Genetic Game of Life

John S McCaskill and Norman H Packard

European Centre for Living Technology

Abstract

The family of semi-totalistic cellular automata (CA) that include Conway’s Game of Life (GoL) is extended to allow evolution by the association of genetic information with individual live cells in the binary CA, allowing increasingly fine distinctions between neighbour configurations. A newly born live cell’s genome is copied (potentially with mutation) from one of the live neighbour cells (there are three in the GoL) and is destroyed when the cell dies. We investigated evolution for four symmetry cases in the nearest neighbour rule tables: semi-totalistic, corner-edge totalistic, 8-rotation symmetric, and physical 2D symmetric (4-rotations and 4-reflections). We discovered a family of deterministic rules to replace genetically stochastic choices of ancestor, making our model deterministic (like the GoL) in the absence of mutation. Just as biological evolution constantly discovers innovations in the space of chemical and physical functionalities that it controls, we seek to explore how the addition of genetic information to the game of life can display the same type of discovery process, where the genetically controlled innovations are now spatially local modifications to the GoL local dynamical rules. Complex cellular automaton dynamics of the game of life become the default “chemistry and physics”, and local genetic modifications that persist are the innovations discovered by evolution. In the work presented here, systematic genetic variations near to the game of life rule are investigated and found to produce signs of computational complexity with an abundance of glider and glider gun structures. We investigated the impact on evolution for both neutral selection and genetically determined fecundity and mortality including population interaction effects. The dynamics are analysed by a novel version of activity statistics and by fast ongoing genealogy construction as well as by fast glider detection relying on bit-parallel 64-bit integer algorithms.

# Introduction

Norman you edited both v2 and more recently v1-red. I am transferring your changes and making improvements to the hot copy v2.

Currently also making transition to Endnote References.

Interest in cellular automata (CA) as models of emergent complexity began with von Neumann’s famous 29-state cellular automaton capable of universal construction [von Neumann Ref 1] and became widespread with Conway’s discovery of the simple two-dimensional (2D) Game of Life CA (GoL) [Conway Ref 2] and Wolfram’s analysis of computational complexity classes in 1D CAs [Wolfram Ref 3] which was extended to 2D CAs [Packard Ref 4]. Both von Neumann’s and Conway’s CAs are defined with strong relaxation to the ground state in order to facilitate computation by rational design, in fact universal computation has been proven by construction in both cases [*Authors?* Ref 5,6], but both involve fundamentally unprotected computations in the open 2D square lattice, in which perturbations, in the form of even the simplest travelling patterns, will almost certainly destroy not only the computed result but the carefully crafted computing architecture as well. Because of this, and despite the widespread continuing interest in novel computational structures in the GoL and related CAs, there is a major jump to evolving systems in which computation needs to survive robustly in the presence of potential interactions with many competitors. This paper is concerned with bridging this gap and we stay deliberately as close as possible to the deterministic CAs that underlie computation in the GoL by retaining completely deterministic computation apart from random and rare mutational changes, while supporting locally determined genetically encoded rule changes that enable evolution.

As analysed by Eppstein [Eppstein Ref 7], the family of semi-totalistic 2D binary CAs (with the next state of a cell depending only on the state and the number of its live neighbours and not on their detailed configuration) can perhaps better be characterized by the …

[this paragraph should discuss the background of computational complexity classes as it applies to 2D CAs. It should include reference to e-machines and Crutchfield [Crutchfield Ref 8].

[Another paragraph is required to explain the relationship with spatial games (Nowak) and with CAs used to describe evolution in reaction diffusion systems [Boerlijst and Hogeweg Ref 9] and in cellular and amphiphile models (Ising and Potts models [Ikegami and Hogeweg Ref 10, McCaskill, Packard et. al. Ref 11]. In this paragraph reference the other GPU paper on GoL plus evolution and contrast with adding additional states/colors to GoL (e.g. immigration game).

