The Evolution of Simplicity: Self-Replicating Systems in Artificial Life

Jason D. Lohn

Caelum Research Corporation

NASA Ames Research Center

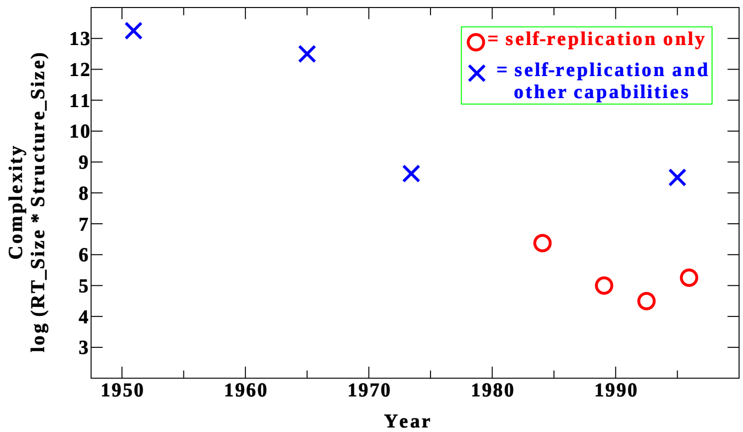
slide 1

Two Areas in Artificial Life

Possible steps to natural life	**Protocell ⟶ Metabolism ⟶**	**Self-replication**
Artificial life	**Synthesis of artificial prebiotic chemistries** (2)	**Synthesis of artificial self-replicating systems** (1)

slide 2

Self-Replicating Systems in Cellular Automata



slide 3

Outline

Part I - Overview and Background

- Motivations
- Cellular space models
- Self-replicating systems
- Artificial life
- Research summary

Part II - Case Studies

- von Neumann
- Codd
- Langton
- Byl
- Reggia
- Tempesti
- Arbib
- Holland
- Sipper
- Lohn (evolved)
- Perrier
- Chou (emergent)

Demonstration of self-replicating structures

Bibliography and On-line References

slide 4

Objectives

To gain understanding of:

- a key research area in Artificial Life
- how self-replicating systems have been constructed
- the underlying cellular space models used
- evolving self-replicating systems
- potential research directions in the field

slide 5

Motivations

Motivation for studying models of self-replication in cellular spaces:

- to further artificial life research
- nanotechnology: atomic–scale manufacturing [Drexler89]

  If assemblers are to process large quantities of material atom-by-atom, many will be needed; this makes pursuit of self-replicating systems a natural goal.
- programming massively parallel computers: evolutionary bred self-replicating programs [Ray92]
- understanding complex system dynamics and emergent properties [Reggia93]
- biomolecular mechanisms of reproduction and origins of life [Hong92]

slide 6

Cellular Space Models

A *cellular space* is a tessellation of cells containing automata that interact with neighboring cells.

Key properties of cellular space models:

- strictly local interactions producing emergent behavior
- rule based, usually deterministic
- highly parallel
- simple automata (in general)
- space iterates in discrete time

Categories of cellular space models:

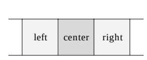
- cellular automata (CA) – most widely studied
- variations of cellular automata
- models with complex automata

slide 7

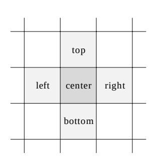
Cellular Automata

Cellular automata model of von Neumann:

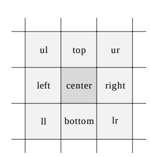
- tessellation of cells iterating in discrete time, each cell containing a finite state machine
- automata behavior governed by rule table, δ
- neighborhood pattern determines an automaton's input:



3-cell, 1-D



5-cell
von Neumann
neighborhood



9-cell
Moore
neighborhood

Well-known CA rule: Conway's Game of Life

slide 8

CA Terminology

- state of the space at a given time is called a configuration
- t=0 configuration is called the initial configuration
- k states, n neighbor, neighborhood of radius r
- one state specially designated as quiescent (blank, inactive); a quiescent cell remains quiescent if all cells in its neighborhood are quiescent
- behavior governed by rule table (rule, transition function) denoted δ
- strong rotational symmetry: all cell states are unoriented – each neighbor to a cell has no special absolute nor relative position
- weak rotational symmetry: at least some of the cell states are directionally oriented – each cell designates specific neighbors as being its top, right, bottom, and left neighbors

slide 9

CA Example

Cellular automata parity function: a cell's next state is 1 if there are an odd number of 1s in the 5-cell neighborhood

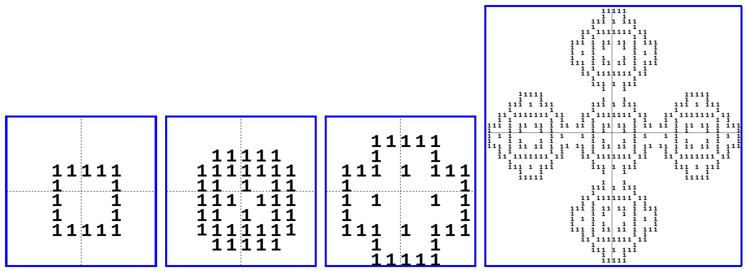
Rule table:

CTRBL	C′	CTRBL	C′	CTRBL	C′	CTRBL	C′
00000	0	01000	1	10000	1	11000	0
00001	1	01001	0	10001	0	11001	1
00010	1	01010	0	10010	0	11010	1
00011	0	01011	1	10011	1	11011	0
00100	1	01100	0	10100	0	11100	1
00101	0	01101	1	10101	1	11101	0
00110	0	01110	1	10110	1	11110	0
00111	1	01111	0	10111	0	11111	1

slide 10

CA Example

Configurations at t=0, t=1, t=2, and t=22 for the parity function:



slide 11

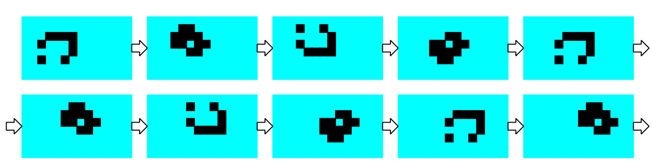
A Famous CA: Conway's Game of Life

John H. Conway's Game of Life (Scientific American, Oct. 1970)

CA: 2-state, 9-neighbor, strong rotational symmetry

- Birth: quiescent cells become active when they have exactly three neighbors
- Survival: active cells that have two or three active neighbors remain active
- Death: active cells with more than 3 or less than 2 active neighbors become quiescent

"Fish" structure:

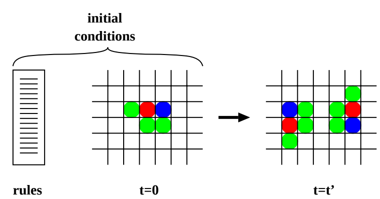


slide 12

Self-replicating structures

Self-replicating structures:

- represented as a configuration of contiguous active (non-quiescent) cells
- initial conditions comprised of a set of rules and an initial configuration
- at some time t', copy of the original structure appears isolated, and possibly rotated
- illustration of basic concept:



slide 13

Self-replicating structures

Some systems allow for more than self-replicating behavior:

- universal computation – universal Turing machine
- universal constructor – a machine that can construct whatever machine's description is given as input (e.g. itself)
- biochemical simulation – automata model polymers in biochemical reactions

Requiring universal computation and construction avoids trivial self-replicating structures, however, it greatly increases the complexity of the automaton (Langton 1984)

In many models, an instruction sequence (tape, circulating string, DNA) is interpreted as instructions to be carried out, and as passive data to simply be copied (translation and transcription)

slide 14

Artificial Life

Artificial Life (ALife) studies life-like behaviors (such as self-replication) from a computational perspective and was largely born out of studies based on self-replicating systems in cellular automata.

