Artificial Life & Complex Systems

Lecture 12

Self-Replication

June 8, 2007

Max Lungarella

Contents

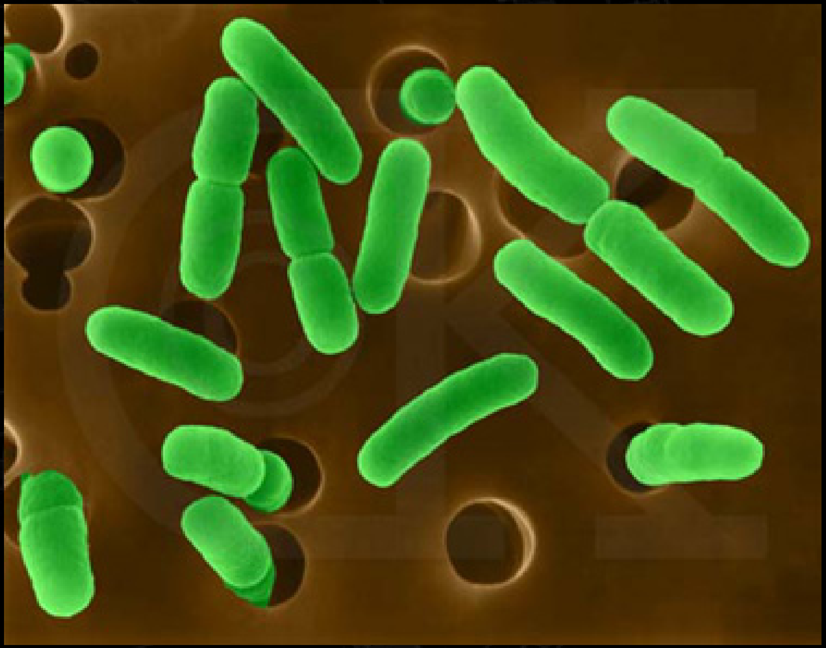
What is Self-Replication?

"The process by which a thing/entity may produce an extremely close functional copy of itself"

Examples:

- Biological cells replicate by cell division (given a suitable environment)
- Biological viruses can replicate by exploiting the reproductive machinery of cells through a process of infection
- Computer viruses replicate

Self replication



Ecoli – division and replication

What is Self-Replication?

- Around 1950, scholars began studying artificial self-replicating systems in order to gain a deeper understanding of complex systems and the fundamental <u>information-processing principles</u> involved in self-replication
- Self-replication differs from self-assembly (see lecture 13)
- Self-assembly systems are not able to make, catalyze, or induce more copies of themselves

Why Artificial Self-Replication?

- Self-replicating programs could facilitate the task of programming massively parallel computers (Ray, 1992)
- Process of self-replication could shed light on computer viruses and may contribute to the creation of bio-inspired "immune systems" (Kephart, 1994)
- Self-replicating devices could play a role in atomic-scale manufacturing or "nanotechnology" (Drexler, 1989; Merkle, 1994)
- SRS may have an important future role in planetary exploration and in creating robust electronic hardware (Freitas, 1982; Mange et al., 1996)
- SRS could shed light on theories of the origins of life (Orgel ,1992)

A Note on Replication and Reproduction

Replication is an ontogenetic, developmental, process, involving no genetic operators, resulting in an exact duplicate of the parent organism.

Reproduction is a phylogenetic, evolutionary, process involving genetic operators such as crossover and mutation, thereby giving rise to variety and ultimately to evolution

(Distinction due to Sipper et al., 1997)

A Note on Replication and Reproduction

Because self-replication seeks to copy an entire system without error, it is not sufficient for life UNLESS it allows for the possibility of heritable mutations (self-reproduction)

Self-replicants are thus a safer form of self-reproducers because incapable of acquiring any significant variations, of, if variations are acquired, may become nonfunctional

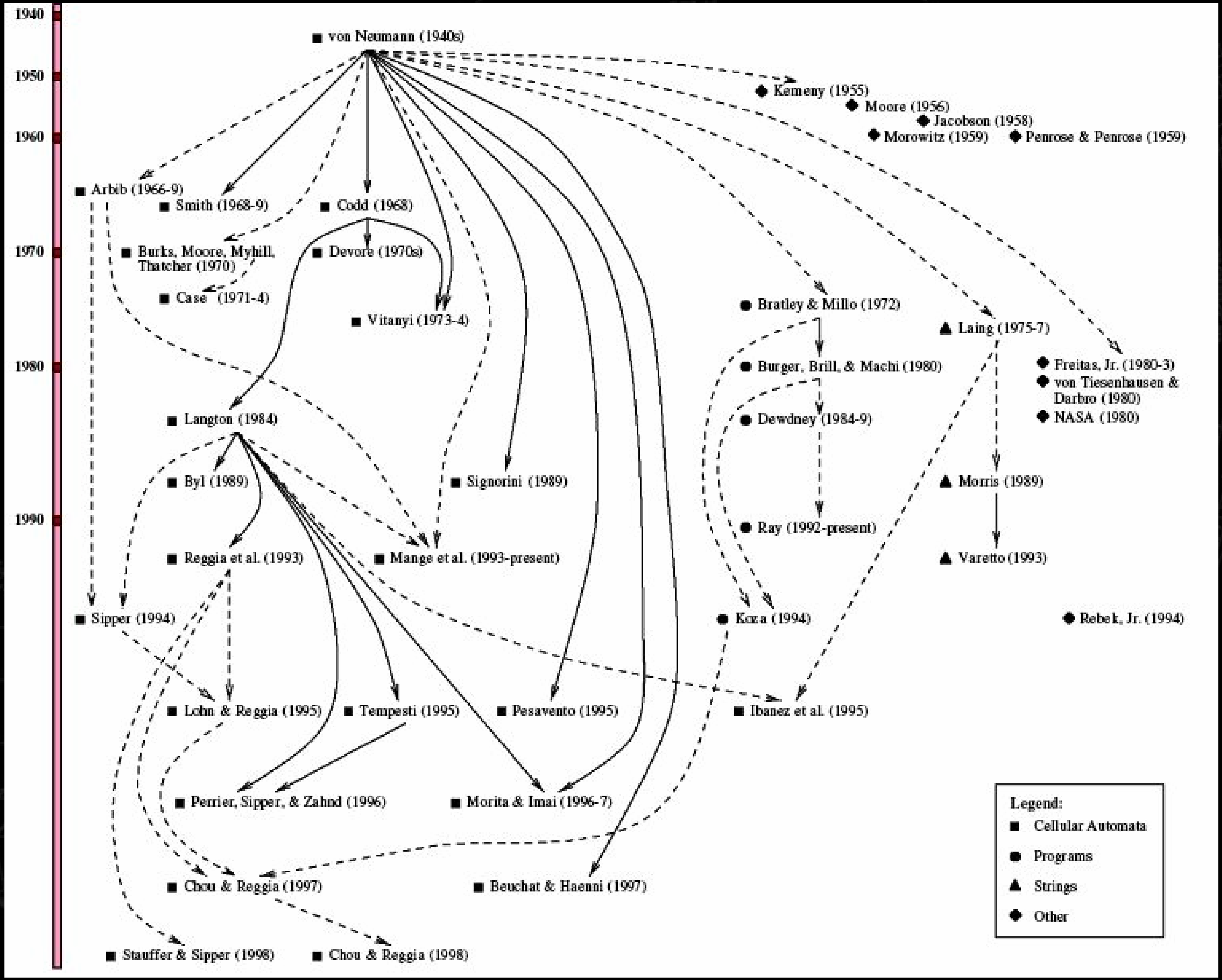
Brief History (Non-Exhaustive)

"Serious" scientific study of artificial self-replicating structures or machines have been underway for more then 70 years:

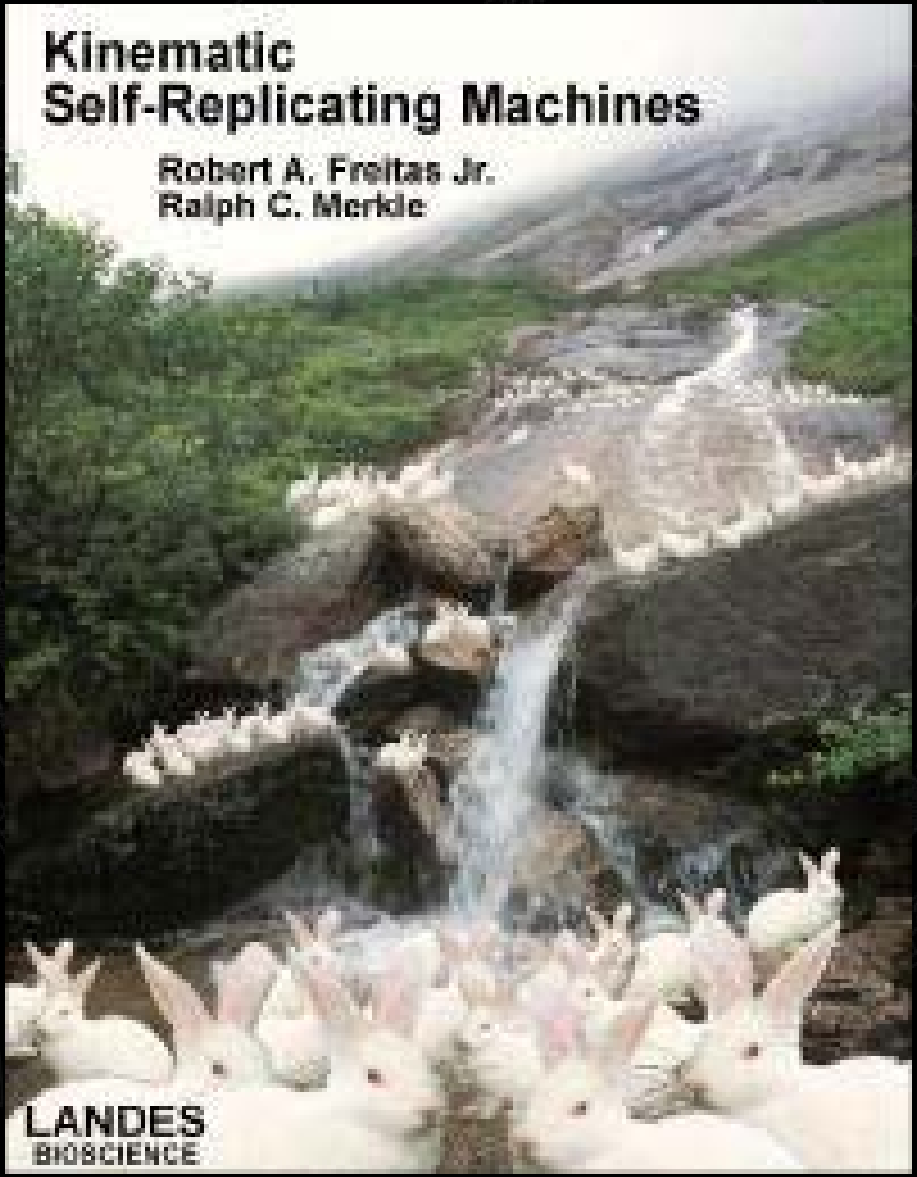
- "The world, the flesh, and the devil" (Bernal, 1929)
- *Recursion theory*, quine (program that produces its complete source code as output; named after W.van Orman Quine, and inspired by Kleene (1930))
- *Universal constructor-computer*: J. von Neumann (1950s), E.F.Codd (1968)
- *Mechanical self-replicator*: L.Penrose and R.Penrose (1958)
- *Sexually reproducing CA*: P.M.B.Vitanyi (1973)
- *Self-replicating loops*: C.G.Langton (1984), J.Byl (1989), J.A.Reggia et al (1993), M. Sipper (1994)
- *"Engines of Creation: The Coming Era of Nanotechnology"* (Drexler, 1986)
- *"Self-replicating systems and molecular manufacturing"* (Merkle, 1992)
- *Self-replicating loops with finite computational capabilities*: G.Tempesti (1995)
- *Self-replicating loops capable of universal computation*: J.Y.Perrier et al (1996)
- *Spontaneous emergence of self-replicating loops*: H.H.Chou and J.A.Reggia (1997)
- *Structurally dissolvable self-reproducing loop*: H.Sayama (1998)
- *Modular self-replicating robots:* Murata et al. (1995s-), Lipson et al (2002-)
- ...