Conway’s Game of Life (GoL) [Conway Ref 2, *Modern Author* Ref 12] is a deterministic dynamical system that takes two dimensional spatial patterns of binary states (‘live’ or ‘dead/empty’) to new patterns as time progresses discretely, through the action of a local rule; Each site’s state at time *t+1* is dependent on its state and the states of its nearest neighbours at time *t*. The game of life is a semi-totalistic cellular automaton (CA) rule, as defined above, and can be denoted by the code B3/S23 in standard notation [[Eppstein](https://www.ics.uci.edu/~eppstein/ca/wolfram.html) Ref 7, + Golly Ref 13]. A *live* cell *survives* (a state “1” at a site at time *t* persists to time *t+1*) if there are either 2 or 3 neighbours *alive* in the 8 cells surrounding the site on a square lattice (otherwise *dying*, *i.e.* changing to state “0” at time *t+1*), and a *dead* or *empty* “0” state cell undergoes birth (transitions to “1” at time *t*+1) only if there are exactly 3 neighbours *alive* at time *t*.

Conway’s Game of Life has become a canonical example of a complex system, with simple local rules that produce complex dynamics. It has a rich phenomenology of dynamics from special initial conditions, documented in massive catalogue projects [Ref 8,9] and other articles [Ref 10,11,12]. Specially engineered initial states can have extremely long transients, occupying large regions of space, and indeed the Gol has been shown to support universal computation [Ref 6,7].

Notwithstanding these properties, the GoL has not been a good model for studying the *emergence of complexity*, for two main reasons. The first is that starting from random initial state patterns on a finite compact domain, it is well known that the GoL almost always settles down to a combination of isolated static and simply periodic structures which are individually of limited spatial extent [Ref 4,5]. The second is that the complexity generated by the GoL is not robust, in the sense that perturbations destroy functionally complex structures. Even when a complex dynamical structure happens to be produced by a random initial condition, it is typically destroyed by glider that perturbs it.

In biology, genetics is coupled to real-world physics and chemistry, enabling evolution to produce a complex biosphere. In the present work, we use the GoL to provide an interesting model of a rich but very simple “physics” or “chemistry”. We then enhance the GoL to include genetic information, with the aim of understanding how complexity may emerge from this simple version of evolution.

The coupling of GoL to genetic information has already been attempted in various ways [Refs], but a systematic investigation of near GoL evolving dynamics is still outstanding. We add genetics to the GoL by associating a genome with all live cells. Genetic inheritance is ensured by a newly born live cell’s genome being copied (potentially with mutation and recombination) from one or more of the live neighbour cells (there are three in the GoL) and being deleted when the cell dies. In this article we focus on the simplest case of mutation and asexual reproduction without recombination.

The genome of a live cell can encode local departures from the GoL rule, making the system spatially inhomogeneous cellular automaton. For the specification of a genetic GoL system, we must:

* Specify how departures from the GoL are determined by genes
* Specify how genes are propagated from one time step to the next

Starting from random initial state patterns on a finite compact domain, it is well known that the GoL almost always settles down to a combination of isolated static and simply periodic structures or gliders which are individually of limited spatial extent [Ref 5,6]. Although specially engineered initial states can have extremely long transients, occupying large regions of space, and indeed the Gol has been shown to support universal computation [Ref 7,8], the absence of complex interconnected pattern persistence starting from random initial conditions means that it is not a good candidate for the emergence of complexity.

However, because of its rich dynamics from special initial conditions, documented in massive catalogue projects [Ref 9,10] and other articles [Ref 11,12], it would appear to provide an interesting model of a rich but very simple “physics” or “chemistry” that may be coupled to biological evolution through genetic information. The coupling of GoL to genetic information has already been attempted in various ways [Ref 13-15], but a systematic investigation of near GoL evolving dynamics is still outstanding. Here, we enhance the deterministic GoL dynamics to create an evolutionary system, by associating a genome with all live cells and deriving deterministic inheritance rules that may be genetically neutral or sequence dependent. Genetic inheritance is ensured by a newly born live cell’s genome being copied (potentially with mutation and recombination) from one or more of the live neighbour cells (there are three in the GoL) and being deleted when the cell dies. In this article we focus on the simplest case of mutation and asexual reproduction without recombination.