From Langton 1988:

> Artificial Life is the study of man-made systems that exhibit behaviors characteristic of natural living systems.
>
> By extending the empirical foundation upon which biology is based beyond the carbon-chain life that has evolved on Earth, Artificial Life can contribute to theoretical biology by locating life-as-we-know-it within the larger picture of life-as-it-could-be.

slide 15

Artificial Life and Self-Replicating Systems

- biology is top-down, analytic; ALife is bottom-up, synthetic
- defining "life" in a precise manner is difficult due to the presence of organisms that are sterile, and organisms that lack a metabolism, such as viruses
- however, self-replication is generally seen as a fundamental property of life
- cellular spaces are a computational paradigm that match ALife nicely: behavior is determined in a bottom-up fashion through strictly local interactions

slide 16

Chronological Summary

Year	*Model Type*	*Rot. Sym-metry*	*States per Cell*	*Neigh-borhood size(s)*	*Struc-ture size(s)*	*Capa-bilit-ies*	*PI*
1951	CA	weak	29	5	$> 10^4$	o	von Neu.
1965	CA	strong	8	5	10^8	o	Codd
1966	CT-Mach.	weak	$\approx 10^{100}$	5	$\approx 10^2$	o	Arbib
1973	CA	strong	8	5	$> 10^4$	s	Vitanyi
1976	α-Univ.	strong	5	var.	(60)	s	Holland
1984	CA	strong	8	5	86	s	Langton
1989	CA	strong	6	5	12	s	Byl
1993	CA	both	6,8	5,9	5–48	s	Reggia
1995	CA,EA	weak	9,13	5	2,3,4	s	Lohn
1995	CA	strong	6	9	52	o	Tempesti
1995	non-uni. CA	strong	2	9	5	s	Sipper
1996	CA/W-Mach.	strong	63	5	127	o	Perrier

s=self-replication, o=other capabilities in addition to self-replication.
All are 2-D models except for the α-Universe model.

slide 17

Part II – Case Studies

- von Neumann
- Codd
- Langton
- Byl
- Reggia
- Tempesti
- Arbib
- Holland
- Sipper
- Lohn
- Perrier

Demonstration video shown for some of the systems

slide 18

John von Neumann

slide 19

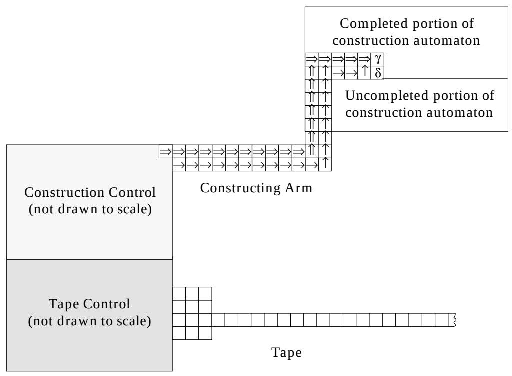
von Neumann's SR Automaton

- John von Neumann was the first to study self-replicating systems in the late 1940s
- motivated to understand: growth of complexity, minimum requirements for a self-replicating machine, and the logical representation of the natural self-replication problem
- developed cellular automata (CA) model from a suggestion by Ulam
- developed the logical design for a self-replicating machine capable of universal computation and construction
- proved that self-replication was achievable by machines
- machine embedded in a 29-state, 5-neighbor, weakly rotation symmetric CA, consisting of many millions of cells
- extreme complexity agreed with his intuition, however much of the complexity is due to the universal constructor/computation capabilities

slide 20

von Neumann's SR Automaton

Overview of von Neumann's self-replicating automaton (Burks, 1970):



slide 21

von Neumann's SR Automaton

Overall functions:

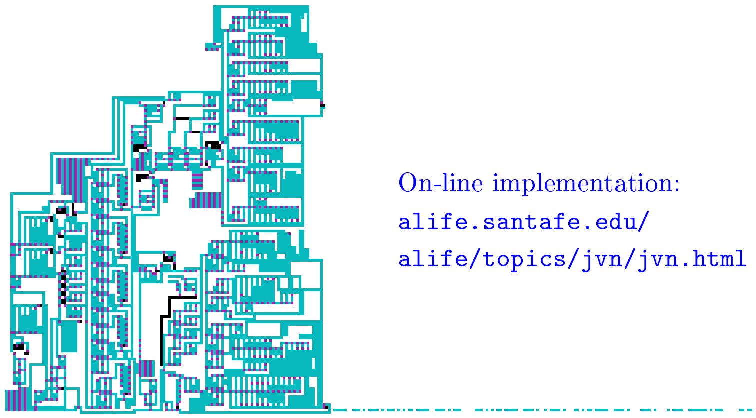
- reading and interpreting the input tape
- constructing new cells in the quiescent area
- "rewinding" the tape, then copying it
- attaching tape copy to newly constructed portion
- signaling the newly constructed portion that construction had completed
- retracting the construction arm

Four main areas of the machine:

- reading loop area: reads input tape containing description of machine to build
- memory area: collects data from the reading loop and outputs excitations to the writing loop area

slide 22

- writing loop area: directs action of the writing arm to construct the offspring machine in an area of quiescent cells
- state control area: contains a counter and outputs excitations to control various parts



On-line implementation:
`alife.santafe.edu/`
`alife/topics/jvn/jvn.html`

slide 23

Codd's Model

Codd (1968) introduced a simpler universal constructor:

- his machine was embedded in an 8-state, 5-neighbor, 2-D strongly rotation symmetric CA, consisting of 100,000,000 cells
- similar in behavior to von Neumann's model, but with reduced complexity
- design influenced by neurophysiology of animals
- later simplified to approximately 95,000 cells (Devore 1992)

Vitányi's Model

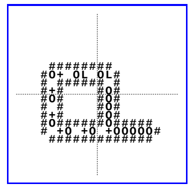
- sexually-reproducing CAs
- 8-state, 5-neighbor, tens of thousands of cells
- produced two structures capable of sexual reproduction (Vitányi 1973).

slide 24

Langton's SR Loop

Langton (1984) relaxed requirement of computation universality; using concepts from Codd's work, he derived an 8-state, 86-cell sheathed loop that requires 108 replication rules, orders of magnitude simpler than previous models

initial configuration:

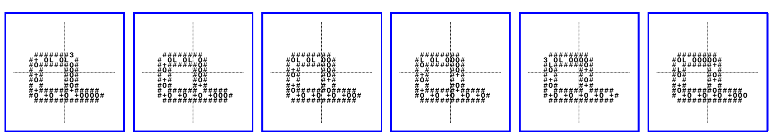


- instruction sequence `++++++LL` is embedded in the core of `O` states

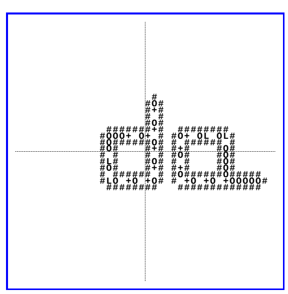
slide 25

Langton's SR Loop

Time-steps 1 through 6:



At time-step 151 the first replicant is formed:



slide 26

Rule Table for SR Loop

```
.....  -> .   0...0 -> 0   #...0 -> #   ###60 -> #   5..#.  -> #
....0 -> #   0...6 -> 0   #...L -> #   ###+0 -> #   5..#0 -> 5
....6 -> 3   0..0# -> 0   #...+ -> 0   ###+# -> #   5..## -> 5
...0# -> #   0..#0 -> 0   #..#L -> #   ###++ -> #   5..#+ -> #
...03 -> #   0..#L -> L   #..L# -> 3   #3..# -> #   5..5# -> .
...#.  -> .   0..#+ -> +   #..5+ -> 5   #3#.# -> 0   5.#3.  -> #
...#3 -> .   0..+.  -> +   #.0.# -> #   #L0## -> #   5#.#.  -> #
...#+ -> #   0.0#+ -> +   #.00# -> #   #L##.  -> #   5#.## -> .
...3.  -> .   0.#0# -> 0   #.0#.  -> #   #L##0 -> #   5#.#+ -> #
...3# -> .   0.##6 -> 3   #.0## -> #   #5..0 -> #   5#0#.  -> #
...5.  -> .   0.#3# -> +   #.0+# -> #   #5..# -> .   5#0#0 -> #
...5# -> 5   0.50. -> 0   #.#.# -> #   #5.#.  -> #   5#0#+ -> #
...6# -> #   0.5L# -> +   #.#0. -> #   #5.#0 -> #   5###0 -> .
...+.  -> 0   00.#+ -> .   #.#0# -> #   #5.#L -> #   5##L. -> L
..#.# -> .   000#0 -> 0   #.##+ -> #   #5.#5 -> #   55.#0 -> #
..#.5 -> .   000#5 -> 0   #.#L# -> #   #50#.  -> #   6...0 -> 0
..##.  -> .   000#6 -> 0   #.#+.  -> #   #5#.# -> .   6#...  -> 0
..3.# -> .   000#+ -> +   #.#+# -> #   #5##.  -> #   6#0#.  -> .
..5## -> #   00#0# -> 0   #.3.# -> #   #6..# -> #   6#0#0 -> 5
..+#.  -> #   00#3# -> 0   #.30# -> #   #6#.# -> #   6##0# -> 5
```

slide 27

Rule Table (con't)

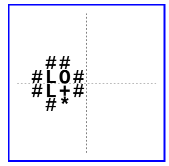
```
.0..# -> #   00#L. -> L   #.3L# -> #   #6##+ -> #   630#0 -> 0
.0.#.  -> #   00#+# -> +   #.50. -> +   #+.#.  -> 3   +..+.  -> +
.00#.  -> .   0#.#.  -> 6   #.5#.  -> #   #+##.  -> #   +00#.  -> .
.06#5 -> 0   0#.#6 -> 6   #.55# -> 0   3...0 -> 3   +#.0# -> .
.0+## -> 0   0#005 -> #   #.+0# -> #   3...# -> #   +#.#0 -> .
.#.#0 -> 5   0#0#L -> L   #.+#.  -> #   3...L -> 0   +###.  -> 0
.#0#L -> 0   0##0. -> 0   #.+L# -> #   3...+ -> 6   +#3#.  -> 0
.#0#5 -> 5   0##00 -> 0   #0.## -> #   3..0# -> 3   +#5.0 -> .
.#0#+ -> 0   0###0 -> 0   #0#60 -> 0   3..L# -> 0   +#5.# -> 0
.###.  -> .   0###L -> L   #0+## -> #   3..6# -> #   +#5#.  -> 5
.#3#.  -> #   0###+ -> +   ##...  -> #   30.#.  -> 0   ++#.# -> .
.#3#0 -> 0   0##+0 -> +   ##..3 -> 6   30##.  -> .
.#50+ -> 5   0#3#L -> L   ##.3# -> 6   350.# -> 0
.#5#+ -> 0   0#3#+ -> +   ##.5+ -> 5   L00#.  -> .
.#6#0 -> 0   0#L.# -> L   ##.6# -> #   L#.#0 -> .
.3#0L -> 0   0#L#.  -> L   ##.6+ -> #   L##.0 -> .
.LL#0 -> 0   0#L#5 -> 5   ##.++ -> #   L###.  -> 0
.6..# -> #   0#50# -> 0   ##0.3 -> 6   L#3#.  -> 6
.+#0L -> 0   0#5#+ -> 5   ##0.5 -> #   L3##.  -> 0
.+50# -> 0   0#5L# -> +   ##05# -> #   L5#.# -> .
```

slide 28

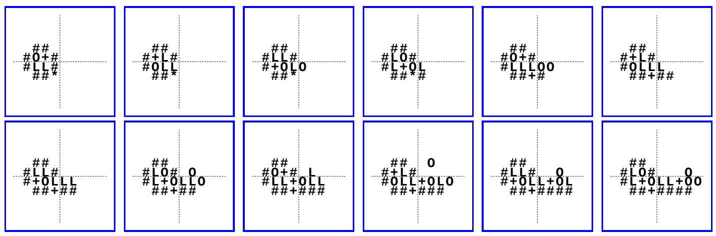
Byl's SR Structure

Byl (1989) made further refinements and derived a 6-state, 12-cell self-replicating structure that requires 57 replication rules and has a single sheath

initial configuration:



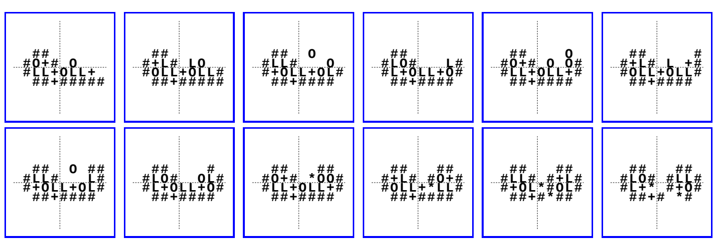
Time-steps 1 through 12:



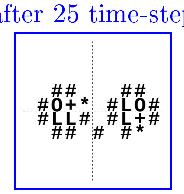
slide 29

Byl's SR Structure

Time-steps 13 through 24:



First replicant produced after 25 time-steps:



slide 30

Reggia's SR Loops

Further simplified self-replicating loops by deriving unsheathed loops (Reggia 1993)

Verified that both strong and weak rotational symmetries can yield simple self-replicating structures

Smallest structures found:

- 6-state, 5-neighbor, 5-cell unsheathed loop under strong rotational symmetry
- 6-state, 5-neighbor, 6-cell unsheathed loop under strong rotational symmetry
- 8-state, 5-neighbor, 6-cell unsheathed loop under strong rotational symmetry
- 8-state, 5-neighbor, 6-cell unsheathed loop under weak rotational symmetry

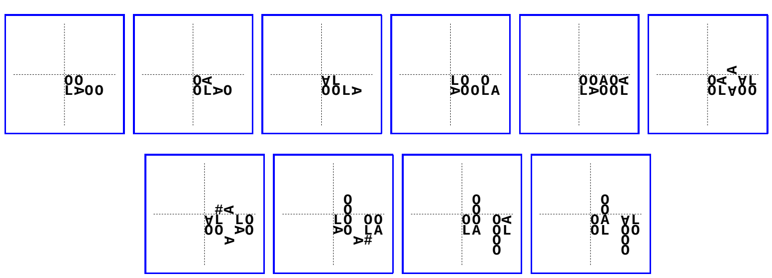
slide 31

Reggia's SR Loops

Structure `UL06W8V` – one of the smallest known CA self-replicating structures

Number of replication rules = 58

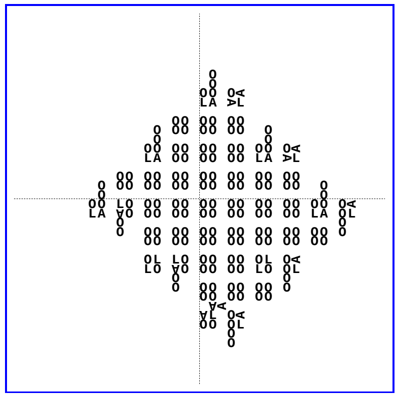
Time-steps 0 through 9:



slide 32

Reggia's SR Loops

Structure `UL06W8V` at time-step 84:



slide 33

Tempesti's SR Loop

Tempesti (1995) reports a 6-state, 9-neighbor, 52-cell self-replicating structure, augmented with additional construction and computational capabilities