Brief History

Sipper (1998)



For Heaps of More Recent History



Self-Replicating Machines

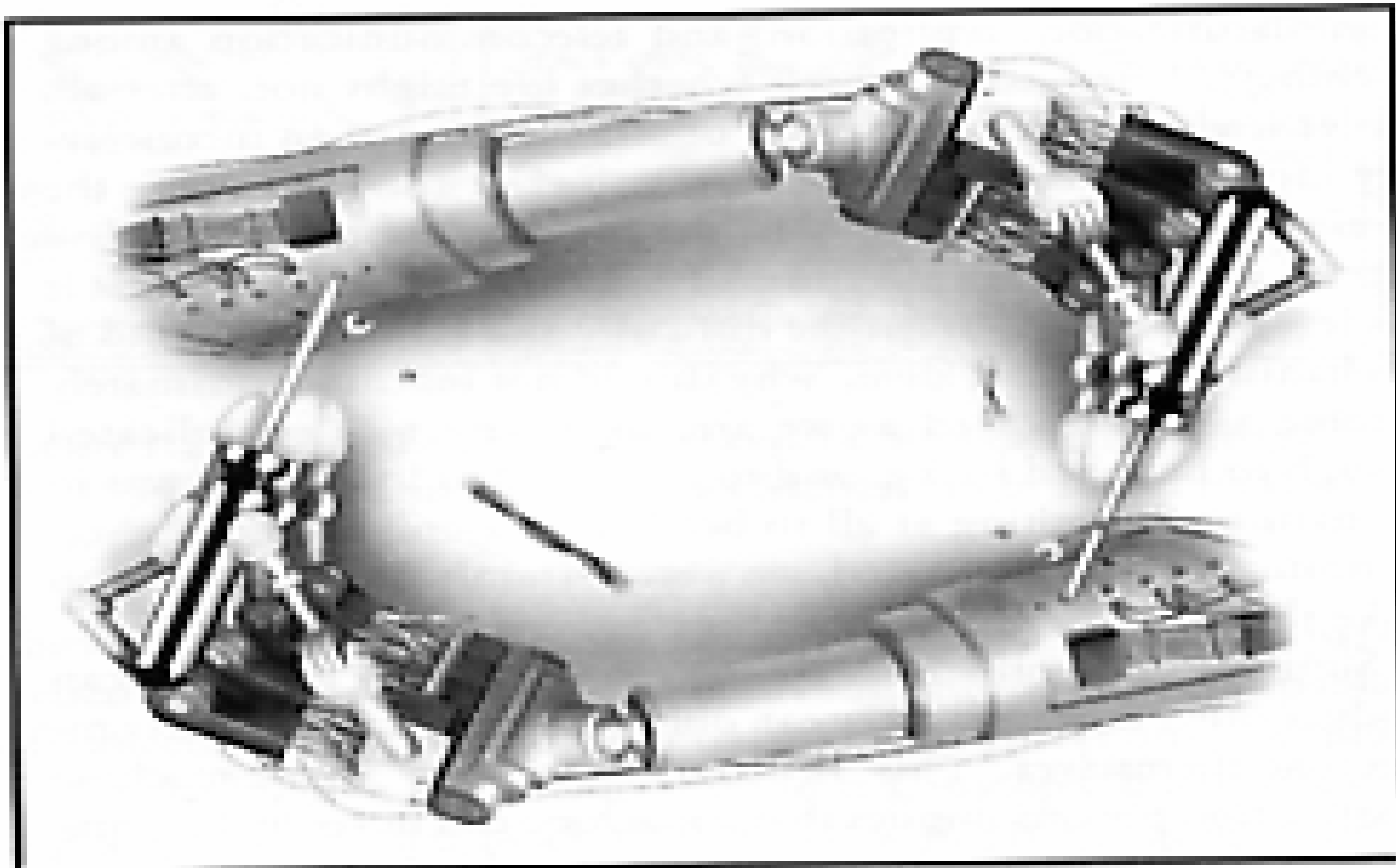
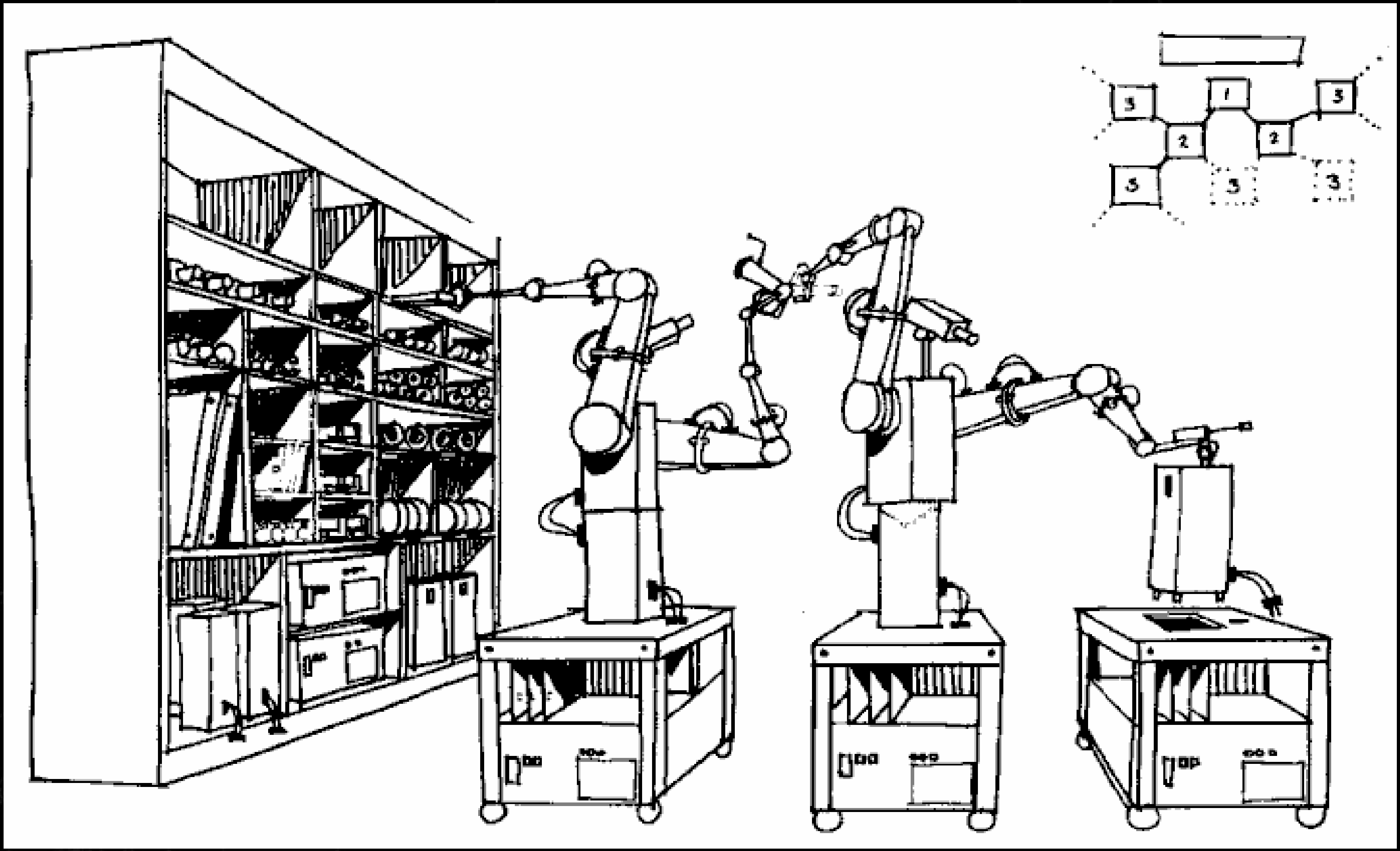


Figure 1.3. Robots made of classical components can make identical copies of themselves and thus self-replicate, but quantum systems cannot.[276] (image courtesy of Cameron Slayden and Science/AAAS[277])

Proposed Demo of Simple Robot Self-Replication



NASA (1982)

Lunar Factories

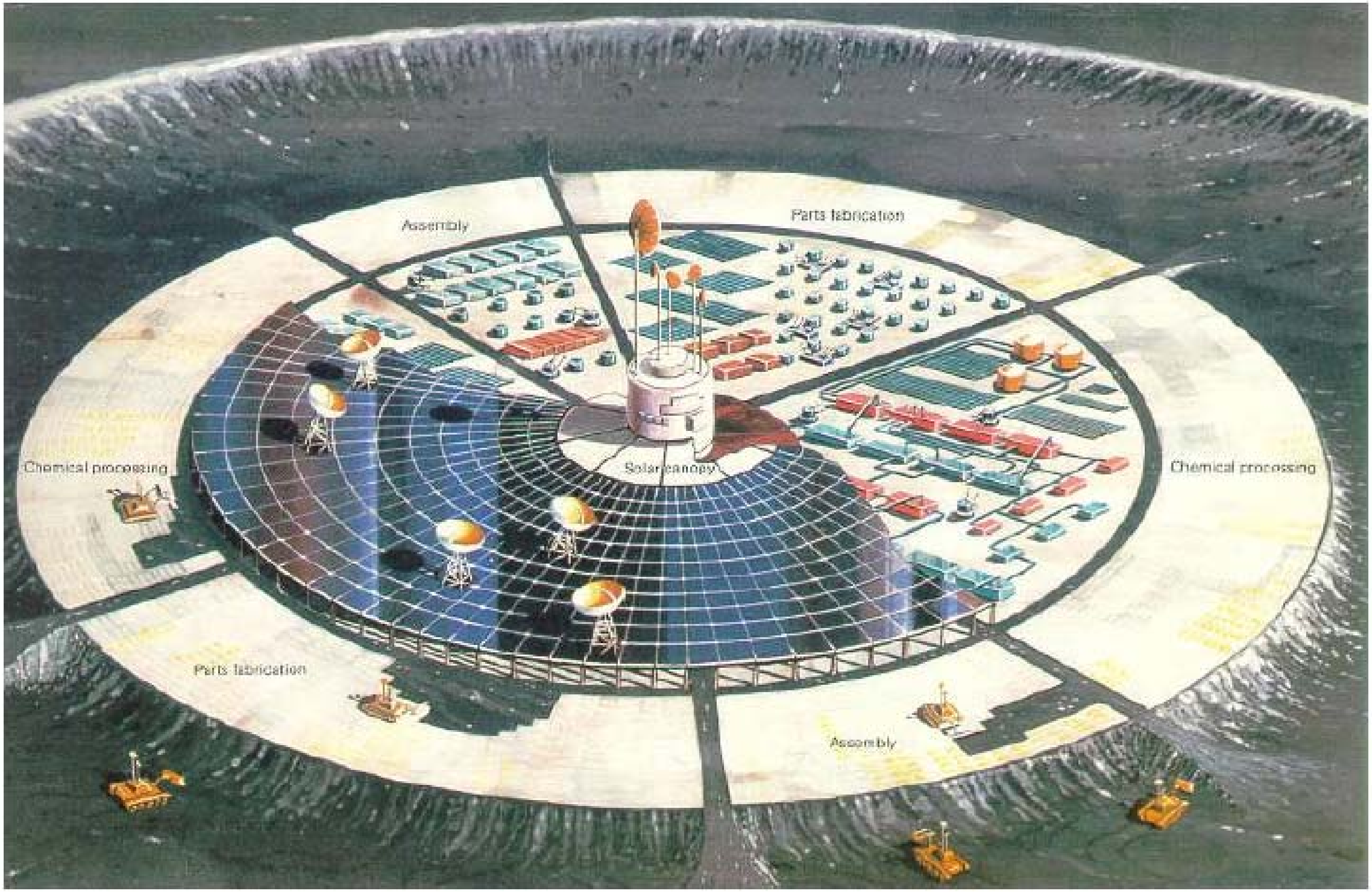
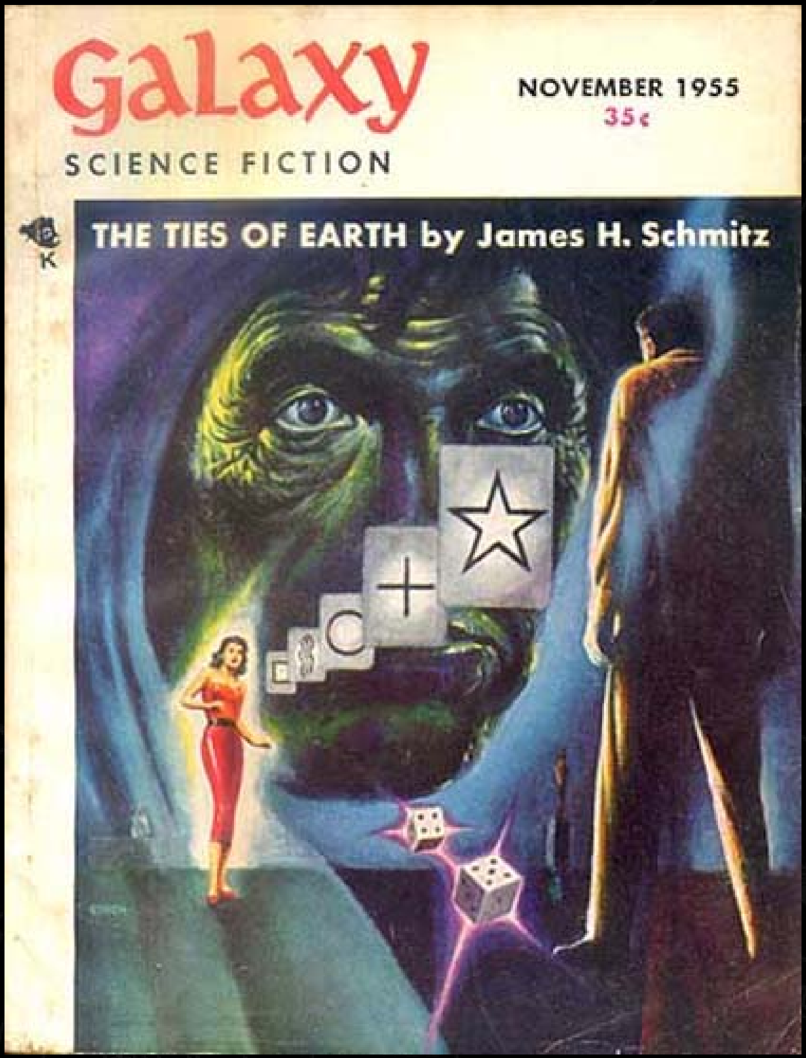


Figure 1: *Artists impression of self-replicating lunar factory.*

NASA (1982)

Self-Replicating Machines: Autofac



The pellet was a smashed container of machinery, tiny metallic elements too minute to be analyzed without a microscope [...] The cylinder had split. At first he couldn't tell if it had been the impact or deliberate internal mechanisms at work. From the rent, an ooze of metal bits was sliding. Squatting down, O'Neill examined them. The bits were in motion. Microscopic machinery, smaller than ants, smaller than pins, working energetically, purposefully - constructing something that looked like a tiny rectangle of steel. “They’re building,” O’Neill said, awed. He got up and prowled on. Off to the side, at the far edge of the gully, he came across a downed pellet far advanced on its construction. Apparently it had been released some time ago. This one had made great enough progress to be identified. Minute as it was, the structure was familiar. The machinery was <u>building a miniature replica</u> of the demolished factory.

Philip K. Dick (1955)

Self-Replicating Machines

1984

Self-Replicating Machines

Agent Smith!

1999

Self-Replicating Machines

1999

Self-Replicating Machines

Attempt to a definition:

“A self-replicating machine is an artificial self-replicating system that relies on conventional large-scale technology and automation”

- The term “clanking replicator” is sometimes used to distinguish such systems from the microscopic nanobots or “assemblers” that nanotechnology may make possible (see E. Drexler)
- Self-replicating machines are also called “von Neumann machines” (or, universal constructor) after John von Neumann, who first rigorously studied the idea

Food for Thoughts: Interestingly, …

Astronomer Robert Jastrow notes:

"The computer – a new form of life dedicated to pure thought – will be taken care of by its human partners, who will minister to its bodily needs with electricity and spare parts. Man will also provide for computer reproduction. Computers do not have DNA molecules; they are not biological organisms. We are the reproductive organs of the computer. We create new generations of computers, one after another …"

Similar thoughts have been put forward by Samuel Butler, Mark Tilden, …

Logical Organization of Self-Replication

Von Neumann's goal: Formalization and logical organization of the process of self-replication; what logical organization is sufficient for an automaton to be able to reproduce?