[We can examine genetics associated with the unmodified cellular automaton dynamics.]

The genome of a live cell can encode local departures from the GoL rule, making the system a spatially inhomogeneous cellular automaton. For the specification of a genetic GoL system, we must:

* Specify how departures (if any) from the GoL next states are determined by genes
* Specify how the genes attached to live states are propagated during birth (and possibly during survival steps, by overwriting)

This investigation is motivated by a fundamentally interest in the interaction between computational complexity and evolution. Whereas the GoL’s absence of

# Genome propagation

## Movement vs copying

Whereas the indistinguishability of GoL “1” states means that it is not possible to distinguish movement from death and rebirth, with genetic information attached to the live states, it would be possible. Is there a meaningful assignment of a subset of GoL birth or survival transitions to movement? It would make a difference if mutation were deemed not to occur for transitions involving movement. Also, it might be appropriate to make the choice of an ancestor sensitive to the interpretation of movement *vs* birth: e.g. to minimize the number of births needed to maintain the dynamics. For example, an isolated rod of three live states is a GoL oscillator between vertical and horizontal configurations. In the deterministic most different ancestor canonical assignment of ancestors from three live neighbours, the central gene is copied to two new sites so that (without mutation) the rod becomes genetically homogeneous in one step. This process is clearly a copy mechanism. On the other hand, in the 0-bit canonical assignment of ancestors, the two peripheral genes circulate anti-clockwise and this is more naturally understood as a process of motion and as such should be carried out without mutation.

# Indirect local rule modifications

Regarding the transfer of genetic information, a deterministic selection mechanism is realized, so that mutation provides the only source of random variation in the dynamics. Four levels of perturbations on the B3/S23 standard GoL rules are investigated: (0) genetic selection on the GoL, with the genes not influencing the GoL rules (1) genetic selection on GoL-like rules, e.g. S2gb3gB2g3 ∈ {S(2g(b))(3g(b))/B(2g)3(g)}, with conditional rules depending on genes denoted by g and survival rules involving birth overwrites denoted by b (2) genetic modulations of the live neighbour counting process that allow different numbers of live neighbours a) for specific 2nd neighbour ring configurations b) using masks on 1st neighbours encoded by the existing live neighbours c) with the gene specifying the allowed numbers directly (3) limitations of the influence of arbitrary gene encoded rule departures in varying symmetries through the requirement that state changes induced by a non-GoL rule inhibit further rule departures in their neighbourhood until corrected by a regular GoL rule. Whereas many perturbations of the GoL rules either quickly die out or proliferate rapidly to fill space with reproducing structures, a family of interesting dynamical systems is found, and its tendency towards open-ended evolution analysed by means of activity statistics. Note, that while the standard B3/S23 game of life, starting from random patterns of 50% 1s and 0s, in most instances does not produce complex dynamics, the new family does.

Notation: Rule extensions  [Golly](https://en.wikipedia.org/wiki/Golly_(program)) open-source cellular automaton package

## Example: selection = 7

## Example: selection = 0

## Example: selection = 2

## Example: selection = 4

## Example: selection = 5

# Direct encoding of local CA rule by genes

The most common family of CAs within which GoL-like rule-tables are defined is the semi-totalistic or outer-totalistic automata family, so-called because a cell’s next state depends only on the current state and sum of the neighbour states *s* (totalistic rules depend only on the sum of the current state and all neighbour states). The birth rules for current state “0” and the survival rules for current state “1” define the exceptions to the default rule which is next state “0”.

