self-reproducing configuration be capable of universal computation (in Turing's sense) and of universal construction – loosely speaking, the ability to fill any compact area in the cellular space with any desired pattern. These properties were required in addition to the ability to make a copy of oneself, but could also be used to support this ability. Namely, *universal construction* (as the ability to fill any compact region of the cellular space with arbitrary configurations) guarantees that a copy of the self (including its 'instruction tape' which is present in many examples) can be constructed. His design of a self-reproducing universal computer and constructor was infeasibly large and has never been fully implemented and executed through a reproduction cycle on a computational device.

Langton's (1984) definition requires that a copy is constructed but not that either of Von Neumann's conditions be met. Langton implemented and studied the first example of feasible self-reproduction in cellular automata, using an 8-state cellular automaton with an initial configuration of 86 cells, that produces a first offspring after 151 time steps and then proceeds to fill up available space with copies. To avoid trivialities while avoiding the complexity of Von Neumann's model, *Langton's criterion* (1984, 1986) was proposed as a necessary condition on self-reproduction and requires that information is treated in two ways: as instructions that are *executed* ('translation') and as data which are blindly *copied* ('transcription'). These properties are also present in and abstracted from Von Neumann's and later Codd's (1968) examples, and by that time also known to be characteristic of biological self-reproduction. Subsequent examples of Byl (1989) and Reggia et al. (e.g., Reggia, Armentrout, Chou, and Peng (1993), Lohn and Reggia (1997)) simplified the self-replicating loop of Langton with fewer states or less cells in the initial configuration.

Subsequently, various researchers kept Langton's requirement for self-reproduction, but have added more and more computational power to the relatively small self-reproducing cellular automata configurations (in comparison to Von Neumann's solution). These trends are surveyed by Lohn (1999), who also describes the evolution of cellular automata rules that support self-reproduction (see also Lohn and Reggia (1997)). A fairly complete and up to date annotated bibliography with links to various relevant on-line resources can be found at Moshe Sipper's Artificial Self-Replication page.

H. Sayama (1998b, 1999) has constructed variants of the self-reproducing Langton loop which exhibit self-dissolution once they can no longer reproduce, thus freeing up space for reuse by progeny, and most interestingly, another similar variant called "evo-loop" which exhibits heritable variability in loop size and is subject to evolution via interaction among descendants of a common ancestor acting as a selective force (Sayama 1998a, 1999). Heritability, variability, and differential survival in an environment with limited resources are present in his evo-loop when run in finite spaces. Thus evo-loop appears to be the first convincing example of an evolutionary process occurring in cellular automata.

4. Self-Reproduction, Individuality, and the Heritability of Fitness

What constitutes self-reproduction?

The definition is not uncontroversial. We have already mentioned that Von Neumann required universal computation and universal construction in order to exclude trivialities, such as the example of spreading activation. Langton abstracted the properties of inherited information being both copied and executed.

E. F. Moore (1962) defines a configuration C to be *capable of self-reproducing n offspring by time t* if starting from the initial conditions of the entire cellular space at time $t = 0$ such that the set of all nonquiescent cells of the space is an array whose configuration is a copy of C there is a time $t' > t$ such that at time t' the set of all nonquiescent cells will then be contained in an array whose configuration includes at least n copies of C.

Lohn and Reggia (1997) give the following definition:

> "A configuration C is self-replicating if the following criteria are met. First, C is a structure comprised of more than one nonquiescent cell and changes its shape during its self-replication process. Second, replicants of C, possibly translated and/or rotated, are created in neighbor-adjacent cells by the structure. Third, there must exist a time t such that C can produce i or more replicants, for any positive integer i, for infinite cellular spaces (Moore's criterion). Fourth, if the self-replication begins at time t, there exists a time $t + \Delta t$ (for finite $\Delta t > 1$) such that the first replicant becomes isolated from the parent structure."

The issue of exactness of the copy is problematic since it is not desirable to exclude the possibility of variability. Variability among offspring is certainly present in biological systems, and, as Darwin showed us, is necessary for evolvability. Vitányi (1973) introduced sexual reproduction in celluar automata and Sayama (1998b), mentioned above, has demonstrated variability and (deterministic) evolution occurring in cellular automata. Encoding of heritable information in the shape of a configuration or using self-inspection might represent another feasible mode of encoding heritable variation in self-reproduction (cf. Laing 1977, Ray 1992, Morita and Imai 1997, Nehaniv 2002).