Von Neumann wished to abstract from the natural self-reproduction problem its logical form (Burks, 1966)

Von Neumann had in mind five (or more) self-replicating machine models:

- **Kinematic machine**
- **Cellular machine**
- Neuron-type machine
- Continuous machine
- Probabilistic machine

Question Posed by von Neumann

1. Universal computation
2. Capability of construction
3. Universal construction
4. Self-reproduction
5. Complexity-increasing evolution (remains unsolved)

Self-Replication: Sufficient Characteristics

Sufficient characteristics and capabilities for replication:

1. Logical (or computational) universality: ability to function as a general-purpose computing machine able to simulate a UTM (and thus execute any computational task); deemed necessary because a replicating machine must be able to read instructions to carry out complex computations
2. Constructional universality: ability to manufacture any kind of finitely sized machines which can be formed from specific kinds of parts, given a finite number of different types of parts and an indefinitely large supply of parts of each type
3. Construction capability: to self-replicate a machine must be capable of manipulating information, energy, and materials of the same sort of which it itself is composed

Self-Replication: Sufficient Characteristics

- Essentially a Turing machine that can not only read and execute instructions, but that can also make copies of itself
- Capable of "universal construction", i.e. capability of building any machine given its description (if supplied with proper materials and is given correct instructions) → see notion of "universal computation"!

Theoretical Self-Replicating Machine

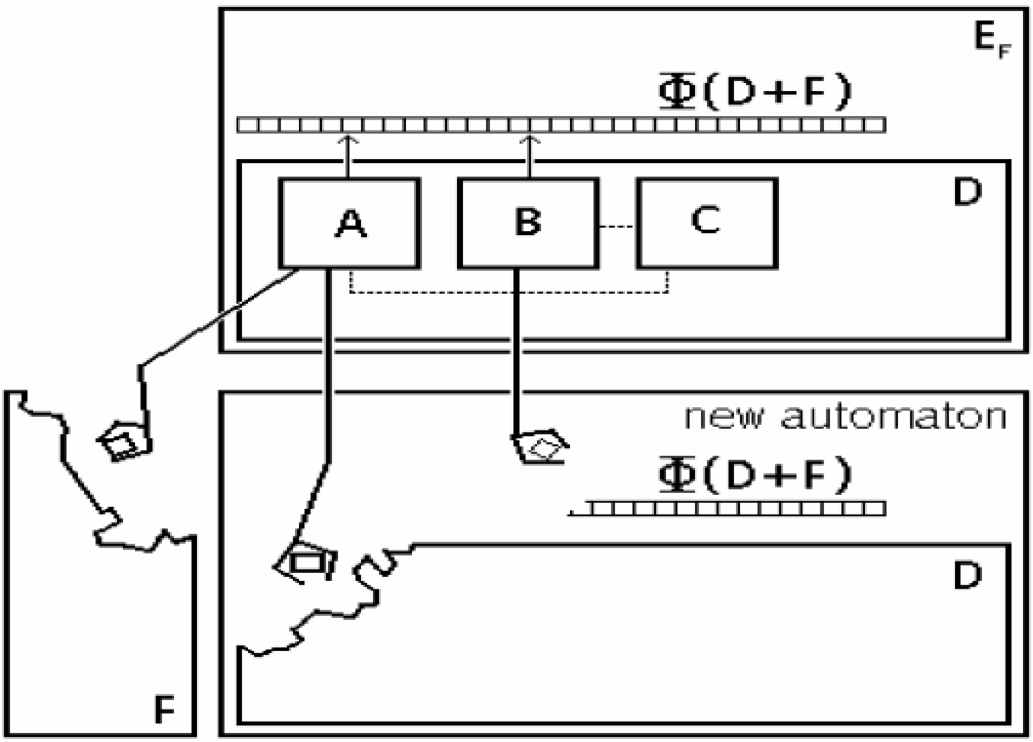


Figure 2: *Schematic view of von Neumann's theoretical self-replicating machine. A: general construction machine; B: general copying machine; C: control machine.*

Formalization of Self-Replication

A: General constructing machine ("constructor") that can build a machine **X**

B: General copying machine ("blueprint copier")

C: Control machine that controls the constructor and the copier

Φ(X): Set of blueprints (instructions) describing how to build **X** (i.e. A, B, and C)

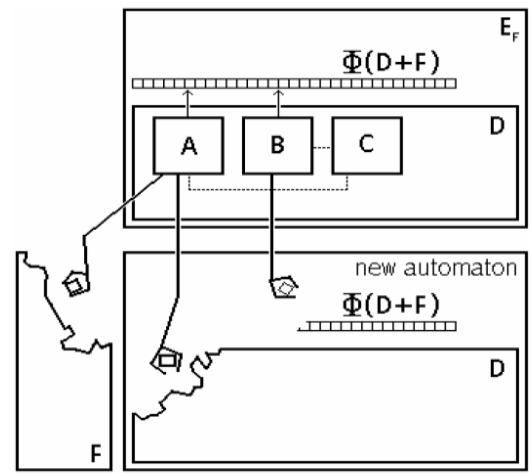


Figure 2: Schematic view of von Neumann's theoretical self-replicating machine. A: general construction machine; B: general copying machine; C: control machine.

Let us assume a machine **X** is desired

A takes **Φ(X)** and constructs **X**: A + Φ(X) → X

B takes **Φ(X)** and copies it: B + Φ(X) → Φ(X)

C activates **A** and **B** in the right order, and then connects **X** and **Φ(X)**, and separates them from the original machine:

A + B + C + Φ(X) → X + Φ(X)

Formalization of Self-Replication

Let us assume: $X = A + B + C$

The entire replicator is: $A + B + C + \Phi(A + B + C)$

Controller C actuates B which copies $\Phi(A + B + C)$ to produce a second copy of $\Phi(A + B + C)$, and then actuates A (using the copy) to produce $A + B + C$ (another constructor, copier, and controller)

The results is a second automaton: $A + B + C + \Phi(A + B + C)$

(self-replication takes place in a sea of spare parts)

$$A + B + C + \Phi(A + B + C) \rightarrow A + B + C + \Phi(A + B + C)$$

Formalization of Self-Replication

Let us define **$D = A + B + C$**

Let us assume that **$X = D + F$** where **F** is any arbitrary automaton

Remember: **$A + B + C + \Phi(X) \rightarrow X + \Phi(X)$** (self-replication)

Then

$$D + \Phi(D + F) \rightarrow D + F + \Phi(D + F)$$

In other words: the machine D not only makes a copy of itself but it also produces an object F; D has become a productive general-purpose manufacturing system

From Theory to Biology

Von Neumann's early schema was later confirmed by subsequent research on the molecular biology of cellular reproduction (starting from the discovery of the DNA):

1) Component **A** (constructor): <u>ribosomes</u> and supporting cellular mechanisms
2) Component **B** (copier): <u>DNA polymerase enzymes</u> (set of mechanisms that copy and repair the genome)
3) Component **C** (controller): <u>repressor and depressor molecules</u> and associated expression-control machinery in the cell
4) Component **Φ(A+B+C)**: genetic material <u>DNA</u> that carries the organism's genome

The correspondence is of course not complete because cells include additional complexities

Von Neumann Kinematic Beast (1948,1949)

Hypothetical self-replicating robot consisting of:

- Four logic elements (receive and trasmit stimuli)
  - Stimulus organ (truth function 'P OR Q')
  - Coincidence organ (truth function 'P AND Q')
  - Inhibitory organ (truth function 'P AND NOT-Q')
  - Stimuli-producing organ (source of stimuli)
- Four mechanical elements (skeleton and mobility)
  - Rigid member (an insulated girder carrying no stimuli that can form a rigid frame)
  - Fusing organ (welds and solders two parts together when stimulated)
  - Cutting organ (unsolders connection when stimulated)
  - A muscle (normally rigid, connected to the parts, producing motion when stimulated)

Environment: large lake (or sea) containing millions of spare parts as its source of raw material

Kinematic Model of Machine Replication

"The machine has a <u>memory tape</u> which instructs it to go through certain mechanical procedures. Using a manipulative appendage and the <u>ability to move</u> around in its environment, the device can <u>gather and connect parts</u>. The tape-program first instructs the machine to reach out and pick up a part, then to go through an identification routine to determine whether the part selected is or is not the specific one called for by the instruction tape. If not, the component is thrown back into the '<u>sea</u>' and another is withdrawn for similar testing, and so on, until the correct one is found. Having finally identified a required part, the device searches in like manner for the next, then joins the two together in accordance with instructions. The machine continues <u>following the instructions</u> to make something, without really understanding what it is doing. When it finishes, it has produced a <u>physical duplicate</u> of itself. Still, the second machine does not yet have any instructions, so the parent machine copies its own memory tape onto the blank tape of its offspring. The last instruction on the parent machine's tape is <u>to activate the tape of its progeny</u>."

(Freitas and Merkle, 2004)

Kinematic Model of Machine Replication

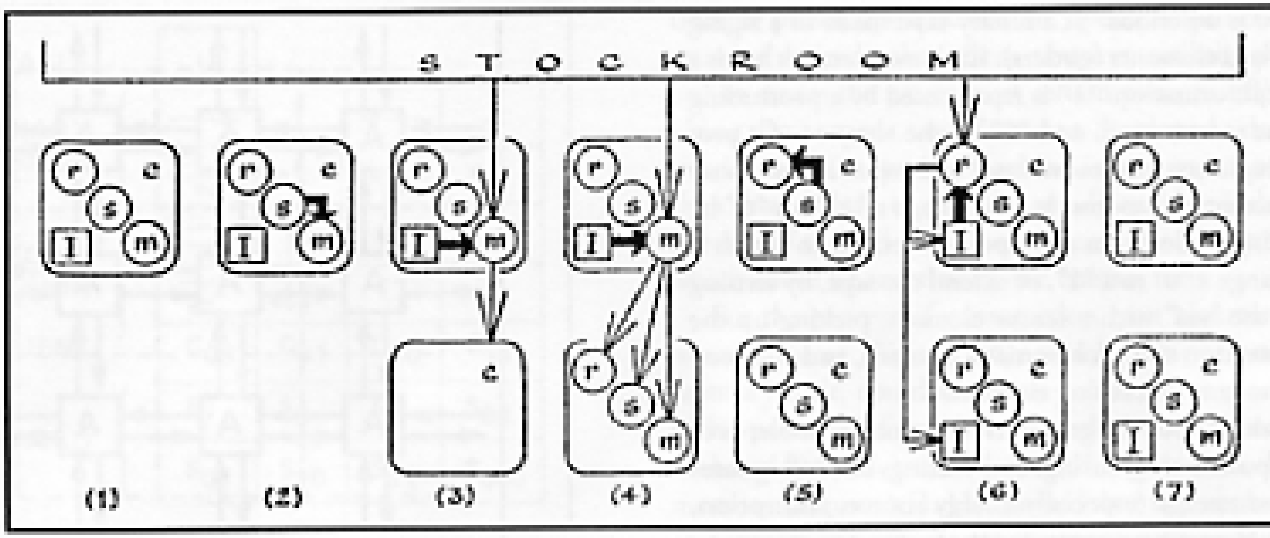
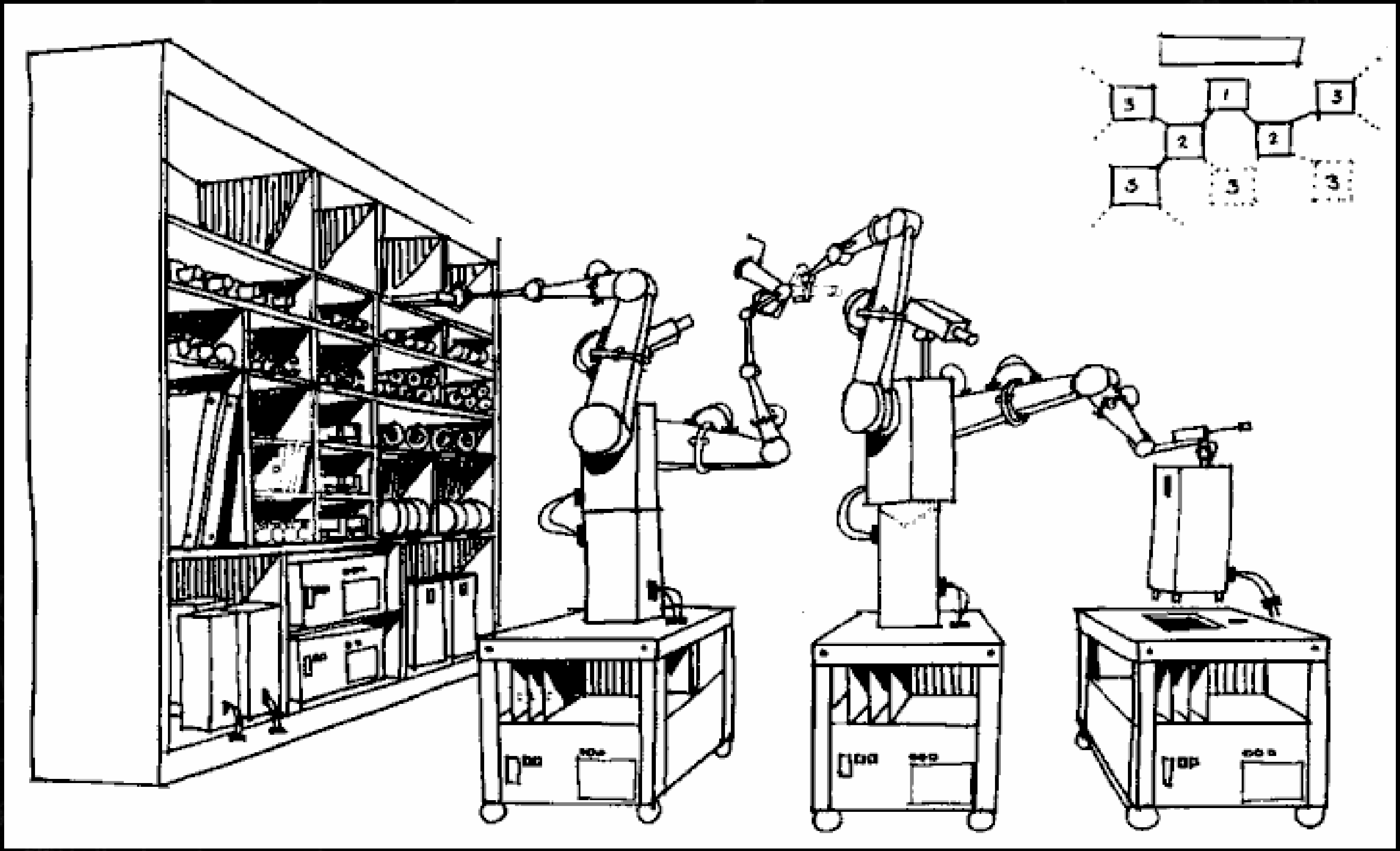


Figure 2.1. Schematic of von Neumann kinematic replicator, from Cairns-Smith.[319] The kinematic replicator machine consists of a chassis "c" which holds a box of instructions [I], machinery (m) and (r) for acting on and for replicating the instructions, respectively, and a time-switch or sequencer (s). Replication proceeds as follows:[319]

1. Resting phase.
2. Sequencer turns on (m).
3. (m) makes another chassis from materials in the stockroom, following instructions drawn from [I].
4. (m) makes and installs another manufacturing unit (m), another instruction replicator (r), and another sequencer (s). (The latter is possible because this machinery is being instructed from outside itself.)
5. Sequencer turns off (m) and turns on (r).
6. (r) takes recording material (e.g., blank punch cards or magnetic tape) from the stockroom and duplicates [I], then installs the copied instructions in the offspring machine, producing a second machine identical to the first.
7. Resting phase....