Genes may specify any look up table (LUT) that maps the sum of neighbors to cell values 0 or 1, and the LUT construction may be generalized if we go beyond the semi-totalistic family, as we shall later on. For the semi-totalistic case, if we exclude the special cases of spontaneous birth (B0 *i.e.* birth for s=0) and lone survival (S0) then there are 8+8=16 distinguished states that may be independently part of an active next state ruleset. Thus there are 216 CA rulesets, and these may be encoded by a binary genome of length 16 with one bit per LUT entry. In this paper, we restrict our attention to genes of maximum length 64, and often use the words gene and genome interchangeably to refer to the full sequence, only rarely using the term gene to refer to a specifier of part of the rule-table. We may also employ multiple bits (ncoding) to encode each LUT entry for an active rule, for example with only one of the possible ncoding gene patterns being active, then genomes of length 16x ncoding are required. A modular variable length encoding could be to encode the values of s (requiring 3 bits) as well as the central state (for survival or birth) for which the next state is live i.e. 4 bits in total per entry. The standard game of life would require 3\*4 = 12 bits to be specified. Longer genomes could contain the same entry repeatedly allowing for mutational error resistance.

Requiring that the birth and survival rules form a single interval of neighbourhood sum values, with lower and upper limits in the sum variable *s,* restricts the possible rule-tables to a family with members specified by four integers SlSuBlBu (lower and upper values of the neighbourhood sum for survival and birth, respectively) and restricts the rule space above to 362=1296 possible rules. Most of these rule-tables either lead to strong proliferation of live states or their extinction, and in order to allow genetic encoding to deliver novel dynamics of interest it turns out to be important to further dissect the rule-tables in the vicinity of the GoL rule 2333.

There are a number of ways that this may be done, taking spatial symmetries into account. We shall list several and investigate a subset of them:

1. Distinguish arbitrary configurations ignoring symmetries: there are 256 binary neighbour patterns in the Moore neighbourhood, and hence 2512 possible rule-tables. We could encode these directly with a 512-bit genome or with a variable length genome by a sequence of 9-bit patterns specifying configurations (including the central site as a 9th bit) that give rise to live states. A maximum of 7 such rules would fit in a 64-bit genome and the standard GoL genome would have to be 9\*(C82+2\*C83)= 9\*(28+56)=756 bits in length. For completeness, we could also consider the continuous interval subset of rules defined by upper and lower bounds for the integer value of the 8-bit neighbor configuration for survival and birth, for which rule-tables can be specified by 2\*(8+8)=32 bits.
2. Distinguish all configurations that are not equivalent by 4-fold rotation and reflection symmetry in 2D. The numbers of these distinguished configurations for s=0…8 are 1,2,6,10,16,10,6,2,1 *i.e.* in total 54, or 52 excluding the s=0 case. The minimum length encoding of a gene specifying this table with one bit per LUT entry is thus 104. Encoding with a variable length genome analogously to above would result in 3+4+1 bit patterns needed to specify a LUT entry, and the standard GoL can be encoded with 8\*(6+2\*10)=208 bits. 64-bit genomes would allow up to 8 such entries to be specified (*e.g.* 4 survival and 4 birth configurations) in a variable length genome. Note that the variable genome length encoding allows the rule-tables of different genomes to be combined simply: any entry that is defined is valid irrespective of multiplicity.
3. Distinguish all configurations that are not an 8-fold rotation of one another. This is both a simpler and somewhat smaller symmetry partition of the 8-neighbour configurations as *s* varies from 0..8 with 1,1,4,7,10,7,4,1,1 a total of 36 distinguished configurations. Configurations can be quickly mapped to their symmetry class, by finding the rotation of 8 bits that yields the minimal numerical value (which we call the canonical rotation) and using these values as identifiers for the different classes (separately for each value of *s*). Restricting attention to the central s-range of 2-6 gives 32 distinguished configurations and 64 LUT entries for survival or birth. This fits neatly into a 64-bit integer genome. The variable length genome encoding still requires 8-bit entries and the standard GoL rule can be specified in 8\*(4+2\*7)=144 bits.
4. Distinguish the four diagonal or corner sites (NW,NE,SE,SW) from the four edge-centered sites (N,E,S,W) in an otherwise semi-totalistic rule-table. The numbers of partitions of *s* into these two classes of sites are for *s*=0…8 1,2,3,4,5,4,3,2,1 = 25 *i.e.* 2\*24=48 bits are required for a direct fixed gene length encoding. The variable length genome encoding above requires 7-bit entries (excluding as before the s=0 entries) and the GoL standard rule is encoded in 7\*(3+2\*4)=77 bits.
5. The semi-totalistic case, distinguishing configurations only by *s*. In addition to the 16-bit gene encoding (excluding *s*=0 entries) there is also the variable gene length encoding involving entries of length 4-bits each. The GoL rule can be specified in 12-bits as described above.