A discussion of the difficulties in formulating a rigorous definition of self-replicating or self-reproduction is given by Nehaniv and Dautenhahn (1998), who point out that even in accepted cellular automata models of self-reproduction there are rarely two copies of the original configuration present at exactly at the same time when reproduction is generally accepted to have occurred (e.g. in the Von Neumann or Langton models), and it is certainly not the case when the first offspring has been produced. The various copies of the configuration may be at different stages in their "lifecycles" and not have exactly the same configuration of states. Nehaniv and Dautenhahn (1998) suggest looser criteria on identity of copies to allow 'species' of non-exact copies to be acknowledged as offspring, and also loosen the restriction on the presence of copies all at the same time (e.g., offspring that have to grow into adults are still regarded as offspring even though they are never in exactly the same state of development as the parent.) Adequate formal definitions of "member of the same species" and of "individual" are still lacking in the sciences of the artificial, including the study of self-reproduction in artificial systems. Although these concepts are clearly fundamental to biological evolution, even within biology there is still on-going controversy and current research into appropriate definitions for these concepts.

Coming back to Darwin's ideas, some meaure of *heritability of fitness* is required for evolution to occur. With self-reproduction, the similarity of offspring to the parents and the similarities of the environments in which the replicators find themselves is enough to account for this. However, beyond the level of simple replicators, heritability of fitness requires more explanation, e.g. in considering multicellular lifeforms with differentiated cell types, subunits which are themselves replicators comprise populations within the body that are potentially subject to evolutionary pressures (Buss, 1987; Maynard Smith and Szathmáry 1995, Michod 1999, Michod and Roze 1999). For example, cancer is an example in which reproduction and evolution occur at the lower cellular level at the expense of the higher organismal one. Guaranteeing that the offspring are similar to the parent by suppression of freedom at the lower level in exchange for benefits is the first functionality required of any higher unit of fitness such as a multicellular organism. The latter must employ mechanisms to balance the tendency of the lower level to defect by sufficient benefits from cooperation in the higher level unit, in order to persist over evolutionary time (Michod and Roze 1999).

In asynchronous self-reproduction the very fact that the relative synchronization of the entire state of the "organism" is uncertain contributes to this problem of heritability of fitness.

5. Construction of Equivalent Asynchronous Models

The construction of the local automata of the asynchronous cellular automaton $\mathcal{A}'$ from local automata of the synchronous cellular automaton $\mathcal{A}$ is extremely straightforward: Suppose the local automaton of $\mathcal{A}$ has states $Q = \{q_0, \ldots, q_{n-1}\}$ with q_0 quiescent and update function $\delta : Q \times X \to Q$.

The states of the local automaton in $\mathcal{A}'$ are the $3n^2$ states $Q \times Q \times \{0, 1, 2\}$.

For $r \in \{0, 1, 2\}$, we say the neighborhood of a node v in the asynchronous cellular automaton $\mathcal{A}'$ is $\mathit{ready}(r)$ if none of v's neighbors is in a state with third component equal to $r + 2 \pmod 3$).[2] Recall that we write $w \in nbhd(v)$ if w is a node in the neighborhood of v.

The update rule of this local automaton is given as follows: suppose node v is in state (q, q', r) with q and q' in Q, and $r \in \{0, 1, 2\}$, and has neighborhood with list of states $(q_w, q'_w, r_w)_{w \in nbhd(v)}$, then, if $r = 0$, the next state of node v is

$$\delta'((q, q', 0), (q_w, q'_w, r_w)_{w \in nbhd(v)})$$

$$= \begin{cases} (\delta(q, (q''_w)_{w \in nbhd(v)}), q, 1) & \text{if the neighborhood of } v \text{ is } ready(0) \\ (q, q', 0) & \text{otherwise,} \end{cases}$$

$$\text{where } q''_w = \begin{cases} q_w & \text{if node } w \text{ is in a state of the form } (q_w, q'_w, 0) \\ q'_w & \text{if node } w \text{ is in a state of the form } (q_w, q'_w, 1). \end{cases}$$