Kinematic Model of Machine Replication



NASA (1982)

Von Neumann Kinematic Beast

Five main flaws:

1) Geometric-kinematic constraints are dealt with, but force and energy (i.e. energy source, dissipation, absorption) are neglected
2) The beast was completely hypothetical; was never realized in hardware nor even fully designed
3) Total number of parts was thought to be large, possibly running "into hundreds of thousands, or millions" (sea of spare parts)
4) The body of the von Neumann kinematic machine "would be a box containing a minimum of 32,000 constituent parts (likely to include rolls of tape, pencils, erasers, vacuum tubes, dials, photoelectric cells, motors, batteries, and other devices)" *(L. Penrose, 1959)*
5) Rigorous analysis extremely difficult

The CA Model of Machine Replication

- Kinematic model was mathematically inelegant and was not susceptible to rigorous treatment
- Idea: ‘Cell space’ format might lead to notion of self-replicating machine amenable to mathematical treatment
- Implementation: Checkerboard system with identical FSA in each square; cell automata can assume 29 possible states (5-bit encoding); communication is with four cardinal directions (NEWS); emergence of coordinated global behavior from “simple” local rules

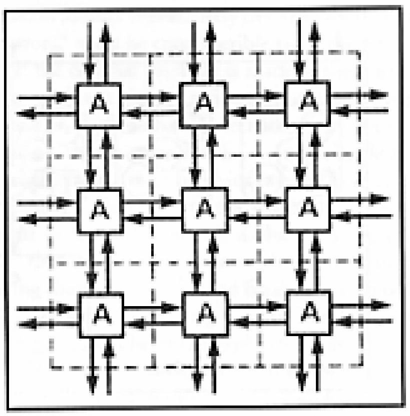


Figure 2.2. Finite state automaton cellular space, from von Neumann.[4]

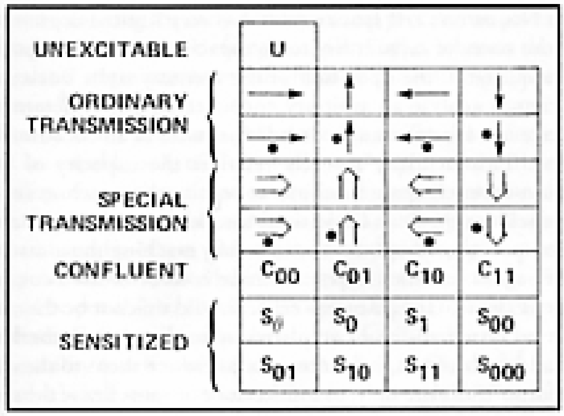


Figure 2.3. Twenty-nine states of von Neumann's cellular automata, from von Neumann.[4]

The CA Model of a Universal Constructor

Constructing arm directed by constructing unit, used to build the offspring

Constructing unit: a machine capable of reading the memory tape and interpreting its contents

Avoid internal self-contradiction (no description of a description of a description of …)

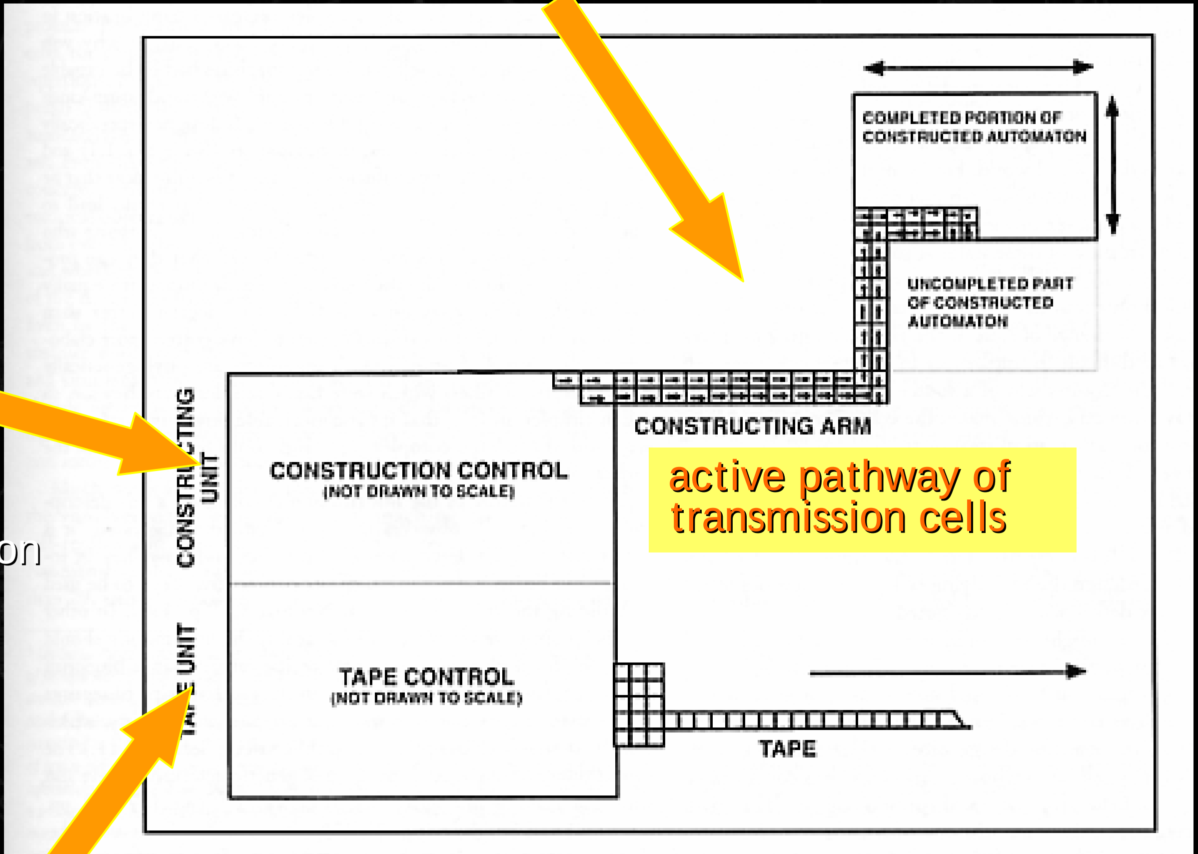


Figure 2.4. Universal construction in the cellular automata model of machine replication, from von Neumann.[4]

Tape unit reads tape which contains a complete description of the machine to be built (50’000-200’000 x 29-state cells!)

The CA Model of Machine Replication

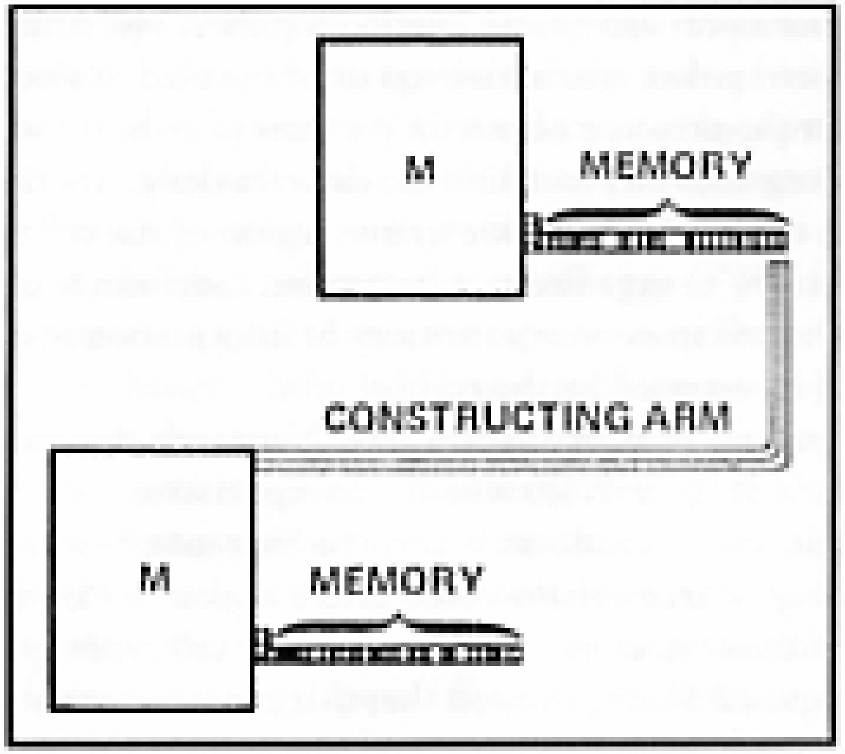


Figure 2.5. Universal construction arm builds the memory tape in the cellular automata model of machine replication, from von Neumann.[4]

The CA Model of Machine Replication

- Initialization: almost all cells are in state “U” (unexcitable; passive state)
- Each cell updates according to rules of local FSA and states of neighbors
- Organism “reads” tape and executes instruction contained in it
- Constructing arm is extended and manufacturing of offspring is started
- When offspring is ready, the instructions call for the parent machine to make a copy of the instructions in its memory and to feed them into the memory of the newly constructed machine
- Finally, the arm (“umbilical cord”) is dissolved and offspring is activated

Note: first, information stored on tape is interpreted (instruction) and then copied (uninterpreted; data)

The CA Model of Machine Replication

- Von Neumann was able to formally demonstrate that this CA model of machine replication possessed several <u>sufficient logical properties</u> including computational universality, construction capability, and construction universality

The CA Model of Machine Replication

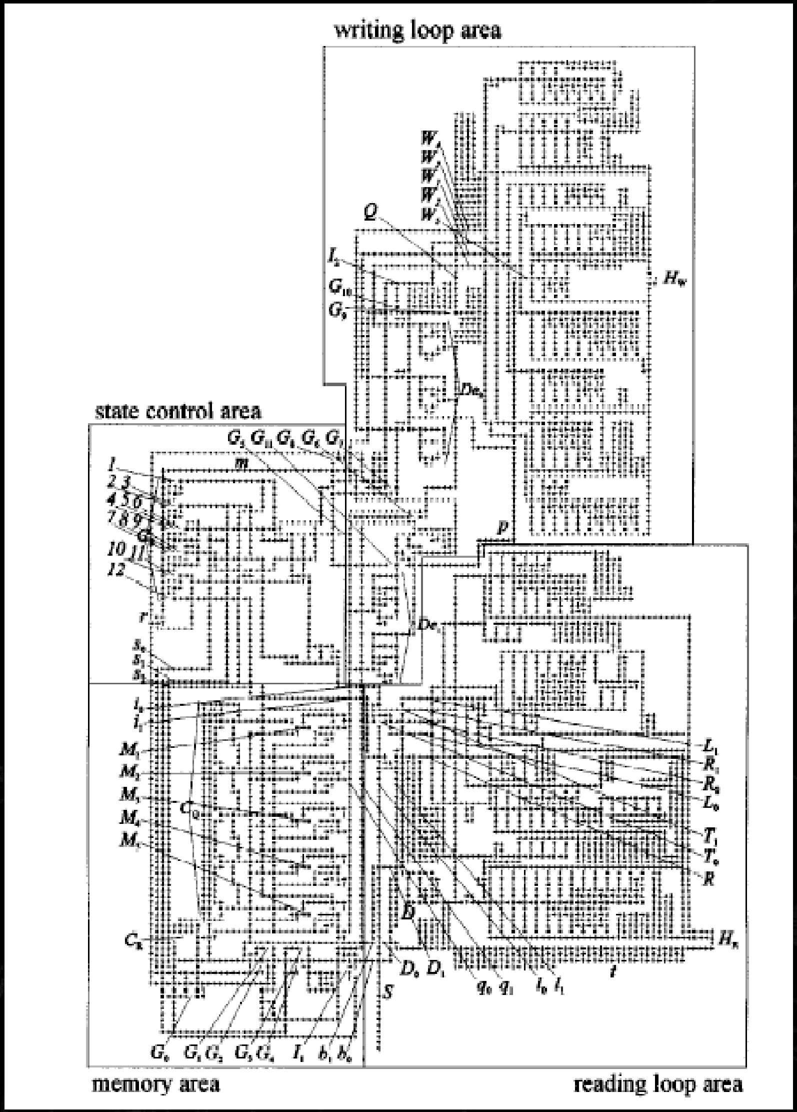
Problem: Way too complicated!