In this work, we find that the semi-totalistic case is too coarse an encoding of CA rules, to allow significant genetic evolution of complex structures beyond the classic game of life. We explore both the fixed length and variable length encodings. Focussing initially on cases (iii) and (iv), rather than the more differentiated (ii), we investigate the more differentiated rules and find that they produce a range of interesting dynamics. When coupled to a genetic population, these differentiated rules rapidly evolve to proliferate unless very strongly constrained.

The key focus in this paper is on sequence- and possibly population-dependent selection mechanisms that attribute an increasing cost to more prolific (less GoL-like) and more specific rule specification, so that survival mandates complex dynamics close to the game of life. We compare these with rule-independent selection mechanisms that interact with the GoL rules only in so far as birth or survival

## Example: selection = 8

## Example: selection = 9

# Genetically controlled coupling to other GoL dynamics

## Example: selection = 10

## Example: selection = 11

## Example: selection = 12

## Example: selection = 13

# Genetically controlled 3d GoL

Carter Bay proposed investigated possible extension of the Game of Life to 3D, finding that amongst the possible semi-totalistic rules with 26 neighbours there were strong constraints for rules that exhibited the central properties of the game of life (with E=S and F=B in our notation):

*“Definition 1. A rule ElEuFlFu defines a "Game of Life" if and only if both of the following are true.*

*1. A glider must exist and must occur "naturally" if we apply ElEuFlFu repeatedly to primordial soup configurations.*

*2. All primordial soup configurations, when subjected to ElEuFlFu, must exhibit bounded growth.*

*(Here we define primordial soup as any finite mass of arbitrarily dense randomly dispersed living cells.)”*

In particular, 5≤Fl≤9 are hard constraints to ensure 1. and 2. and Carter focused on the range 4 to 7 as most relevant to GoL-likeness for both E and F. He found only two rules 4555 and 5766 to satisfy definition 1, and of these only the rule 5766 supported an extension of many 2D-GoL objects to 3D (by plane duplication). It seems that in 3D, as for 2D, the semi-totalistic rules provide very tight constraints without significant alternatives to the known GoL rules. Bay also discusses possible extensions to rules that distinguish the 26 neighbours into three classes: face-centered (6), edge-centered (12) and corner (8) sites. Given the preferred z-axis of our asymmetric 64xNxN space and the desire to relate 3D rules to 2D rules it makes sense to further distinguish the (xy) in-plane and out-of-plane sites yielding 5 classes with 4,2,4,8,8 members. The number of rules with the total sum in the range 4-7 is then …

## Example: selection = 14

# Discussion

# Some text blocks

## Direct encoding of local CA rule by genes

The most common family of CAs within which GoL-like rule-tables are defined is the semi-totalistic or outer-totalistic automata family, so-called because a cell’s next state depends only on the current state and sum of the neighbour states *s* (totalistic rules depend only on the sum of the current state and all neighbour states). The birth rules for current state “0” and the survival rules for current state “1” define the exceptions to the default rule which is next state “0”.