(Since the neighborhood is $ready(0)$, these are the only possibilities. Note the use of the original local transition function δ of the synchronous cellular automaton $\mathcal{A}$ in this case in the definition of δ'.)
For the remaining cases with $r \in \{1, 2\}$, the next state is

$$\delta'((q, q', r), (q_w, q'_w, r_w)_{w \in nbhd(v)})$$

$$= \begin{cases} (q, q', r + 1 \pmod 3) & \text{if the neighborhood of } v \text{ is } ready(r) \\ (q, q', r) & \text{otherwise.} \end{cases}$$

The state (q, q', r) can be thought of as encoding the following information: The first coordinate q shows the "visible" state of the corresponding network $\mathcal{A}$ at this node. The second coordinate q' is "hidden" and is used to remember the most recent old state of this node, in case any neighbor needs to refer to it. The third coordinate is used to locally synchronize updates of the visible and hidden nodes.

The above rule δ' only allows a local state of $\mathcal{A}'$ to change if no node in the neighborhood will get more than one step behind if the update were to be made, otherwise it allows no change at all to the current state. The important $ready(0)$ and $r = 0$ case occurs exactly when the asynchronous cellular automaton is locally ready to emulate the transition of the synchronous cellular automata at this local node using the local transition function of $\mathcal{A}$. Intuitively, the third component of a node v's state can locally distinguish 'present', 'future', and 'past' for neighboring nodes, respectively, by whether they have modulo 3 value equal to, one more than, or one less than the value of the third component of v's state.

5.1. Properties of the Emulation

The asynchronous cellular automaton has two important properties established with detailed mathematical proofs in the main theorem of (Nehaniv, 2002 (accepted)):

Suppose the synchronous cellular automaton $\mathcal{A}$ is started in configuration C with node v in state $C(v)$ and all other cells quiescent. Also suppose that asynchronous cellular automaton $\mathcal{A}'$ is started with each node v of the configuration active in state $(C(v), C(v), 0)$ and all other nodes in state $(q_0, q_0, 0)$.

Freedom from Deadlocks. At each node v of the asynchronous network, if the state of v has third component $r \in \{0, 1, 2\}$ eventually the neighborhood of v will be $ready(r)$, and the third component will change value to $r + 1 \pmod 3$.

Existence of Continuous Spatio-Temporal Sections. The state $q_v(0)$ at time 0 of node v of $\mathcal{A}$ is equal to the first component of the state of node v of $\mathcal{A}'$ in its initial configuration. By mathematical induction, one shows that the state $q_v(t)$ of node v in $\mathcal{A}$ at time t is equal to the first component of the state of node v in $\mathcal{A}'$ on the t^{th} time that the third component of node v becomes 1.

The latter property implies that it is possible to completely recover the behavior of $\mathcal{A}$ from the behavior of $\mathcal{A}'$ by simply recording the first components in the initial configuration of $\mathcal{A}'$ and recording the first component of each node v whenever its local automaton makes a transition so that the third component changes from 0 to 1.

5.2. Temporal Waves

To an observer, time as it occurs in the synchronous cellular automata may seem to pass at different rates in different parts of the asynchronous cellular space. In fact, locally all neighboring cells are guaranteed to show a visible state (first component) that occurs at most one unit of time in the past or future relative to the corresponding state in the synchronous automaton. Thus, time of the synchronous cellular automaton is emulated in a manner such that it can never get very far out of sync locally in the emulating asynchronous cellular automaton.

This results in waves of temporal update in the cellular space with continuous wavefronts all in the same state of temporal synchronization (i.e., all with the same third component) representing the same moment in the synchronous cellular automaton.

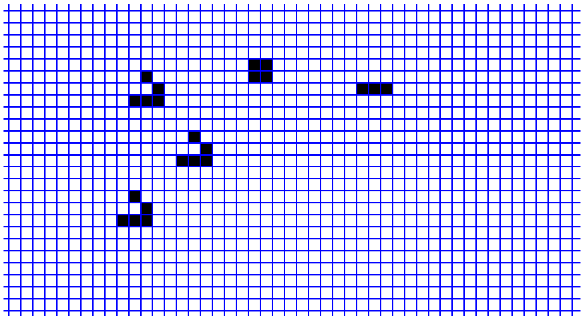
This is illustrated below with an asynchronous version of John Conway's famous "Game of Life".