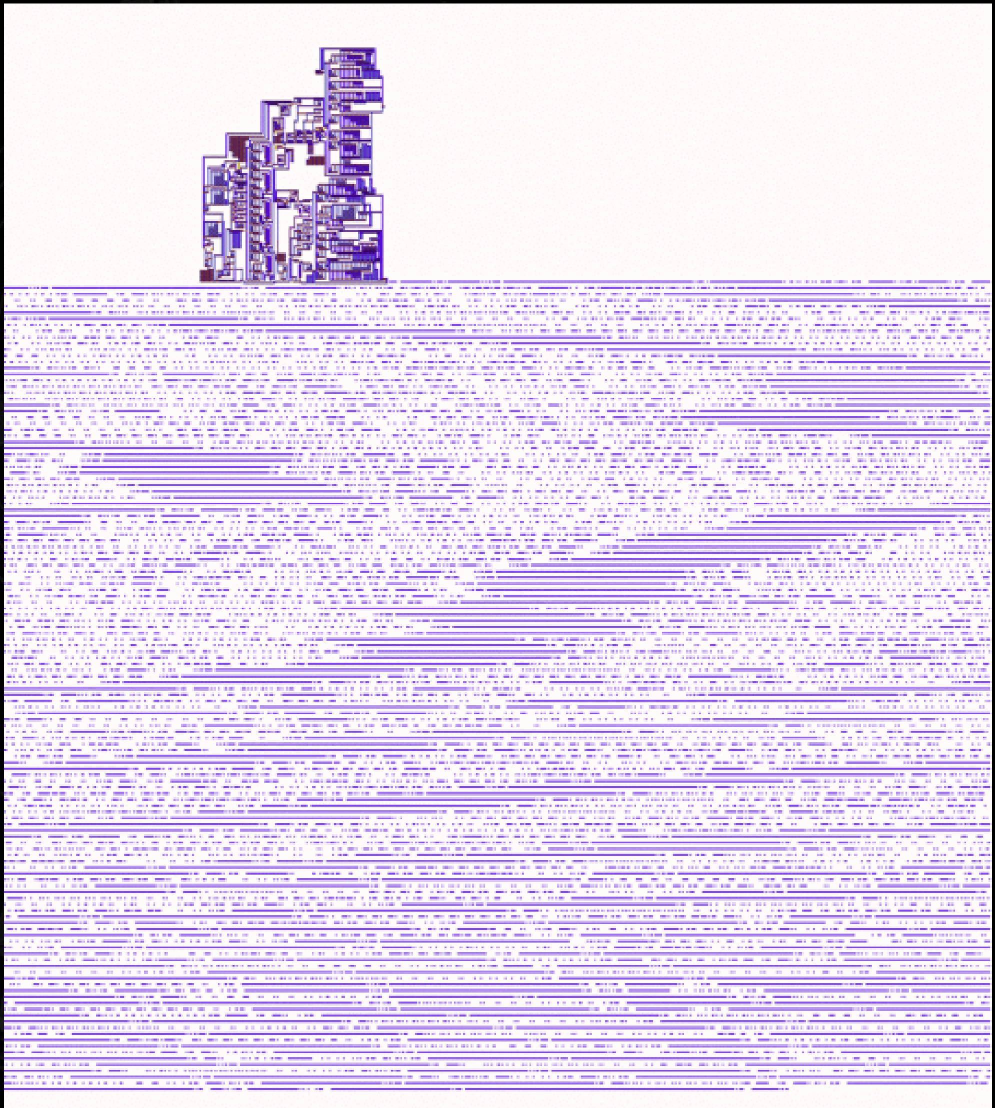
- CA had 29 states
- Each organism is composed of many functional blocks (organs: decoders, pulsers, crossing organs, ...) in the shape of a box spanning 80 cells by 400
- The blueprint is stored in a huge tape composed of approximately 150’000 cells (literature disagrees on exact number)

Improvements on Original Design

- Codd (1968): 29 → 8 states, 100’000’000 cells
  - Still too much (even) for simulation
- Devore and Hightower (1992): 8 states, 95’000 cells
- Nobili and Pesavento (1994): Software implementation
- Beuchat and Haenni (2000): Partial HW implementation (FPGA)

Implementation in 1994 – DEMO (JvN)





Nobili and Pesavento (1994); Pesavento (1996)

Intermediate Conclusions

- Complexity of early CA models seemed consistent with remarkable complexity of biological self-replicating system
- From an information-processing perspective, self-replication is an inherently complex phenomenon
- Question: What kind of logical organization is <u>necessary</u> for an automaton to reproduce itself? (converse to von Neumann's question)

Non-Trivial Self-Replicating Automata

- Question of "necessity" is not precise because it admits trivial solutions
- Distinction between trivial and non-trivial
- Replication is <u>actively directed</u> by the automaton itself, rather than being a mere consequence of the transition rules ("transition physics")
- Of course: construction universality rules out trivial cases
- But then: where earliest self-replicating molecules capable of universal construction??

Non-Trivial Self-Replicating Automata

Relaxing the criteria:

- No universal construction
- No universal computation
- Construction of copy should be actively directed by configuration itself (trivial cases are passive)
- Transition physics should be exploited
- Interaction with cellular space (environment)

Self-Replicating Loops

- Langton loops
- Sayama SDSR Loops
- Evoloops
- Tempesti loops

Langton Loop

- Von Neumann’s UC was very complex
- Reason: self-reproduction as a particular case of construction universality
- Langton approached the problem by attempting to define the simplest CA capable exclusively of self-reproduction; a self-replication structure just needs to be able to replicate its OWN structure!
- Key criterion: automaton must treated stored information in two different manners: interpreted (translation; instructions) and uninterpreted (transcription; copy)
- As in Codd (1968), Langton used 8 states and a von Neumann neighborhood

Langton Loop

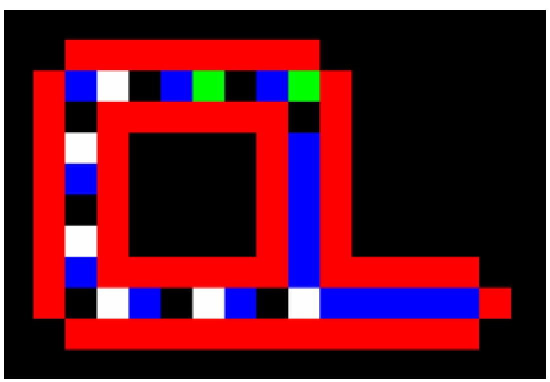
Primary characteristic emphasized by Langton is the two different modes in which information is used:

- Interpreted information (translation: information used to extend or bend the constructing arm; execution of instruction as they reach end of arm)
- Uninterpreted information (transcription: information copied at the umbilical cord; duplication of signals at arm junctions)

Langton Loop - Fact Sheet

- Inspired by Codd's "periodic emitter" element (originally used for timing purposes)
- Langton loop consists of a single loop
- Sole purpose (functionality): self-replication (no computing or constructing capabilities)
- 86 active 8-state cells and a few hundred transition rules
- Time to replicate = 151 steps

Langton (1984)



```
 22222222
2170140142
2022222202
272    212
212    212
202    212
272    212
21222222122222
20710710711111 2
 2222222222222
```

Figure 4: *Initial configuration of a Langton Loop. Note the red "sheath" (state 2) that protects the flowing information. The data flows round the loop anti-clockwise. It is copied at the junction of the arm and one copy is sent round the loop, to keep the structure "alive", and the other copy is sent down the arm to be interpreted into extending and bending it.*

Langton Loop – Data Path

- As in von Neumann's design there is a construction arm
- The data path provides a means for a signal to flow

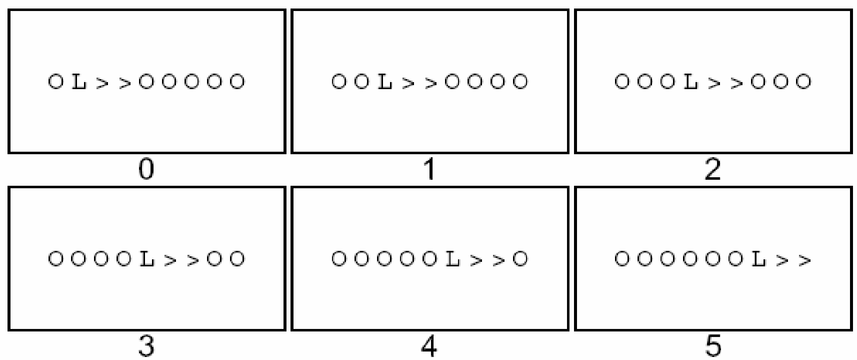


Figure 4: Signal sequence flow over a data path. Each box is a snapshot of the same region in the cellular automata space during successive times (iterations). Numbers below each box denote the times. The rules forming the transition function are such that signal > is followed by either a signal > or by the signal L. Signal L always changes to O, the latter changing to > if pointed at by >. The net effect is that the signal sequence L>> progressively moves to the right at a rate of one cell per unit time.

From Reggia et al. (1998)

Langton Loop

- The data path is sheathed by ‘2’
- The “program” consists of individual instructions ‘70’ (“extend current data path by one cell”), and ‘40’ (“extend and turn left”)
- Program = “70-70-70-70-70-70-40-40”

	2	2	2	2	2	2	2	2					
2	1	7	0	1	4	0	1	4	2				
2	0	2	2	2	2	2	2	0	2				
2	7	2					2	1	2				
2	1	2					2	1	2				
2	0	2					2	1	2				
2	7	2					2	1	2				
2	1	2	2	2	2	2	2	1	2	2	2	2	2
2	0	7	1	0	7	1	0	7	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2	2

Langton Loop - 207 Transition Rules

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00000->0	02527->1	11322->1	20242->2	30102->1
00001->2	10001->1	12224->4	20245->2	30122->0
00002->0	10006->1	12227->7	20252->0	30251->1
00003->0	10007->7	12243->4	20255->2	40112->0
00005->0	10011->1	12254->7	20262->2	40122->0
00006->3	10012->1	12324->4	20272->2	40125->0
00007->1	10021->1	12327->7	20312->2	40212->0
00011->2	10024->4	12425->5	20321->6	40222->1
00012->2	10027->7	12426->7	20322->6	40232->6
00013->2	10051->1	12527->5	20342->2	40252->0
00021->2	10101->1	20001->2	20422->2	40322->1
00022->0	10111->1	20002->2	20512->2	50002->2
00023->0	10124->4	20004->2	20521->2	50021->5
00026->2	10127->7	20007->1	20522->2	50022->5
00027->2	10202->6	20012->2	20552->1	50023->2
00032->0	10212->1	20015->2	20572->5	50027->2
00052->5	10221->1	20021->2	20622->2	50052->0
00062->2	10224->4	20022->2	20672->2	50202->2
00072->2	10226->3	20023->2	20712->2	50212->2
00102->2	10227->7	20024->2	20722->2	50215->2
00112->0	10232->7	20025->0	20742->2	50222->0
00202->0	10242->4	20026->2	20772->2	50224->4
00203->0	10262->6	20027->2	21122->2	50272->2
00205->0	10264->4	20032->6	21126->1	51212->2
00212->5	10267->7	20042->3	21222->2	51222->0
00222->0	10271->0	20051->7	21224->2	51242->2
00232->2	10272->7	20052->2	21226->2	51272->2
00522->2	10542->7	20057->5	21227->2	60001->1
01232->1	11112->1	20072->2	21422->2	60002->1
01242->1	11122->1	20102->2	21522->2	60212->0
01252->5	11124->4	20112->2	21622->2	61212->5
01262->1	11125->1	20122->2	21722->2	61213->1
01272->1	11126->1	20142->2	22227->2	61222->5
01275->1	11127->7	20172->2	22244->2	70007->7
01422->1	11152->2	20202->2	22246->2	70112->0
01432->1	11212->1	20203->2	22276->2	70122->0
01442->1	11222->1	20205->2	22277->2	70125->0
01472->1	11224->4	20207->3	30001->3	70212->0
01625->1	11225->1	20212->2	30002->2	70222->1
01722->1	11227->7	20215->2	30004->1	70225->1
01725->5	11232->1	20221->2	30007->6	70232->1
01752->1	11242->4	20222->2	30012->3	70252->5
01762->1	11262->1	20227->2	30042->1	70272->0
01772->1	11272->7	20232->1	30062->2	

Langton (1984)

Langton Loop

Black = space

Blue = information carrier medium

White = extend arm

Green = bend arm (two green data bend arm at 90 deg.)