Genes may specify any look up table (LUT) that maps the sum of neighbors to cell values 0 or 1, and the LUT construction may be generalized if we go beyond the semi-totalistic family, as we shall later on. For the semi-totalistic case, if we exclude the special cases of spontaneous birth (B0 *i.e.* birth for s=0) and lone survival (S0) then there are 8+8=16 distinguished states that may be independently part of an active next state ruleset. Thus there are 216 CA rulesets, and these may be encoded by a binary genome of length 16 with one bit per LUT entry. In this paper, we restrict our attention to genes of maximum length 64, and often use the words gene and genome interchangeably to refer to the full sequence, only rarely using the term gene to refer to a specifier of part of the rule-table. We may also employ multiple bits (ncoding) to encode each LUT entry for an active rule, for example with only one of the possible ncoding gene patterns being active, then genomes of length 16x ncoding are required. A modular variable length encoding could be to encode the values of s (requiring 3 bits) as well as the central state (for survival or birth) for which the next state is live i.e. 4 bits in total per entry. The standard game of life would require 3\*4 = 12 bits to be specified. Longer genomes could contain the same entry repeatedly allowing for mutational error resistance.

Requiring that the birth and survival rules form a single interval of neighbourhood sum values, with lower and upper limits in the sum variable *s,* restricts the possible rule-tables to a family with members specified by four integers SlSuBlBu (lower and upper values of the neighbourhood sum for survival and birth, respectively) and restricts the rule space above to 362=1296 possible rules. Most of these rule-tables either lead to strong proliferation of live states or their extinction, and in order to allow genetic encoding to deliver novel dynamics of interest it turns out to be important to further dissect the rule-tables in the vicinity of the GoL rule 2333.

There are a number of ways that this may be done, taking spatial symmetries into account. We shall list several and investigate a subset of them:

1. Distinguish arbitrary configurations ignoring symmetries: there are 256 binary neighbour patterns in the Moore neighbourhood, and hence 2512 possible rule-tables. We could encode these directly with a 512-bit genome or with a variable length genome by a sequence of 9-bit patterns specifying configurations (including the central site as a 9th bit) that give rise to live states. A maximum of 7 such rules would fit in a 64-bit genome and the standard GoL genome would have to be 9\*(C82+2\*C83)= 9\*(28+56)=756 bits in length. For completeness, we could also consider the continuous interval subset of rules defined by upper and lower bounds for the integer value of the 8-bit neighbor configuration for survival and birth, for which rule-tables can be specified by 2\*(8+8)=32 bits.
2. Distinguish all configurations that are not equivalent by 4-fold rotation and reflection symmetry in 2D. The numbers of these distinguished configurations for s=0…8 are 1,2,6,10,16,10,6,2,1 *i.e.* in total 54, or 52 excluding the s=0 case. The minimum length encoding of a gene specifying this table with one bit per LUT entry is thus 104. Encoding with a variable length genome analogously to above would result in 3+4+1 bit patterns needed to specify a LUT entry, and the standard GoL can be encoded with 8\*(6+2\*10)=208 bits. 64-bit genomes would allow up to 8 such entries to be specified (*e.g.* 4 survival and 4 birth configurations) in a variable length genome. Note that the variable genome length encoding allows the rule-tables of different genomes to be combined simply: any entry that is defined is valid irrespective of multiplicity.
3. Distinguish all configurations that are not an 8-fold rotation of one another. This is both a simpler and somewhat smaller symmetry partition of the 8-neighbour configurations as *s* varies from 0..8 with 1,1,4,7,10,7,4,1,1 a total of 36 distinguished configurations. Configurations can be quickly mapped to their symmetry class, by finding the rotation of 8 bits that yields the minimal numerical value (which we call the canonical rotation) and using these values as identifiers for the different classes (separately for each value of *s*). Restricting attention to the central s-range of 2-6 gives 32 distinguished configurations and 64 LUT entries for survival or birth. This fits neatly into a 64-bit integer genome. The variable length genome encoding still requires 8-bit entries and the standard GoL rule can be specified in 8\*(4+2\*7)=144 bits.
4. Distinguish the four diagonal or corner sites (NW,NE,SE,SW) from the four edge-centered sites (N,E,S,W) in an otherwise semi-totalistic rule-table. The numbers of partitions of *s* into these two classes of sites are for *s*=0…8 1,2,3,4,5,4,3,2,1 = 25 *i.e.* 2\*24=48 bits are required for a direct fixed gene length encoding. The variable length genome encoding above requires 7-bit entries (excluding as before the s=0 entries) and the GoL standard rule is encoded in 7\*(3+2\*4)=77 bits.
5. The semi-totalistic case, distinguishing configurations only by *s*. In addition to the 16-bit gene encoding (excluding *s*=0 entries) there is also the variable gene length encoding involving entries of length 4-bits each. The GoL rule can be specified in 12-bits as described above.