Similar to Langton's self-replicating loop, except

- ability to execute programs in offspring structures
- use of a single sheath, which is constructed prior to the signal being sent out
- parent loops remain active (and are capable of program execution)
- larger Moore neighborhood used
- construction arm extends in four directions simultaneously

slide 34

Tempesti's SR Loop

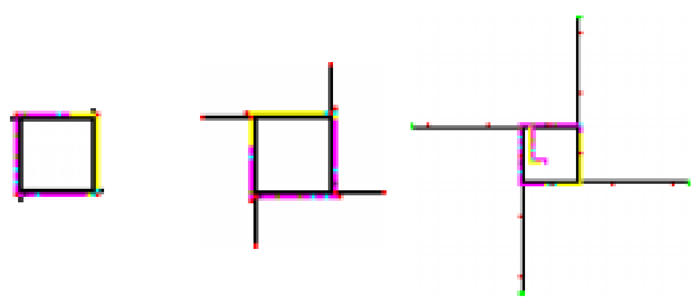
Time steps 0 and 121 state d represents the data state

```
                                          1
                                          1
      1                      1            1
 dddd2ddd            1dd2ddddd      dddd2ddd
1d111111d             d111111d     1d111111d
 d1     1d            d1    12      d1    12
 21     1d            d1    1d      d1    1d
 d1     12            d1    1d      d1    1d
 d1     1d            21    1d      21    1d
 d111111d1            d111111d      d111111d111
 ddd2dddd             ddddd2dd1     ddddd3dd
  1                   1             1
  time = 0                time = 121
```

slide 35

Tempesti's SR Loop

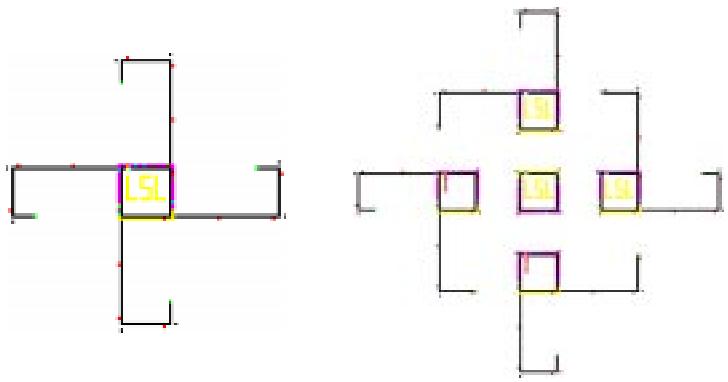
Configurations from time-steps 0, 20, 67 (not to scale):



slide 36

Tempesti's SR Loop

Configurations from time-steps 130 and 396 (not to scale):



Arbib's CT-machine

Motivation for Arbib's model:

- the large degree of complexity of von Neumann's and Codd's self-replicating automata could be greatly reduced if the fundamental components were more complex
- automata are analogous to biological cells, as opposed to molecules

Using a very complex automaton, having $\approx 10^{100}$ states and capable of both universal computation and construction, his model achieves a simpler design than that of von Neumann and Codd.

Cellular space model (Arbib, 1966):

- automata are embedded in a 2-D cellular space model called Constructing Turing machines, or CT-machines (Thatcher 1970)

Arbib's CT-machine

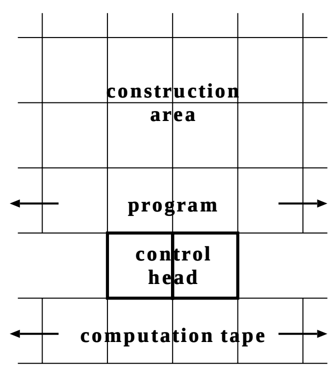
Cellular space model:

- each cell in this space contains a finite-state automata that execute short 22-instruction programs
- instructions consist of actions such as weld and move, and internal control constructs such as if/then and goto
- self-replication occurs when individual CT-machines copy their instructions into empty cells
- composite structures consisting of multiple CT-machines are able to move as one unit since individual automata can be welded to each other

slide 39

Arbib's CT-machine

Overview of embedded automata in the CT-Machine model:

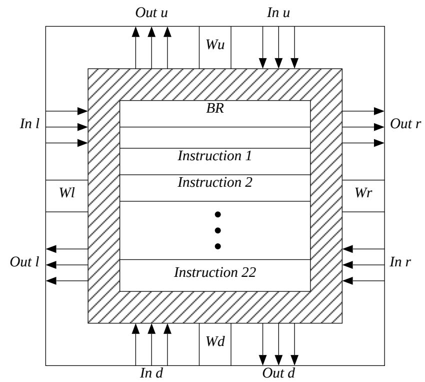


Each of the four components is constructed out of identical automata, each programmed specifically for the appropriate function

slide 40

Arbib's CT-machine

Automaton in Arbib's CT-Machine model:



`W` denotes weld positions, `BR` denotes bit register, module is programmed using instructions such as weld, emit, move, goto

slide 41

Arbib's CT-machine

Machine operation:

- since cells can be "welded", a tape can be formed
- control head can read and write the computation tape in the same manner as a Turing machine
- construction area is initially quiescent
- program cells can only write into the construction area
- such write operations are equivalent to the placing of new components

Notes on the CT-machine cellular space:

- automata programs are demonstrated which accomplish certain functions involving control and construction
- self-replication was not implemented, rather, demonstrated as a mathematical proof (as was done by von Neumann)

slide 42

Holland's α-Universe

Holland (1976) explores automatic discovery of self-replicating automata by focusing on spontaneous emergence of such structures

- developed a theoretical framework and provided existence proofs for emergence of self-replicating structures
- defines a set of model "universes" containing counterparts to chemical and kinetic mechanisms such as bonding and movement
- aim was to loosely model natural chemical processes (diffusion, activation) acting on structures composed of elements (nucleotides, amino acids)
- wanted to show that even with random agitations, the tendency of such a system would not be sustained randomness, but rather, life "in the sense of self-replicating systems undergoing heritable adaptations."

slide 43

Holland's α-Universe

The α-Universe cellular space model represents cell states as elements that are logical abstractions of physical entities (e.g. atoms) and obey the conservation of mass.

An example of a few cells from an α-Universe:

... | - | 0 | : | 1 | 0 | 0 | 1 | - | N_0 | N_1 | - | - | 0 | : | ...

Element	Codon
0	N_0N_0
1	N_0N_1
:	N_1N_0

elements are the fundamental units, codons encode elements

slide 44

Holland's α-Universe

Operation of the model:

- interactions among the elements are strictly local as in CA, but some are localized to aggregate structures (strings of bonded elements)
- elements behave as automata during the first of three "phases" of each discrete time-step
- during the second and third phases, they are acted upon by the four operators: bonding, movement, copy, and decode
- example: "copy" operator is activated if the sequence `-0:`$e_1 e_2 \cdots e_l$`-` forms (e_i being one of the three elements), and it would cause elements to be reshuffled so that a codon-encoded copy of the string $e_1 e_2 \cdots e_l$ would be assembled

slide 45

Holland's α-Universe

Holland parameterizes important aspects of the α-Universes and then uses these to derive formulas that predict the expected time required for emergence of a self-replicating system

Substituting reasonable values into his derivations, a waiting time of 1.4×10^{43} time-steps is computed (no emergence)

Relaxing the requirement from *fully* self-replicating to *partially* self-replicating, a waiting period of 4.4×10^8 time-steps (4.4×10^8 seconds is about 14 years) is obtained

Since this is a reasonable amount, it lends credence to spontaneous emergence of self-replicating structures in general, given that Holland's model and derivations are accurate