Red = sheath which protects data

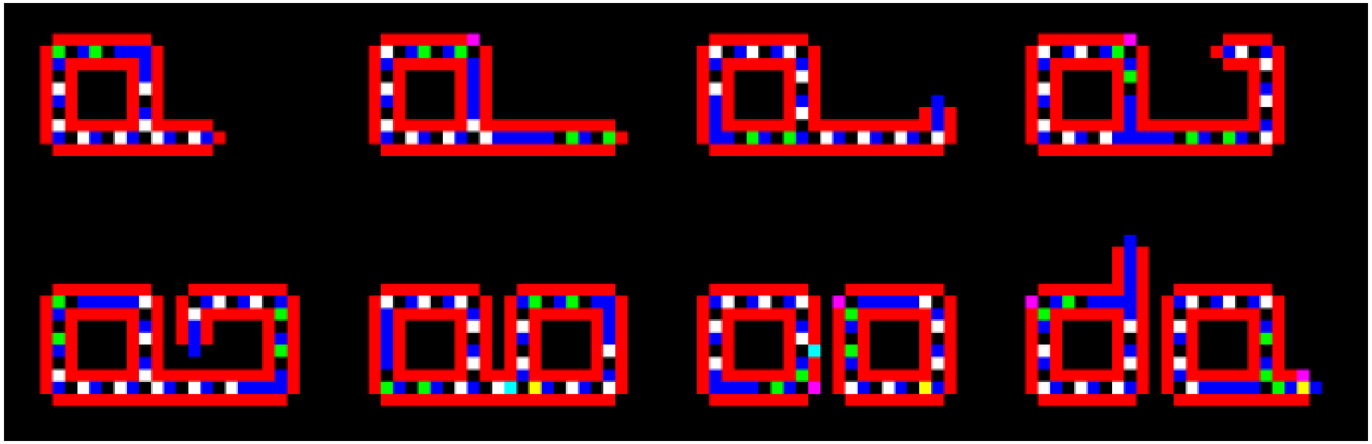


Figure 5: *Self-replication process of a Langton Loop. The parent loop extends an "arm" and bends it round to form a daughter loop. Then the umbilical dissolves to separate the loops. Now both loops are ready to self-replicate again.*

Langton Loop

Summary of features:

- Very simple structures compared to von Neumann and Codd's machine
- Not capable of universal construction
- Loop actively directs its reproduction employing transcription and translation (non-trivial replication)

Langton Loop

http://necsi.org/postdocs/sayama/sdsr/java/

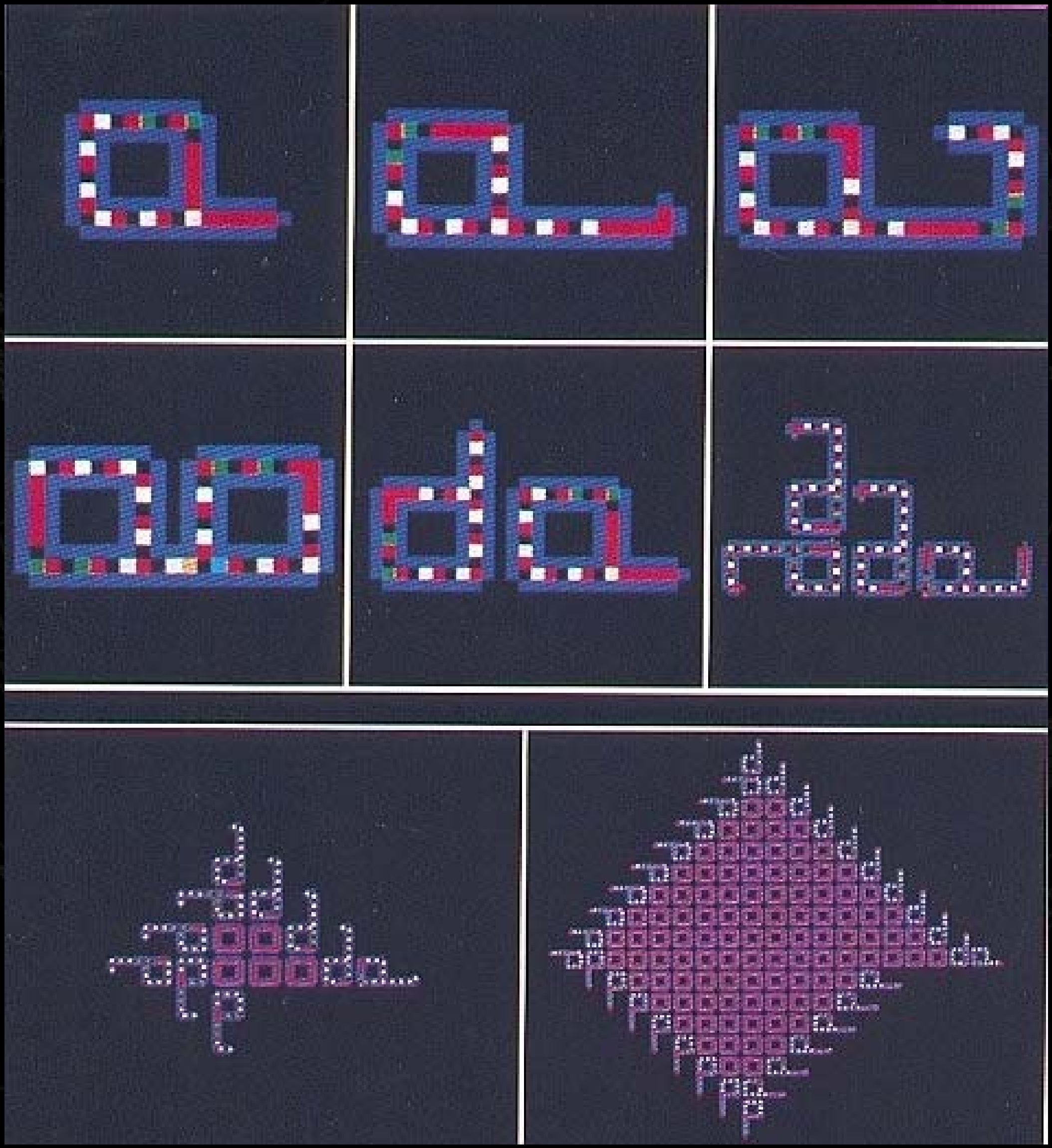
The Growth of a Loop Colony

Structure at multiple scales!

The loops of each generation behave differently

Developmental stages: birth, fetal stage, adolescence, adulthood, and death

Langton Loops



Langton Loop

Movie on DVD (31:30-)

Langton Loop - Problems

- Lack of robustness due to basic assumptions:
  - Presence of sheath (adds complexity)
  - “Perfect” single seed loop
  - Infinite CA with no obstacles
  - Presence of other structures in the universe or bounded universe lead to corruption of the loop and loss of “life”
  - Replication has a preferential direction

Unsheathed Self-Replicating Loops

Q: How can sheathed loops be simplified?

Such simplification is important for:

- Understanding the minimal information-processing requirements of self-replication
- Relating formal models to theories of the origins of life
- Identifying configurations so simple that they might actually be synthesized or fabricated

Unsheated Self-Replicating Loops

Simple Systems That Exhibit Self-Directed Replication

James A. Reggia,* Steven L. Armentrout, Hui-Hsien Chou, Yun Peng

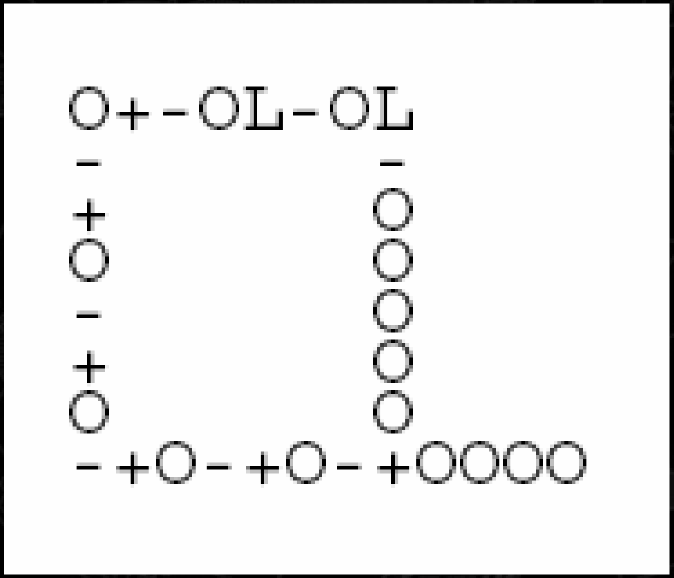
Biological experience and intuition suggest that self-replication is an inherently complex phenomenon, and early cellular automata models support that conception. More recently, simpler computational models of self-directed replication called sheathed loops have been developed. It is shown here that "unsheathing" these structures and altering certain assumptions about the symmetry of their components leads to a family of nontrivial self-replicating structures, some substantially smaller and simpler than those previously reported. The dependence of replication time and transition function complexity on initial structure size, cell state symmetry, and neighborhood are examined. These results support the view that self-replication is not an inherently complex phenomenon but rather an emergent property arising from local interactions in systems that can be much simpler than is generally believed.

Reggia et al. (1993)

Unsheathed Version of Langton Loop

"Only" 177 rules

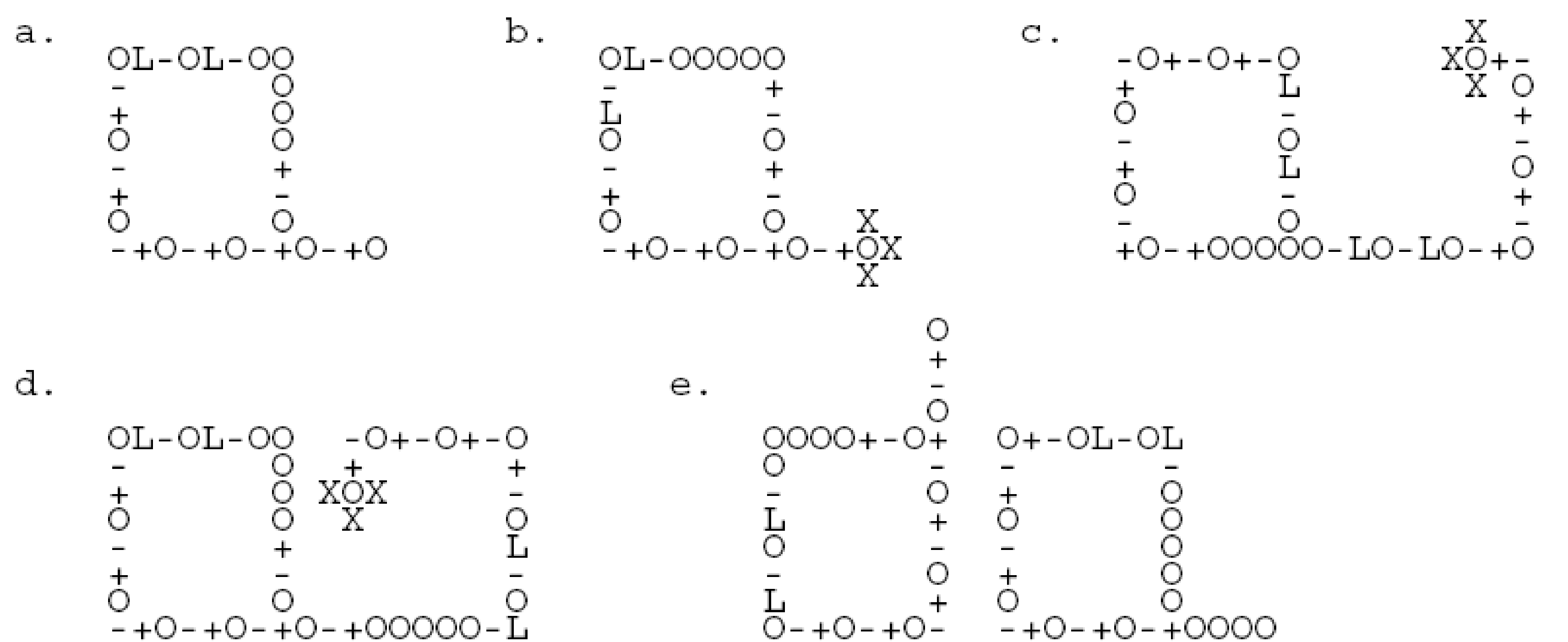
Less than 40% of the size of original sheathed loop (32 active cells vs. 86 active cells)



Reggia et al. (1992)

Unsheathed Self-Replicating Loops

- Program = "+-+-+-+-+-+-L-L-" (same as in Langton loop but with different instruction code: '+-' = extend, 'L-' = extend and turn left)
- Growth cap of X's at tip of arm enables: directional growth + right-left discrimination at growth site



Sayama SDSR Loop - Basic Info

- Extension of Langton’s loop to increase its robustness (Sayama, 1998)
- If the constructing arm hits another structure it is either absorbed or the obstacle is deleted
- The self-replicating structure can dissolve, i.e. it “decomposes” gracefully in order to leave space for other organisms
- SDSR Loop = Structurally Dissolvable Self-Replicating Loop
- 86 active 9-state cells

Sayama SDSR Loops

http://necsi.org/postdocs/sayama/sdsr/java/

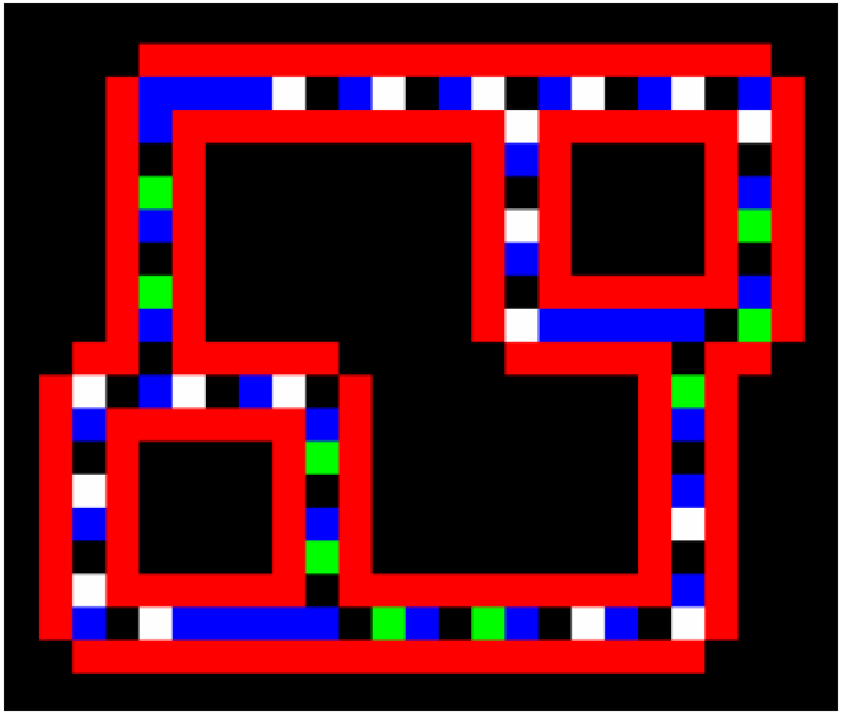


Figure 6: Stable configuration of two colliding SDSR Loops.

Programming Self-Replicating Loops

- Von Neumann: set of instructions (signals) on replicating structure’s tape that describe the structure can be viewed as the machine’s program
- Sequence of instructions that circulate around a self-replicating loop form a program that directs the loop’s replication

→ Only concerned with replication!