---

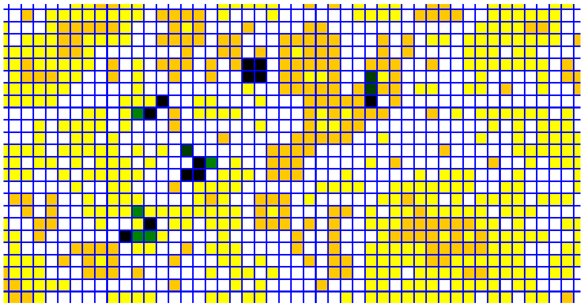
[2] For any integer n, "$n \pmod 3$" denotes the least nonnegative integer k such that $n - k$ is divisible by 3. Of course k must then be one of $\{0, 1, 2\}$.

ASYNCHRONOUS GAME OF LIFE

Initial State:



Progress of Gliders in Asynchronous Life. Note that the upper left hand glider is not recognizable as one due to small local temporal variation in its cells:



Further Progress of Gliders in Asynchronous Life. All their parts are nearly in the same spatial-temporal section; all three gliders are now recognizable again:

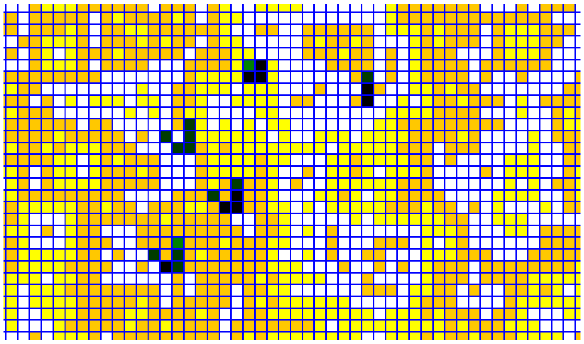


Figure 1. Temporal Waves and Progression of 3 Gliders, with Box, and Blinker in Asynchronous Game of Life. Contiguous regions of the same shade are "temporal wavefronts" that represent the same moment in a spatio-temporal section giving the global state of the corresponding synchronous cellular automaton.

5.3. Asynchronous Example: Conway's Game of Life and Universal Computation

Let us apply the construction to Conway's (synchronous) Game of Life. A local automaton in synchronous Life has two possible states (quiescent (0) or alive (1)) and the following transition function: if a cell is quiescent and has exactly 3 neighbors that are alive, its next state is alive. If a cell is alive, and it has either 2 or 3 live neighbors (not including itself) then it stays alive, otherwise it becomes quiescent. It has been shown that, in principle, universal computation can be implemented in a infinite two-dimensional (synchronous) cellular automaton running Conway's rule (for an enjoyable yet highly readable and detailed overview see chapter 1 of Sigmund (1995)).

Figure 1 (top panel) shows an initial configuration of some well-known structures in Conway's Game of Life as an initial configuration for the corresponding asynchronous cellular automtaton: Three gliders which move across the space, a stable 2 × 2 box, and a blinker (a row of 3 cells, that becomes a column of 3 cells, then a row of 3 cells, and so on).

The next panel shows the state of the world a few time steps later, the shading indicates the synchronization state of the cell in the space, while the darker cells of various shades are live cells in various stages of temporal synchronization. Contiguous cells of the same shade are in sync and reflect the same instant of time in the synchronous cellular automaton. The third panel down shows the state of the system a little later.

The possibilty of implementation of Conway's Game of Life in an asynchronous cellular automaton as illustrated here entails that universal computation is possible in a two-dimensional asynchronous cellular automata running the modified rules.

6. Self-Reproduction in Asynchronous Cellular Automata

Using this method it is now straightforward to construct asynchronous cellular automata that exhibit self-reproduction. Taking any of the models of Von Neumann, Codd, Langton, Byl, Reggia et al., Sayama, etc., we merely apply the construction above. Now we give the first implemented example of self-reproduction in asynchronous cellular automata by applying our construction to Langton's self-reproducing loop.