In (McMullin 1992), an empirical investigation claims that some of the conjectures were flawed

slide 46

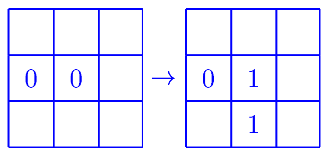
Sipper's SR Loop

Sipper (1995) describes a self-replicating loop motivated by Langton's work

The cellular space model is a modified cellular automata whereby:

- like a CA, the space iterates in discrete time with cells updated in a local, synchronous manner
- unlike a CA, a given cell can change a neighboring cell's state
- also, a cell can copy its rule into a neighboring cell (non-uniform CA)

An example transition rule:



slide 47

Sipper's SR Loop

Using 10 transition rules, a 6-cell structure in a 2-state, 9-neighbor model, exhibits self-replication

```
                                                 1
11     11      11      11       11     11       11
111    1110    11100   11101    11101  11011    11 11
                           1       11     11       11
                                          1        1
                                                   0
t=0     t=1     t=2     t=3      t=4    t=5       t=6
```

slide 48

Perrier's SR Loop

Perrier (1996) reports a 63-state, 5-neighbor, 127-cell self-replicating structure, exhibiting universal computation (following on Tempesti, 1995)

- universal computation achieved by using Turing machine model called W-Machine, programmed using a small instruction set
- complexity is reduced by eliminating requirement of construction universality
- loop structure self-replicates in the same manner as Langton's loop
- program and data tapes are copied using transmitted signals
- after a daughter structure is produced, it can execute a W-Machine program

slide 49

Perrier's SR Loop

The structure consists of three parts: loop, program, and data:

- state `D` represents the data state
- state `P` represents the program state
- state `A` represents the position of the program

```
 ........
.70170170.
.1......1.
.1.    .7.
.1.    .0.
.1.    .1.
.1.    .7.
.0......0....
.410410710711.
.A...........
.P.    .D.
.P.    .D.
.P.    .D.
.P.    .D.
.P.     .
.P.
.P.
 .
```

slide 50

Emergent Self-Replicating Structures

Previous self-replicating structures have always been initialized with a pre-defined structure.

Work by Chou & Reggia (to appear in Physica D) asks:

> Does there exist a CA transition rule that can promote the emergence of self-replicating structures from a *randomly* initialized CA space?

The answer is Yes! The CA rule tables that were found:

- support replication of different-sized structures
- show growth of small structures into larger ones
- allow interactions between structures
- are robust: independent of space size and initial component density

slide 51

Emergent self-replicating loops: t=500 2×2, 3×3; t=1500 4×4; t=3000 8×8; t=5000 10×10. At t =7500 large loops have been replaced by smaller ones.

slide 52

Self-Replicating Loops: Problem Solving and Artificial Selection

Further work on models that incorporate problem-solving capabilities into self-replicating loops.

Previous models incorporated a fixed "program" that is copied unchanged to replicants.

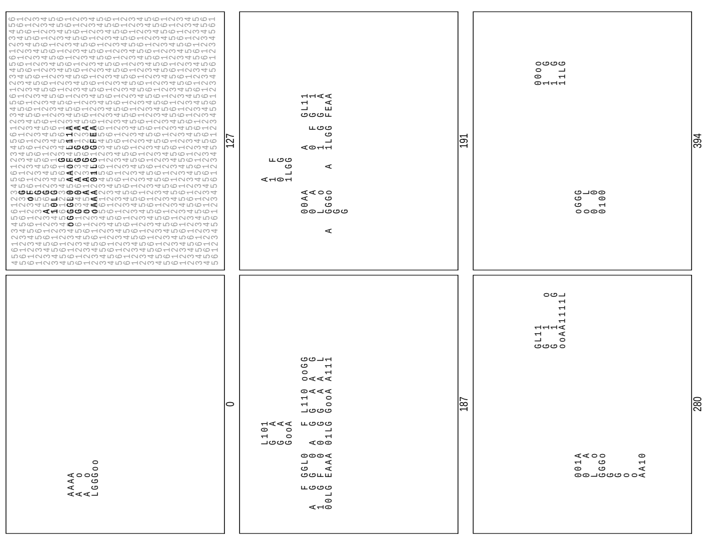
Work by Chou & Reggia (to appear in Physica D) demonstrates solutions to the SAT (satisfiability) problem in which:

- replicants receive partial solutions that are modified during replication
- artificial selection: promising solutions proliferate, failed solutions are lost

Environment selects satisfied clauses: CA space is filled with "monitor" cells which destroy unsatisfied loop fragments.

slide 53

$$(\neg x_1 \vee x_3)\wedge(x_1 \vee \neg x_2)\wedge(x_2 \vee \neg x_3)\wedge(x_4 \vee x_4)\wedge(\neg x_4 \vee \neg x_5)\wedge(x_5 \vee \neg x_6)$$



slide 54

Evolving Self-Replicating Structures

Past models of self-replication in cellular spaces were manually designed:

- difficult to develop
- time-consuming process
- subject to biases of the implementor

Lohn (1995, 1996) explores automatic discovery of self-replicating structures.
Major impediments:

- extremely large and unknown search spaces: $k^{(k^n)}$ for a k state, n neighbor 2-D cellular automata (typically $2 < k < 18$, $n = 4$)
- biases to guide the self-replication process

slide 55

Evolving Self-Replicating Structures

Genetic algorithms (GAs) were used to search rule table spaces for those promoting self-replicating behavior:

- other search techniques did not yield self-replicating structures during initial studies
- two cellular space models were used: cellular automata (CA), and effector automata (EA)
- the EA model:
  - rules specify actions (eg. movement, automaton division)
  - retains desirable cellular space properties (local interactions, emergent behavior, highly parallel)
  - reduced search space sizes
  - more computationally feasible

slide 56

EA Model

- cellular space: 2-D, isotropic, von Neumann neighborhood, discrete time
- automata: directionally oriented, types $\in$ {A, B, C, ...}
- rules: condition action format:
  CTRBL $\rightarrow$ *action(s)*

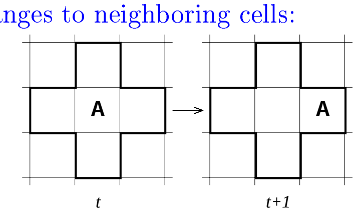
action	*description*
MOVE <dir> <rot>	move one cell in the specified direction and rotate the specified number of degrees
DIVIDE <dir> <rot> <dir> <rot>	divide into two daughter automata according to the specified directions and rotations
DESTRUCT	cease to exist
NULL	no action

- DIVIDE action is at the cell-level, vs. structure-level
- actions have biological counterparts: cell movement, cell division, programmed cell death, inactivity

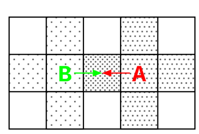
slide 57

EA Model

- EA model derives its name from the fact that each automaton can *effect* changes to neighboring cells:



- because actions can modify neighboring cells, collisions are possible:

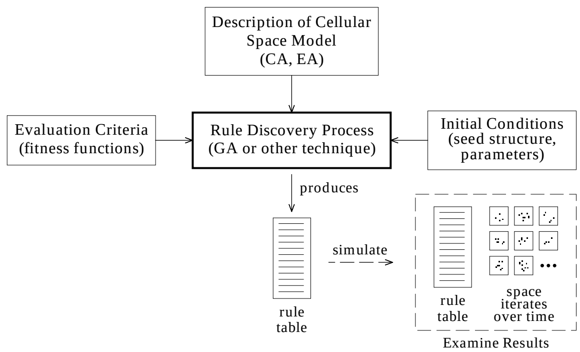


- collision policy used: mutual annihilation

slide 58

Rule Discovery System

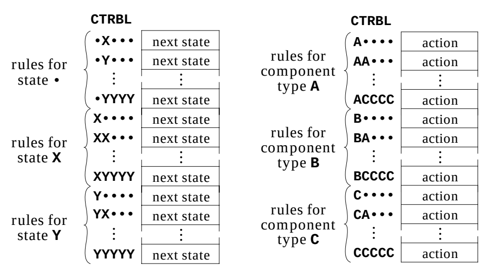
- goal of rule discovery process is to discover rule tables that promote self-replicating behaviors
- overview of rule discovery system:



slide 59

GA for SRS Rule Discovery

Rule table encodings:



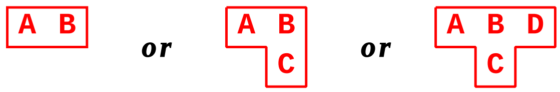
(a) CA (b) EA

Examples of chromosome representation in the GA. Shown are weakly rotation symmetric models comprised of three components

slide 60

GA for SRS Rule Discovery

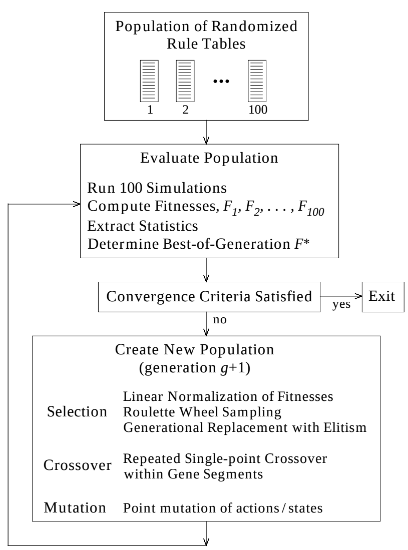
- initial configurations (t=0) are fixed:



- fitness judgment based on 10-15 time-steps of iteration
- trivial SR (simultaneous splitting) not acceptable, guarded against
- replicants must completely separate from parent structures

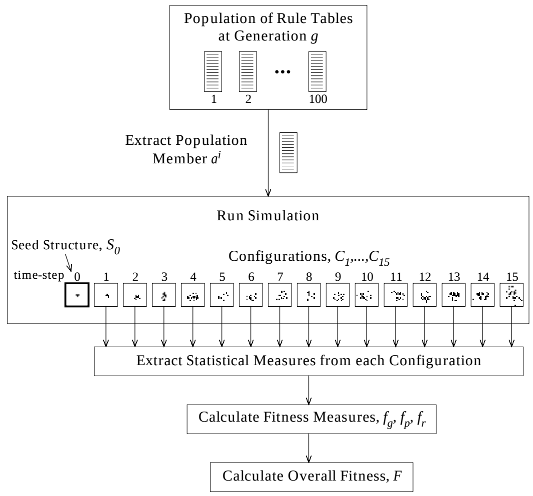
slide 61

GA Overview



slide 62

Evaluation Phase of GA



slide 63

Fitness Functions

- fitness functions use statistics collected over time to evaluate degree of self-replication
- difficulties:
  - partial credit (random initial chromo.)
  - choice of statistics to collect
  - cannot impose specific replication process
- one solution: fitness function based on:
  - growth (production of components)
  - relative positioning (vs. seed)
  - $F = w_1 f_{gm} + w_2 f_{rpm}$, where $w_1 + w_2 = 1, w_1 \geq 0, w_2 \geq 0$
- growth measure: reward monotonic increase of components

$$f_{gm} = \frac{1}{2TN} \sum_{a \in A} \sum_{t=1}^{T} s_a(t)$$

slide 64

Fitness Functions

where

$$s_a(t) = \begin{cases} 2 & \text{increase in component } a \\ 1 & \text{no change} \\ 0 & \text{decrease} \end{cases}$$

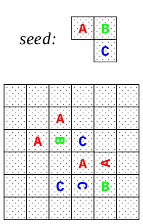
- relative position measure: compare adjacencies of individual automata over time to that of the seed structure
- $\text{adj}(t, a)$ = adjacency score function: measures to what degree each a-type automata have the same neighborhood as in the the seed at time t

$$\text{adj}(t, a) = \frac{\#\text{ neighbors of automata } a \text{ correctly positioned}}{\#\text{ possible matches}}$$

slide 65

Fitness Functions

- f_{rpm} is obtained by averaging $\text{adj}(t, a)$ over time and automata-types
- Example of adjacency calculation:



t = 5

$$\text{adj}(5,\text{ A}) = \frac{1}{4 \cdot 1} = 0.25$$
$$\text{adj}(5,\text{ B}) = \frac{3}{2 \cdot 2} = 0.75$$
$$\text{adj}(5,\text{ C}) = \frac{1}{3 \cdot 1} = 0.33$$

slide 66

Partial Results

- 6 self-replicating structures found in the course of 75 GA runs of 2000 generations

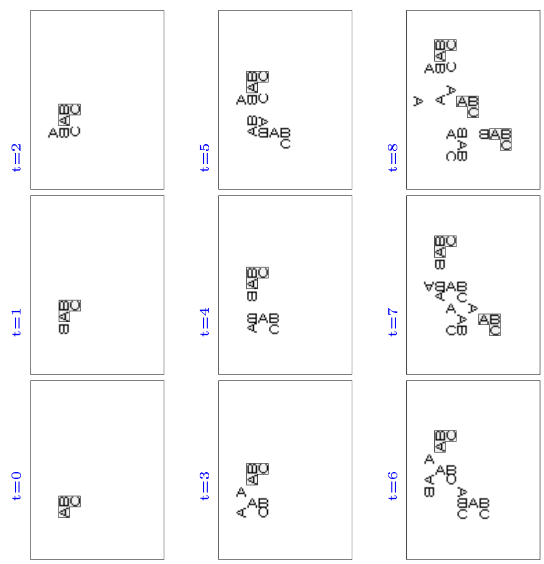
structure	*rules**	f_{gm}	f_{rpm}	F
SRS2a	8	0.944	0.885	0.914
SRS2b	21	1.000	0.612	0.806
SRS2c	12	0.861	0.761	0.811
SRS3a	13	0.741	0.810	0.796
SRS4a	15	0.815	0.887	0.872
SRS4b	8	0.926	0.869	0.881

*replication rules

- small rule counts consistent with those generated manually
- structures move and generate debris
- SRS4a and SRS4b derive from 3-component seed

slide 67

Structure SRS3a

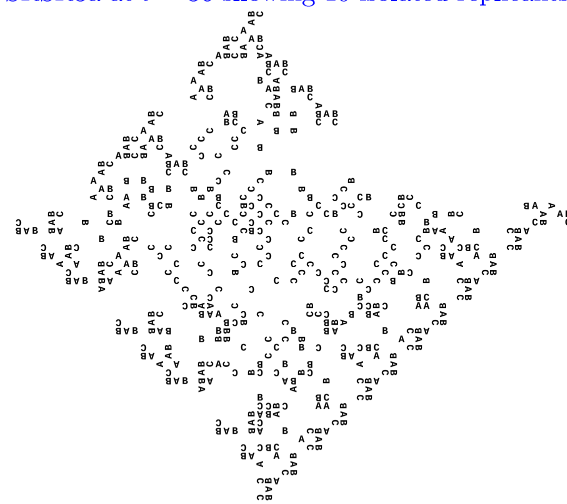



```
A.... -> DESTRUCT
A.B.. -> DIVIDE RIGHT 0 BOTTOM 270
A...B -> MOVE RIGHT 270
A.B.B -> DIVIDE TOP 90 RIGHT 0

CB... -> DIVIDE LEFT 0 RIGHT 0
C.B.. -> MOVE BOTTOM 90

B...A -> DESTRUCT
BA..A -> MOVE BOTTOM 90
BAA.A -> MOVE LEFT 0
B..AA -> DIVIDE LEFT 270 TOP 180
BC.AA -> DIVIDE TOP 270 LEFT 180
B.AAA -> MOVE BOTTOM 90
B..CA -> DIVIDE LEFT 180 RIGHT 0
```

slide 68

Structure SRS3a (cont)