Intermediate Summary

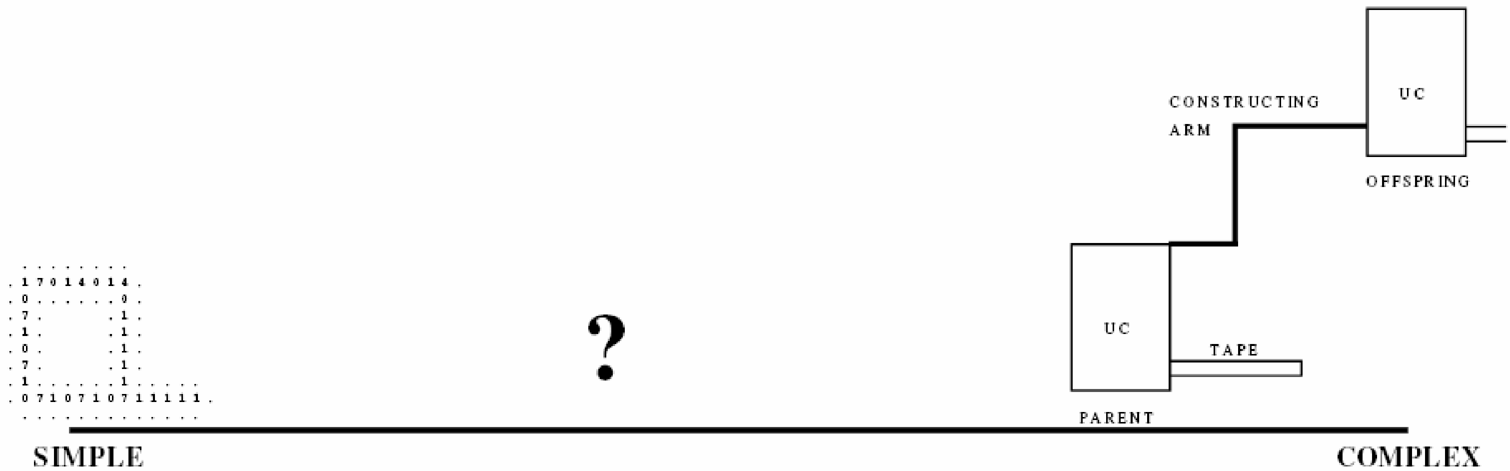


Figure 7. The complexity scale of self-replicating structures in cellular automata. On one end there are the highly complex universal constructor-computers, and on the opposite end one finds the simple structures that can do nothing but self-replicate. What kinds of structures can be devised in between the two?

from Sipper (1998)

Programming Self-Replicating Loops

- Signal sequences directing a structure's replication can be extended to solve a specific class of problems while replication occurs
- Programmed replicators provide a novel, massively parallel computational environment that may lead over the long term to powerful, very fast computing methods

Duplicated Program Sequences

- Extend sequence of signals circulating around loop by adding signals representing a program carrying out some task (application program)
- Application program is copied along with replication program and is executed once by each loop in between replications

Tempesti Loop

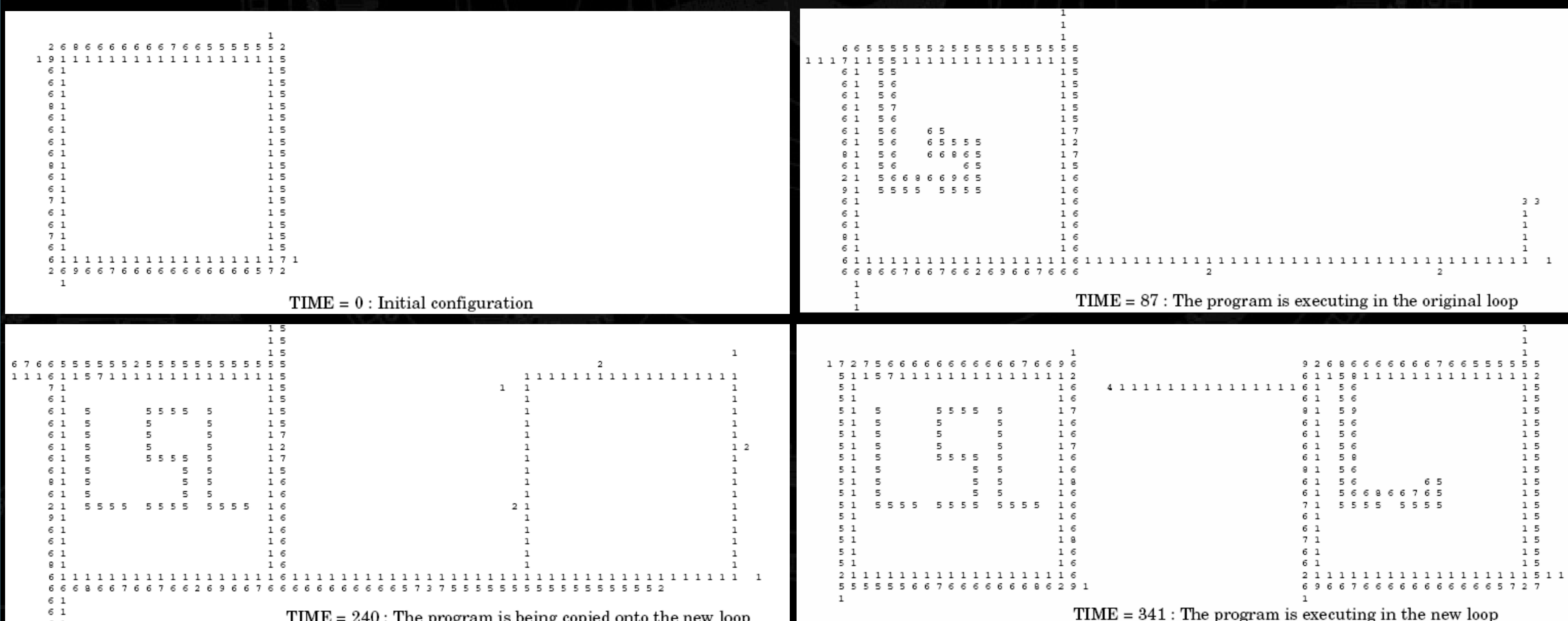
The whole point of artificial self-replication is to provide “service to mankind” from a small “seed”

- Structures need to be able to replicate but need also be capable of additional function
- Tempesti loops can self-replicate and write the letters ‘LSL” inside themselves (Tempesti, 1995)
- > 6 states
- 692 = 4*173 transition rules
- 8-cell neighborhood (Moore)
- Time for one replication = 321 steps

Tempesti Loop

Loops replicate in two stages:

1) Copy main loop
2) Daughter loop sends signal back to parent loop, parent loop sends instruction for writing letters



TIME = 0 : Initial configuration

TIME = 87 : The program is executing in the original loop

TIME = 240 : The program is being copied onto the new loop

TIME = 341 : The program is executing in the new loop

Tempesti (1995)

Tempesti Loop

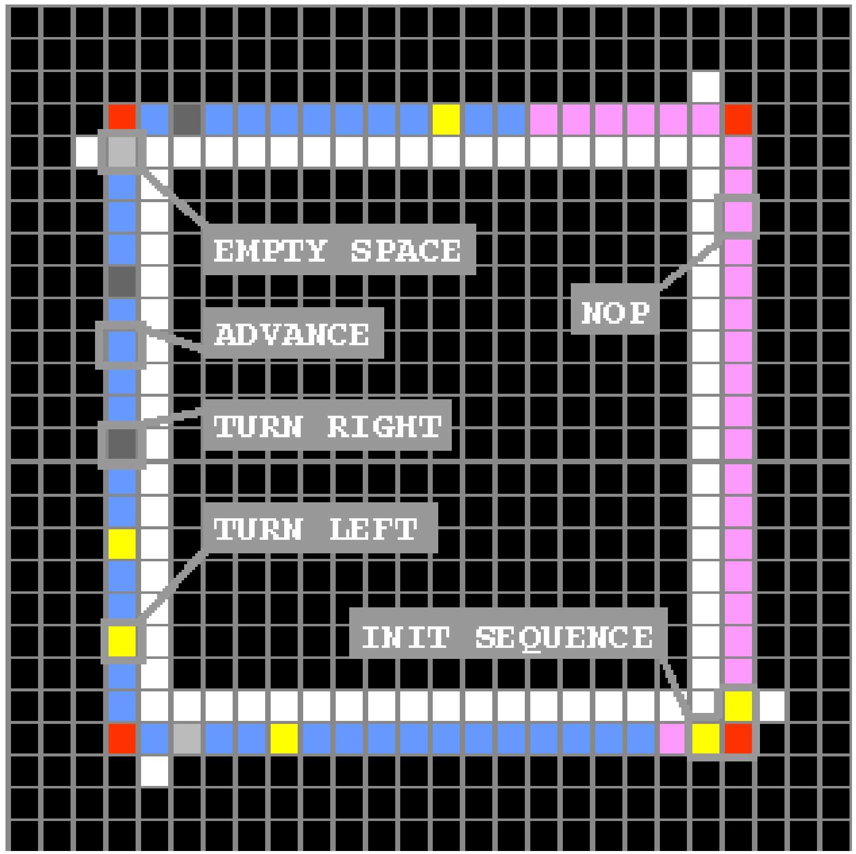


Figure 3-23: Configuration of the LSL automaton at iteration 0.

http://lslwww.epfl.ch/pages/embryonics/thesis/home.html

Tempesti Loop

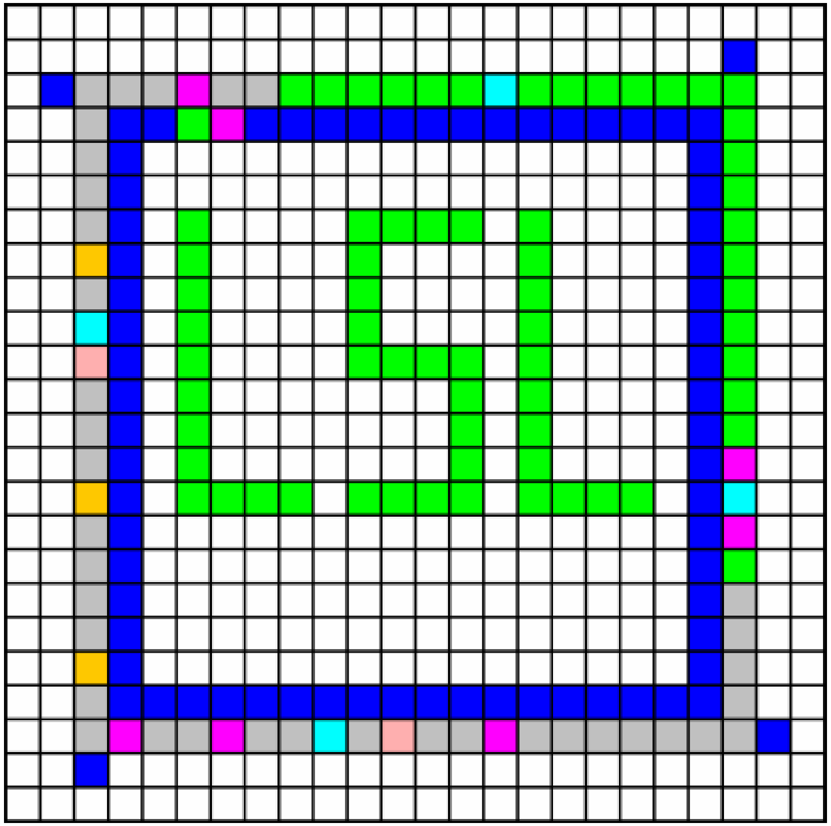


Figure 7: *A Tempesti loop after it has spawned four children and constructed the letters "LSL." Both the information for self-replication and that for building the letters flow around the outside of the loop. Notice the concept of the sheath is still here but only the inner sheath. The rules are robust enough for the data to be subjected to the "void." The inner sheath merely acts as a guide for the data flow, not protection.*

Tempesti Loop

Movies:

Loop_Basic_Small.mov

Loop_LSL_Big.mov

Expanding Problem Solutions

Programmable self-replicating loops encode a set of instructions on the loop or an attached tape that directs solution of a problem

More general approach: initial problems solution is not copied exactly from parent to child but is modified from generation to generation; each loop get different partial problem solution

If a loop determines it has found a valid complete problem solution, it stops replicating and retains that solution as a circulating pattern in its loop

SRL as Universal Computer

Physica D, 97:335-352, 1996.

Toward a Viable, Self-Reproducing Universal Computer

Jean-Yves Perrier Moshe Sipper Jacques Zahnd
Logic Systems Laboratory
Swiss Federal Institute of Technology
IN-Ecublens, CH-1015 Lausanne, Switzerland
e-mail: {Jean-Yves.Perrier, Moshe.Sipper, Jacques.Zahnd}@di.epfl.ch

SRL as Universal Computer

```
  . . . . . . . .
. 7 0 1 7 0 1 7 0 .
. 1 . . . . . . 1 .
. 1 .         . 7 .
. 1 .         . 0 .
. 1 .         . 1 .
. 1 .         . 7 .
. 0 . . . . . . 0 . . . .
. 4 1 0 4 1 0 7 1 0 7 1 1 .
. A . . . . . . . . . . .
. P .         . D .
. P .         . D .
. P .         . D .
. P .         . D .
. P .           .
. P .
. P .
  .
```

Tape 1

Tape 2

Figure 4: The automaton structure. P denotes a state belonging to the set of program states, D denotes a state belonging to the set of data states, and A is a state which indicates the position of the program.

Perrier et al., (1996)

SRL as Universal Computer

ELSEVIER

Physica D 115 (1998) 293–312

PHYSICA D

Problem solving during artificial selection of self-replicating loops ★

Hui-Hsien Chou [a,*], James A. Reggia [b,1]

[a] *The Institute for Genomic Research, 9712 Medical Center Drive, Rockville, MD 20850, USA*

[b] *Department of Computer Science and Institute for Advanced Computer Studies, A.V. Williams Bldg., University of Maryland, College Park, MD 20742, USA*

Received 21 July 1997; received in revised form 5 September 1997; accepted 24 September 1997

Communicated by A.M. Albano

Abstract

Past cellular automata models of self-replication have generally done only one thing: replicate themselves. However, it has recently been demonstrated that such self-replicating structures can be programmed to also carry out a task during the replication process. Past models of this sort have been limited in that the "program" involved is copied unchanged from parent to child, so that each generation of replicants is executing exactly the same program on exactly the same data. Here we take a different approach in which each replicant receives a distinct partial solution that is modified during replication. Under artificial selection, replicants with promising solutions proliferate while those with failed solutions are lost. We show that this approach can be applied successfully to solve an NP-complete problem, the satisfiability problem. Bounds are given on the cellular space size and time needed to solve a given problem, and simulations demonstrate that this approach works effectively. These and other recent results raise the possibility of evolving self-replicating structures that have a simulated metabolism or that carry out useful tasks.