6.1. Asynchronous Langton Loop

Using the rule of Langton (1984) for a synchronous cellular automaton exhibiting self-reproduction, we derive by the method above an asynchronous cellular automaton with

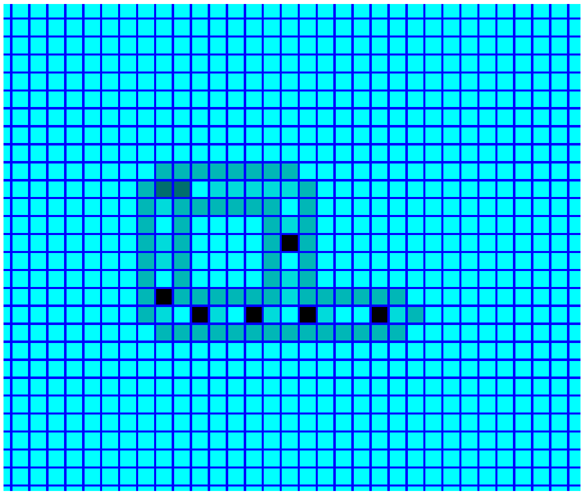


Figure 2. Asynchronously Self-Replicating Loop Soon after Starting Reproduction Cycle.

$192(= 3 \times 8^2)$ states possible in the local automata at each node. Using random asynchronous update of the nodes we achieved an implementation of self-reproduction in an asynchronous cellular automaton. Figure 2 shows the state of the cellular automaton near the beginning of the first reproduction cycle: a sheath of cells encloses an asynchronously counterclockwise circulating stream of instructions to extend a construction arm and turn left. The instructions are copied as they flow through a fork in the sheath. After this stream has been executed four times, an offspring asynchronous loop is present in the cellular space (right in Figure 3). Figures 4 and 5 show the state of the space at later times when more descendants of the original loop are present. As with Langton's synchronous loop, an asynchronous loop which has no more space to reproduce in eventually "dies", leaving a configuration of inert cells.

The rules and initial configuration for Langton's synchronous self-reproducing cellular automaton model are given in (Langton, 1984) and are widely available electronically in various synchronous implementations easily downloadable from the internet. From these rules and configuration, the reader can reproduce our experimental results (implemented here in java) using the explicit method given above in the previous section.

7. Discussion and Conclusions

We have demonstrated the systematic implementation of asynchronous cellular automata that fully emulate the behavior of any synchronous cellular automata. This was used to give the first examples of self-reproduction in asynchronous cellular automata.

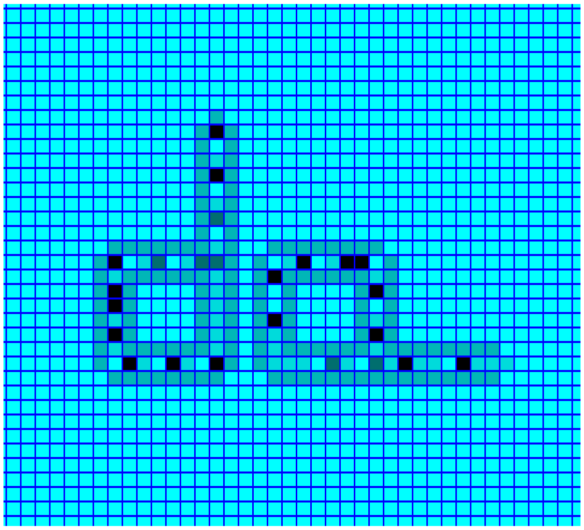


Figure 3. Asynchronously Self-Replicating Loop after First Reproduction.

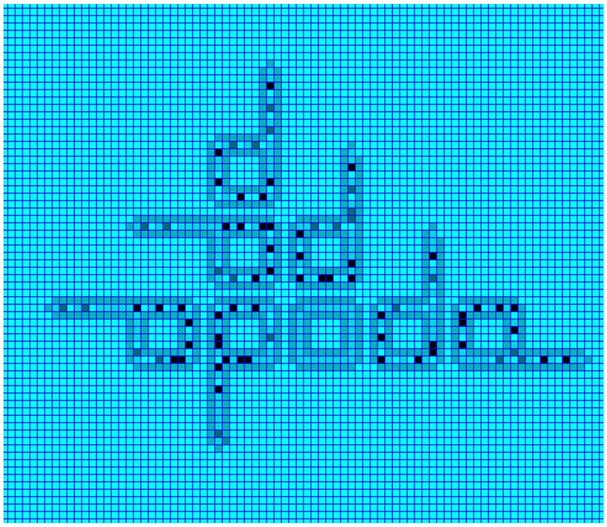


Figure 4. Asynchronously Self-Reproducing Loop after 7 Descendants have been Produced; Original Loop No Longer Active.