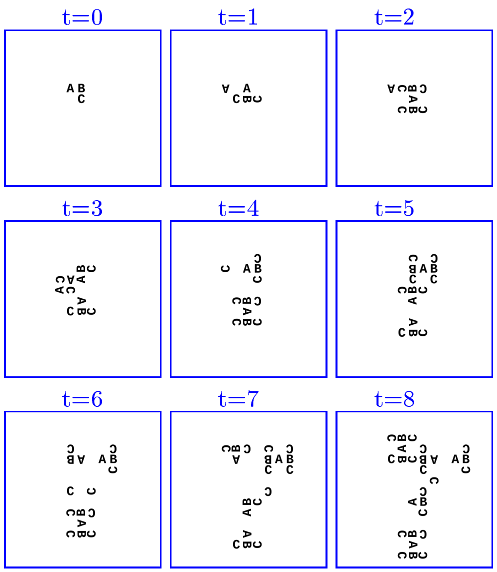
SRSR3a at $t = 36$ showing 10 isolated replicants



slide 69

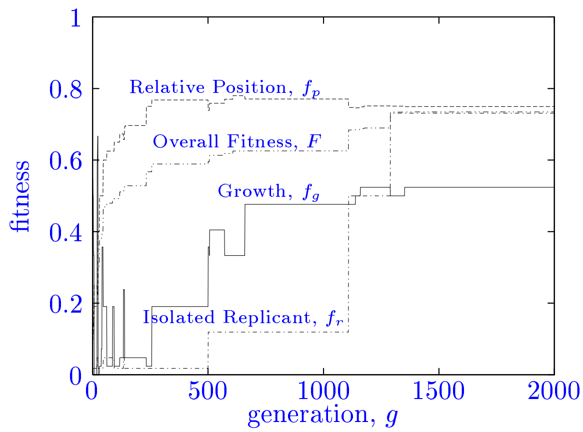
Structure SRS4a

Development of SRSR4a:



slide 70

GA Performance



slide 71

Summary and Conclusion

Summary of points that were covered:

- descriptions of the main models of self-replication in cellular spaces, the study of which began in the late 1940s with von Neumann
- the progressive simplification of self-replicating structures in cellular automata: first by relaxing the requirements of construction and computation universality, then by reducing structure size
- how studies of self-replicating systems fit into artificial life research
- how self-replicating structures have been constructed in cellular space models other than cellular automata
- automatic discovery of self-replicating structures using genetic algorithms

See attached bibliography and WWW references for more information

slide 72

Summary and Conclusion

Some directions for further research in this area:

- investigation of minimal structure size in cellular automata
- effect of varied neighborhoods (size and shape) and varied seed structures
- investigation of other cellular space models, such as stochastic automata
- other techniques for automatic discovery of self-replicating structures
- co-evolving the seed structure and the rule table simultaneously
- biochemical simulation: a few promising studies have appeared in which modified cellular automata models are used to mimic biochemical interactions and simulate template-directed oligonucleotide replication

slide 73

Bibliography

M. A. Arbib, "Simple Self-Reproducing Universal Automata," *Information and Control*, Vol. 9, pp. 177–189, 1966.

M. A. Arbib, "Automata Theory and Development: Part I," *Journal of Theoretical Biology*, Vol. 14, pp. 131–156, 1967.

M. A. Arbib, *Theories of Abstract Automata,* Prentice-Hall, Englewood Cliffs, NJ, 1969.

M. A. Arbib, "Self-Reproducing Automata: Some Implications for Theoretical Biology." In *Towards a Theoretical Biology*, Vol. 2, C. H. Waddington (ed), University of Edinburgh Press, Edinburgh, pp. 204–226, 1969.

R. A. Brooks and P. Maes (eds), *Artificial Life IV*, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, MIT Press, 1994.

A. Burks, *Essays on Cellular Automata*, Univ. of Illinois Press, 1970.

J. Byl, "Self-Reproduction in Small Cellular Automata," *Physica D*, **34**, North-Holland, pp. 295–299, 1989.

E. F. Codd, *Cellular Automata*, Academic Press, 1968.

slide 74

Bibliography

J. Devore, R. Hightower, "The Devore Variation of the Codd Self-Replicating Computer," Third Workshop on Artificial Life, Santa Fe, New Mexico, draft November 30, 1992.

K. E. Drexler, "Biological and Nanomechanical Systems: Contrasts in Evolutionary Capacity." In Artificial Life I, pp. 501–519, 1989.

J. H. Holland, "Studies of the Spontaneous Emergence of Self-Replicating Systems Using Cellular Automata and Formal Grammars." In *Automata, Languages, Development,* A. Lindenmayer and G. Rozenberg (eds), North-Holland, pp. 385–404, 1976.

J. Koza, "Artificial Life: Spontaneous Emergence of Self-Replicating and Evolutionary Self-Improving Computer Programs." In Artificial Life III, pp. 225–262, 1994.

C. G. Langton, "Self-Reproduction in Cellular Automata," *Physica D*, **10**, pp. 135–144, 1984.

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

C. G. Langton (ed), *Artificial Life*, Santa Fe Institute Studies in the Sciences of Complexity, Vol. VI, Addison-Wesley, 1988.

slide 75

Bibliography

C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen (eds), *Artificial Life II*, Santa Fe Inst. Studies in the Sciences of Complexity, Vol. X, Addison-Wesley, 1991.

C.G. Langton (ed), *Artificial Life III*, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVII, Addison-Wesley, 1994.

J. Lohn, "Automated Discovery of Self-Replicating Structures in Cellular Space Automata Models," Dept. of Comp. Sci. Tech. Rep. CS-TR-3677, Univ. of Maryland at College Park, August 1996.

B. McMullin, "The Holland α-Universes Revisited." In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life,* F. Varela and P. Bourgine (eds), MIT Press, pp. 317–326, 1992.

D. Mange, S. Durand, E. Sanchez, A. Stauffer, G. Tempesti, P. Marchal, and C. Piguet, "A New Self-Reproducing Automaton Based on a Multi-Cellular Organization." Dept. of Comp. Sci. Tech. Report, Swiss Federal Institute of Technology (Lausanne, Switzerland CH 1015), 1995.

J.-Y. Perrier, "Toward a Viable Self-Reproducing Universal Computer," *Physica D*, 97, pp. 335–352, 1996.

U. Pesavento, "An Implementation of von Neumann's Self-Reproducing Machine," *Artificial Life*, 2 (4), pp. 337–354, 1995.

slide 76

Bibliography

J. A. Reggia, S. Armentrout, H. H. Chou, and Y. Peng, "Simple Systems That Exhibit Self-Directed Replication," *Science*, 259, pp. 1282–1288, February, 1993.

M. Sipper, "Studying Artificial Life Using a Simple, General Cellular Model," *Artificial Life*, 2 (1), pp. 1–35, 1995.

A.H. Taub,*John von Neumann: Collected Works. Vol. V: Design of Computer, Theory of Automata and Numerical Analysis*, Oxford, Pergamon Press, 1961.

G. Tempesti, "A New Self-Reproducing Cellular Automaton Capable of Construction and Computation." In *ECAL95: Proceedings of the Third European Conference on Artificial Life,* F. Moran, A. Moreno, J.J. Morelo, and P. Chacon (eds), Springer, pp. 555–563, 1995.