In this work, we find that the semi-totalistic case is too coarse an encoding of CA rules, to allow significant genetic evolution of complex structures beyond the classic game of life. We explore both the fixed length and variable length encodings. Focussing initially on cases (iii) and (iv), rather than the more differentiated (ii), we investigate the more differentiated rules and find that they produce a range of interesting dynamics. When coupled to a genetic population, these differentiated rules rapidly evolve to proliferate unless very strongly constrained.

The key focus in this paper is on sequence- and possibly population-dependent selection mechanisms that attribute an increasing cost to more prolific (less GoL-like) and more specific rule specification, so that survival mandates complex dynamics close to the game of life. We compare these with rule-independent selection mechanisms that interact with the GoL rules only in so far as birth or survival

## Movement vs copying

Whereas the indistinguishability of GoL “1” states means that it is not possible to distinguish movement from death and rebirth, with genetic information attached to the live states, it would be possible. Is there a meaningful assignment of a subset of GoL birth or survival transitions to movement? It would make a difference if mutation were deemed not to occur for transitions involving movement. Also, it might be appropriate to make the choice of an ancestor sensitive to the interpretation of movement *vs* birth: e.g. to minimize the number of births needed to maintain the dynamics. For example, an isolated rod of three live states is a GoL oscillator between vertical and horizontal configurations. In the deterministic most different ancestor canonical assignment of ancestors from three live neighbours, the central gene is copied to two new sites so that (without mutation) the rod becomes genetically homogeneous in one step. This process is clearly a copy mechanism. On the other hand, in the 0-bit canonical assignment of ancestors, the two peripheral genes circulate anti-clockwise and this is more naturally understood as a process of motion and as such should be carried out without mutation.

## Search algorithms for gliders, spaceships and more

https://www.ics.uci.edu/~eppstein/ca/search.html

## 3D GoL Extension

Carter Bay proposed investigated possible extension of the Game of Life to 3D, finding that amongst the possible semi-totalistic rules with 26 neighbours there were strong constraints for rules that exhibited the central properties of the game of life (with E=S and F=B in our notation):

*“Definition 1. A rule ElEuFlFu defines a "Game of Life" if and only if both of the following are true.*

*1. A glider must exist and must occur "naturally" if we apply ElEuFlFu repeatedly to primordial soup configurations.*

*2. All primordial soup configurations, when subjected to ElEuFlFu, must exhibit bounded growth.*

*(Here we define primordial soup as any finite mass of arbitrarily dense randomly dispersed living cells.)”*

In particular, 5≤Fl≤9 are hard constraints to ensure 1. and 2. and Carter focused on the range 4 to 7 as most relevant to GoL-likeness for both E and F. He found only two rules 4555 and 5766 to satisfy definition 1, and of these only the rule 5766 supported an extension of many 2D-GoL objects to 3D (by plane duplication). It seems that in 3D, as for 2D, the semi-totalistic rules provide very tight constraints without significant alternatives to the known GoL rules. Bay also discusses possible extensions to rules that distinguish the 26 neighbours into three classes: face-centered (6), edge-centered (12) and corner (8) sites. Given the preferred z-axis of our asymmetric 64xNxN space and the desire to relate 3D rules to 2D rules it makes sense to further distinguish the (xy) in-plane and out-of-plane sites yielding 5 classes with 4,2,4,8,8 members. The number of rules with the total sum in the range 4-7 is then …