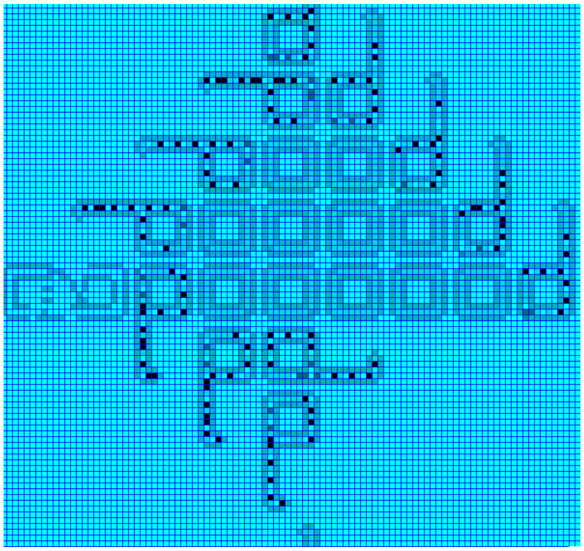


Figure 5. Asynchronously Self-Reproducing Loop after 24 Descendants have been Produced; Original Loop and Some Other Descendants No Longer Active. (100 × 100 Torus Topology).

Similarly, implementing the asynchronous version of Sayama's evo-loop we also have created the first instance of evolution in a population of self-replicators in an asynchronous cellular automaton (Nehaniv, 2002 (submitted)).

Universal computation and universal construction can be implemented in asynchronous cellular automata by applying our construction to Von Neumann's self-reproducing configuration. Universal computation can also be realized by our method of implementing Conway's Game of Life using asynchronous cellular automata as discussed above.

In general, most results about synchronous cellular automata carry over automatically now to the asynchronous realm. Thus these methods free those using cellular automata models of computation, self-reproduction, and evolution from the restriction of synchronous global update. With asynchronous (e.g. random) update, the same results are attained as guaranteed by the theorem of (Nehaniv, 2002 (accepted)).

Fault-tolerance and self-repair in the asynchronous cellular automata is a natural next direction to explore. A generalization of the methods presented here to cellular automata networks of variable size and shape, i.e. allowing dynamic growth or changing topology of the cellular space, would also a desirable development.

References

[1] Buss, L. W., *The Evolution of Individuality*, Princeton University Press, 1987.

[2] Byl, J. "Self-Reproduction in Small Cellular Automata", *Physica D*, Vol. 34, pp. 295–299, 1989.

[3] Burks, A. W. (ed.), *Essays on Cellular Automata*, University of Illinois Press, Urbana, Illinois, 1970.

[4] Codd, E. F. *Cellular Automata.* Academic Press, New York, 1968.

[5] Freitas, R. A., Jr., and Gilbreath, W. P. (eds.) *Advanced Automation for Space Missions, Proceedings of the 1980 NASA/ASEE Summer Study; sponsored by the National Aeronautics and Space Administration and the American Society for Engineering Education.* NASA Conference Publication 2255. On-line extracts at `http://www.zyvex.com/nanotech/selfRepNASA.html` and `http://www.islandone.org/MMSG/aasm/` .

[6] Laing, R. "Automaton Models of Reproduction by Self-Inspection", *Journal of Theoretical Biology*, Vol. 66, 437–456, 1977.

[7] Langton, C. G. "Self-reproduction in Cellular Automata", *Physica D*, Vol. 10, pp. 135–144, 1984.

[8] Langton, C. G. "Studying Artificial Life with Cellular Automata", *Physica D*, Vol. 22, pp. 120–149, 1986.

[9] Lohn, J. D. "Self-replicating Systems in Cellular Space Models". In C. L. Nehaniv (ed.), *Mathematical and Computational Biology: Computational Morphogenesis, Hierarchical Complexity, and Digital Evolution*. Vol. 26 in Lectures on Mathematics in the Life Sciences, Providence, Rhode Island, American Mathematical Society, pp. 11–30, 1999.

[10] Lohn, J. D. and Reggia, J. A. "Automatic Discovery of Self Replicating Structures in Cellular Automata", *IEEE Transactions on Evolutionary Computation*, Vol. 1 No. 3, pp. 165–178, 1997.