J. W. Thatcher, "Universality in the von Neumann Cellular Model." In (Burks 1970), pp. 132–186, 1970.

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

J. von Neumann, "The General and Logical Theory of Automata." In (Taub 1961), pp. 288–328, 1951.

J. von Neumann, *Theory of Self-Reproducing Automata*, A. Burks (ed), University of Illinois Press, 1966.

S. Wolfram, *Cellular Automata and Complexity*, Addison-Wesley, Addison-Wesley, Reading, Mass, 1994.

slide 77

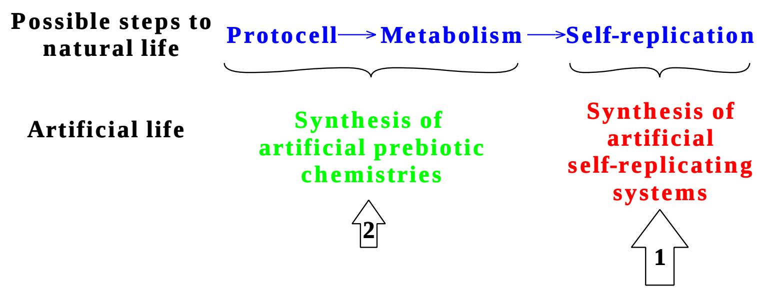
On-line References

Some on-line sources of information and software:

- WWW:
  - The Self-Replication Page (Moshe Sipper): `http://lslwww.epfl.ch/ moshes/selfrep`
  - Alife On-Line: `alife.santafe.edu`
  - Artificial Life and Genetic Algorithms: `www.brunel.ac.uk:8080/depts/AI/alife/home.htm`
  - CA simulator for self-replicating loops: `rucs2.sunlab.cs.runet.edu/ dana/ca/examples /loops/loops.html`
  - SRS and nanotech: `nano.xerox.com/nanotech/selfRepNATO.html`
  - Logic Systems Lab, Swiss Fed. Inst. of Tech.: `lslwww.epfl.ch`
- USENET Newsgroups:
  - `comp.ai.alife`
  - `comp.theory.cell-automata`

slide 78

Two Areas in Artificial Life



- Recent work on evolving catalytic reaction networks.

slide 79

Pre-genomic Processes

What kinds of dynamical processes could lead to the self-organization of protocells (precursors to the first living cells)?

Understanding such processes could potentially be useful for:

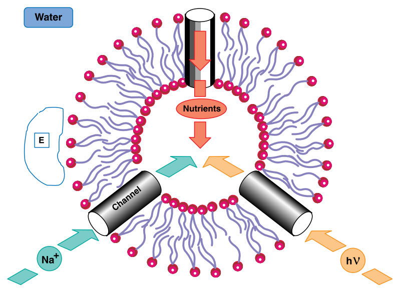
- biologists (origins of life hypotheses, astrobiology)
- computer scientists (abstract into useful problem-solving techniques)
- nanotechnologists (molecular self-assembly)

Ongoing interdisciplinary research at NASA Ames Research Center (Astrobiology and Computational Sciences).

slide 80

Model of a Protocell

Hypothetical protocell: bilayer, amphiphilic membrane



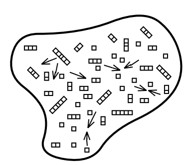
Basic functions (Pohorille, et. al., 1996):

- transport of ions across membranes
- formation of energy source (photoactivated proton gradient) to drive chemical synthesis
- peptide organization for catalytic activity

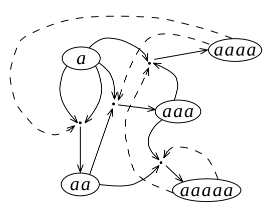
slide 81

Simple Computational Protocell Model

Interacting polymers enclosed in a protocell (well-stirred reactor)



Example of a catalytic reaction set: graphical depiction and reaction set. Reverse reactions are not shown explicitly in graph.

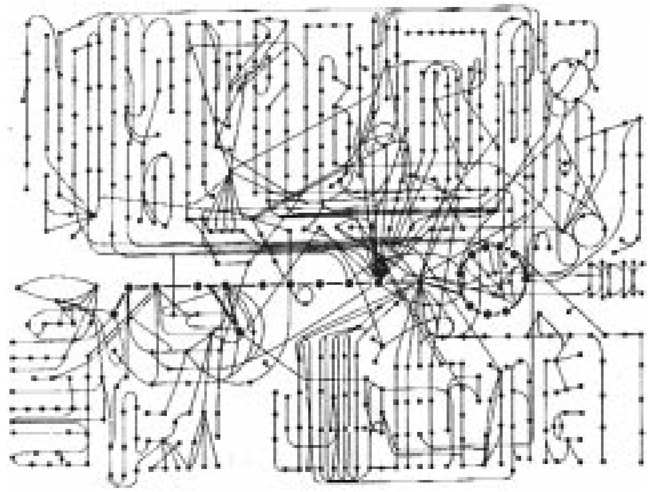


$a + a \overset{aaaa}{\rightleftharpoons} aa \qquad a + aaa \overset{aaaaa}{\rightleftharpoons} aaaa$

$a + aa \overset{aaaa}{\rightleftharpoons} aaa \qquad aa + aaa \overset{aaaaa}{\rightleftharpoons} aaaaa$

slide 82

"Simplified" view of reaction network for biological cells:



• denotes enzymatic action

from *The Cell*, Alberts, 1983

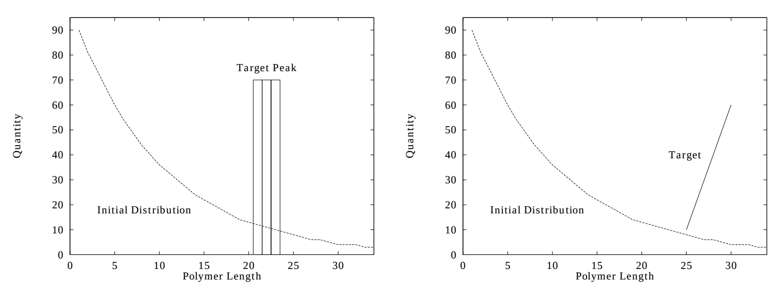
slide 83

Evolution of Catalytic Reaction Networks

Use of genetic algorithms (GAs) to artificially evolve simple artificial chemistries

Objective of GA:

Find a set of N reactions that moves a pre-specified initial distribution of polymers to an arbitrary distribution of polymers.



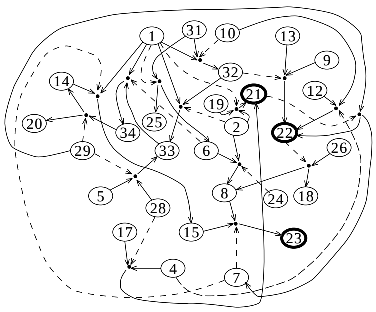
slide 84

Evolved Reaction Network

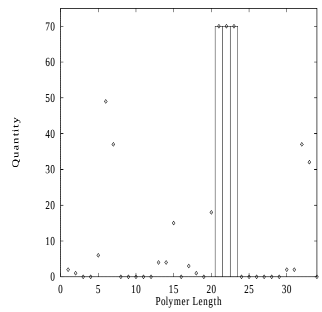
Evolved reaction network:
- short cycle formation
- key polymers acting as both reactants and catalysts
- target polymers acting as catalysts

Best found reaction set from 100 GA runs:

partial reaction graph:



resultant distribution:



Conclusions

From these early results, we have demonstrated that small, artificial chemical reaction networks can be synthesized to move a system of polymers into states of increasing complexity.

The reaction sets found are robust in the sense that they produce desirable behavior in equilibrium.

Ongoing efforts:

- restriction to more biochemically plausible reactions
- two and three-letter alphabets and longer polymers
- membrane functionality
- communities of interacting protocells