[11] Maynard Smith, J. and Szathmáry, E. *The Major Transitions in Evolution.* W. H. Freeman, 1995.

[12] Michod, R. E., *Darwinian Dynamics: Evolutionary Transitions in Fitness and Individuality*, Princeton University Press, 1999.

[13] Michod, R. E. and Roze, D. "Cooperation and Conflict in the Evolution of Individuality. III. Transitions in the Unit of Fitness". In C. L. Nehaniv (ed.), *Mathematical*

and Computational Biology: Computational Morphogenesis, Hierarchical Complexity, and Digital Evolution. Vol. 26 in Lectures on Mathematics in the Life Sciences, Providence, Rhode Island, American Mathematical Society, pp. 47–91, 1999.

[14] Moore, E. F., "Machine Models of Self-Reproduction", *Proceedings of the Fourteenth Symposium on Applied Mathematics*, American Mathematical Society, pp. 17–33, 1962. (Reprinted in Burks (ed.), 1968.)

[15] Morita, K. and Imai, K. "A Simple Self-Reproducing Cellular Automaton with Shape-Encoding Mechanism". In C. G. Langton and K. Shimohara (eds.), *Artificial Life V*, MIT Press, pp. 489–496, 1997.

[16] Nehaniv, C. L. "Internal Constraints and Ecology in Evolution: A Case Study in Tierra", in *Proceedings of the Fifth German Workshop on Artificial Life (GWAL V)*", pp. 243-252, 2002.

[17] Nehaniv, C. L. "Asynchronous Automata Networks Can Emulate Any Synchronous Automata Network", *International Workshop on Semigroups, Automata, and Formal Languages (June 2002 – Crema, Italy)*, 2002 (accepted). *journal version in preparation.*

[18] Nehaniv, C. L. "Evolution in Asynchronous Cellular Automata", 2002 (submitted).

[19] Nehaniv, C. L. and Dautenhahn, K., "Self-Replication and Reproduction: Considerations and Obstacles for Rigorous Definitions", *Proceedings of the Third German Workshop on Artificial Life (GWAL III)*, pp. 283–290, 1998.

[20] Ray, T. S., "An Approach to the Synthesis of life". In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, Volume X of SFI Studies in the Sciences of Complexity, pp. 371–408, Redwood City, CA, 1992. Addison-Wesley.

[21] Reggia, J. A., Armentrout, S., Chou, H. H., Peng, Y., "Simple Systems that Exhibit Self-Directed Replication", *Science*, Vol. 259, pp. 1282–1288, 1993.

[22] Sayama, H. *Constructing Evolutionary Systems on a Simple Deterministic Cellular Automata Space*, Ph.D. Dissertation, Department of Information Science, Graduate School of Science, University of Tokyo, December 1998.

[23] Sayama, H., "Introduction of Structural Dissolution into Langton's Self-Reproducing Loop", *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor, eds., pp. 114-122, 1998, MIT Press. On-line material at: `http://necsi.org/postdocs/sayama/sdsr/`

[24] Sayama, H. "A New Structurally Dissolvable Self-Reproducing Loop Evolving in a Simple Cellular Automata Space", *Artificial Life*, Vol. 5, no. 4, pp. 343-365, 1999.

[25] Schönfisch, B. and de Roos, A. "Synchronous and Asynchronous Updating in Cellular Automata", *BioSystems*, Vol. 51, 123–143, 1999.

[26] Sigmund, K. *Games of Life*, Penguin, 1995.

[27] Sipper, M. The Artificial Self-Replication Page `http://www.cs.bgu.ac.il/∼sipper/selfrep/`

[28] Szathmáry, E . "Chemes, Genes, Memes: A Classification of Replicators". In C. L. Nehaniv (ed.), *Mathematical and Computational Biology: Computational Morphogenesis, Hierarchical Complexity, and Digital Evolution*. Vol. 26 in Lectures on Mathematics in the Life Sciences, Providence, Rhode Island, American Mathematical Society, pp. 1–10, 1999.

[29] Vitányi, P. M. B. "Sexually Reproducing Cellular Automata", *Mathematical Biosciences*, Vol. 18, pp. 23–54, 1973.

[30] Von Neumann, J. *Theory of Self-Reproducing Automata.* Edited and completed by A. W. Burks, University of Illinois Press, Illinois, 1966.