Keywords: Self-replication; Self-organization; Cellular automata; SAT problem; Artificial selection

Summary: CA-Based Self-Replicators

1960s-1970s: large, complex universal systems modeled after Turing machines

Mid 1980s: criteria of universal computation and construction were relaxed; models characterized by self-replicating loops

Mid 1990s – now: focus on emergence of self-replicating structures; rules to control self-replication are discovered by artificial evolution

Self-Replicating Loops: Future Work

- A high-level programming language that specifically supports development of CA transition rules would be of great value
- Hardware that directly supports the massively parallel but local computations of CA modeling
- Further development of programmable self-replicators
- **Interesting conclusion:** artificial self-replicators show that self-replication of information-carrying structures can be far simpler than one might think

Complexity of Self-Replicating Systems (Bits)

- Von Neumann's universal constructor ~500,000
- Internet worm (Robert Morris, Jr., 1988) ~500,000
- Mycoplasma genitalium 1,160,140
- E. Coli 9,278,442
- Drexler's assembler ~100,000,000
- Human ~6,400,000,000
- NASA Lunar Manufacturing Facility > 100,000,000,000

Self-Replication in Physical Systems

- First achieved by L. Penrose and R. Penrose (1958)
- System consists of a number of identical units with particular shape and mechanical motions
- If many units are put into a box and the box is shaken nothing happens ("no life expect from life")
- If a seed is placed in the box, however, the seed ("self") can use the other units ("food") to replicate (self-replication)
- In 2-D using tumbling wooden tiles (tilt blocks built out of plywood) and a machine comprising four different components that are assembled following tracks
- Blocks are subjected to horizontal agitation (energy necessary to engage units into process of self-replication)

The Penrose Block Replicator - Type 1

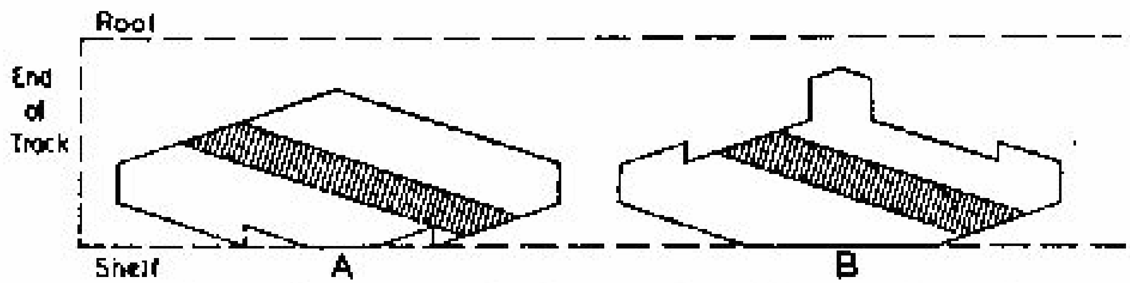


Fig. 1. Elements A and B in neutral positions on the track

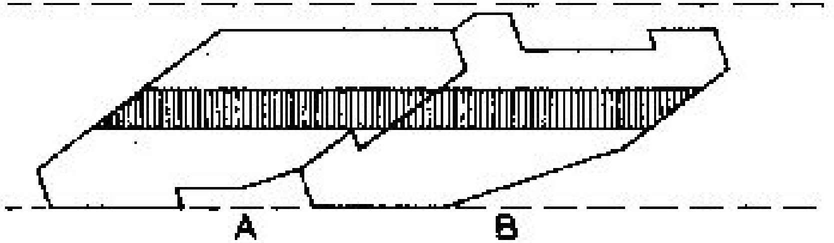


Fig. 2. Elements A and B hooked together

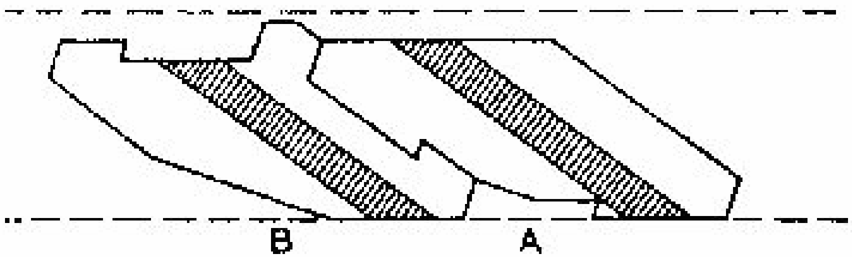


Fig. 3. Alternative complex of B and A.

Penrose (1959)

The Penrose Block Replicator - Type 1

Collisions between the two-block unit and other lone parts in the box cause new two-block units to form, each identical to the original, demonstrating self-replication as a simple form of mechanical autocatalysis

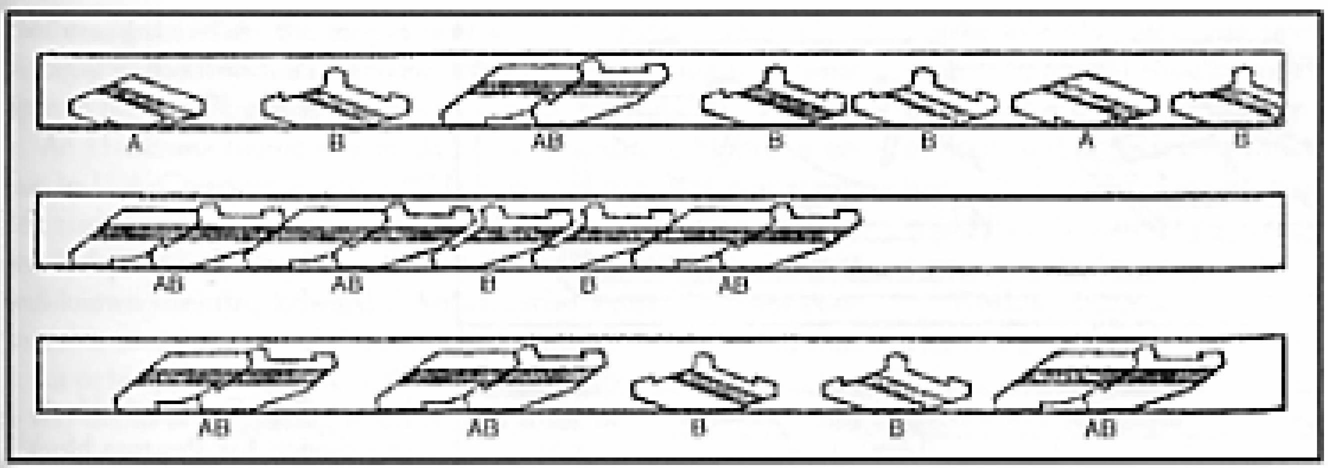


Figure 3.3. A 1-D self-replicating machine made of parts of two kinds, from Penrose.[681]

Penrose (1959)

The Penrose Block Replicator - Type 2

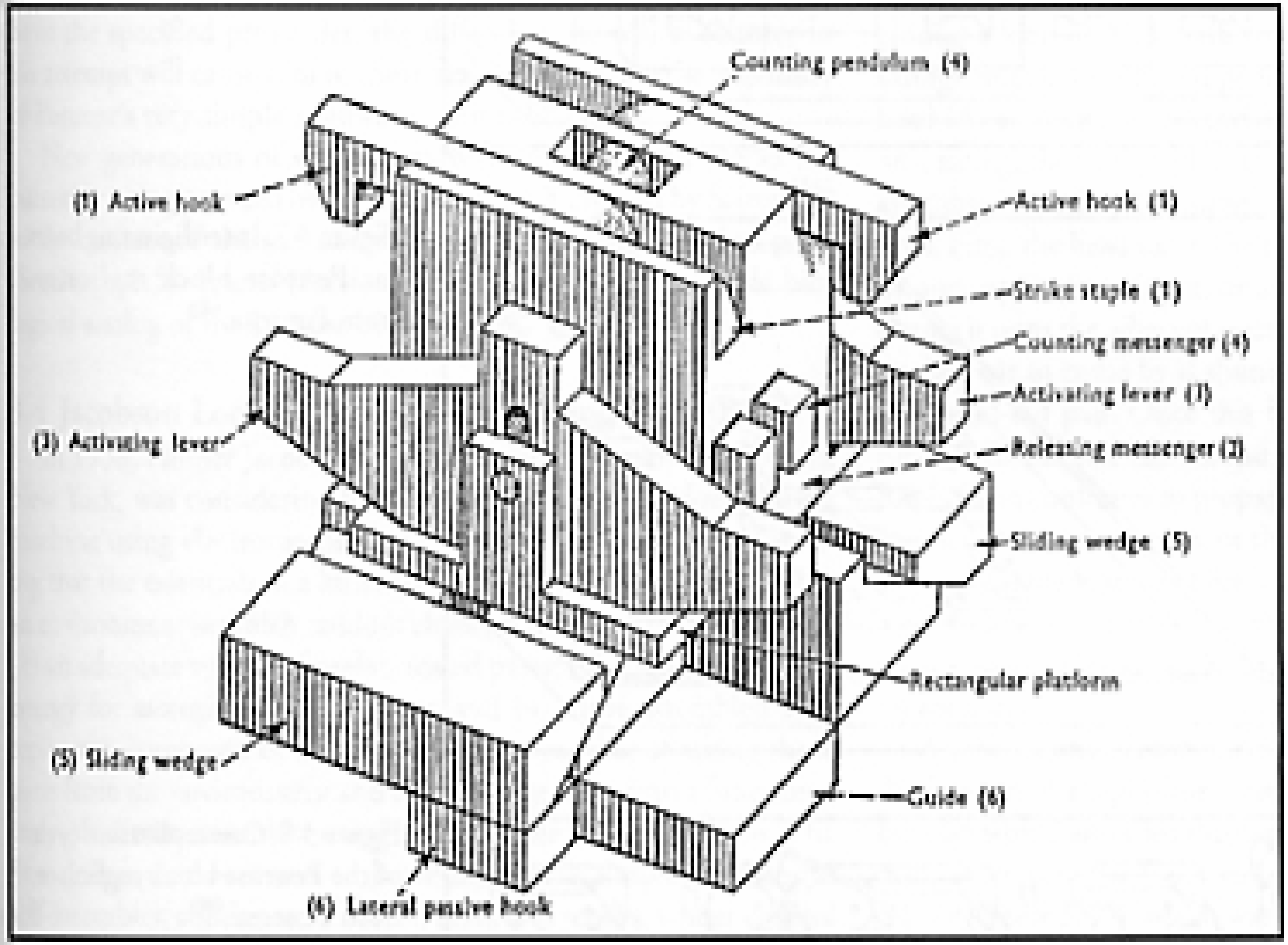


Figure 3.4. A double-hook "food" unit for Penrose block replicator, from Penrose.[68]

Penrose (1959)

The Penrose Block Replicator - Type 2

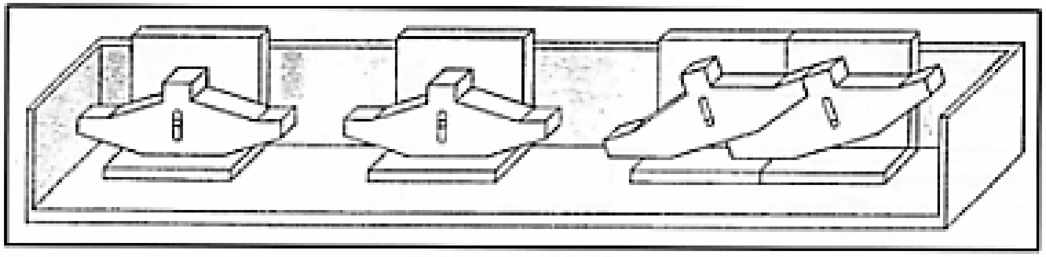


Figure 3.5. Activating cam levers for Penrose block replicator, from Penrose.[683]

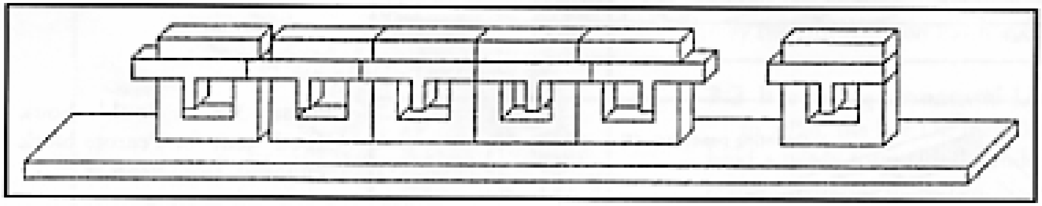


Figure 3.6. Four-unit blocking device for Penrose block replicator, from Penrose.[683]

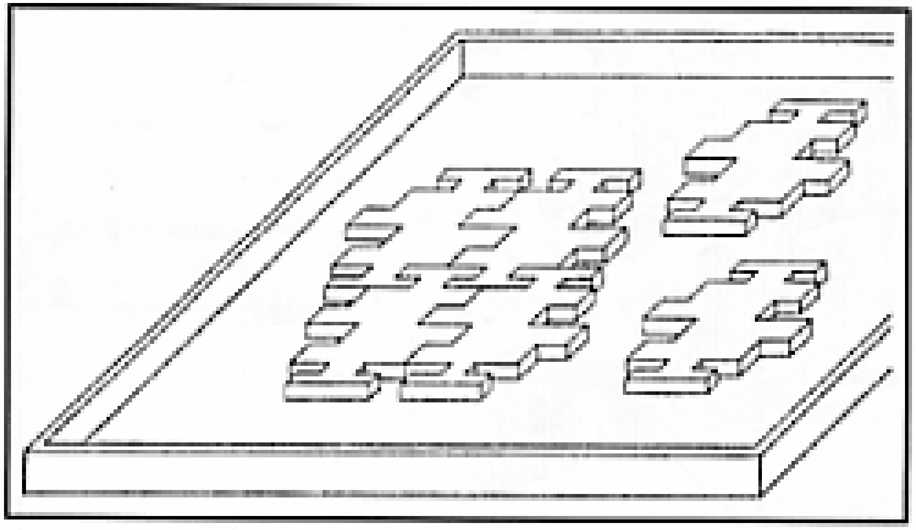


Figure 3.7. Interdigitating bases for Penrose block replicator, from Penrose.[683]

The Penrose Block Replicator

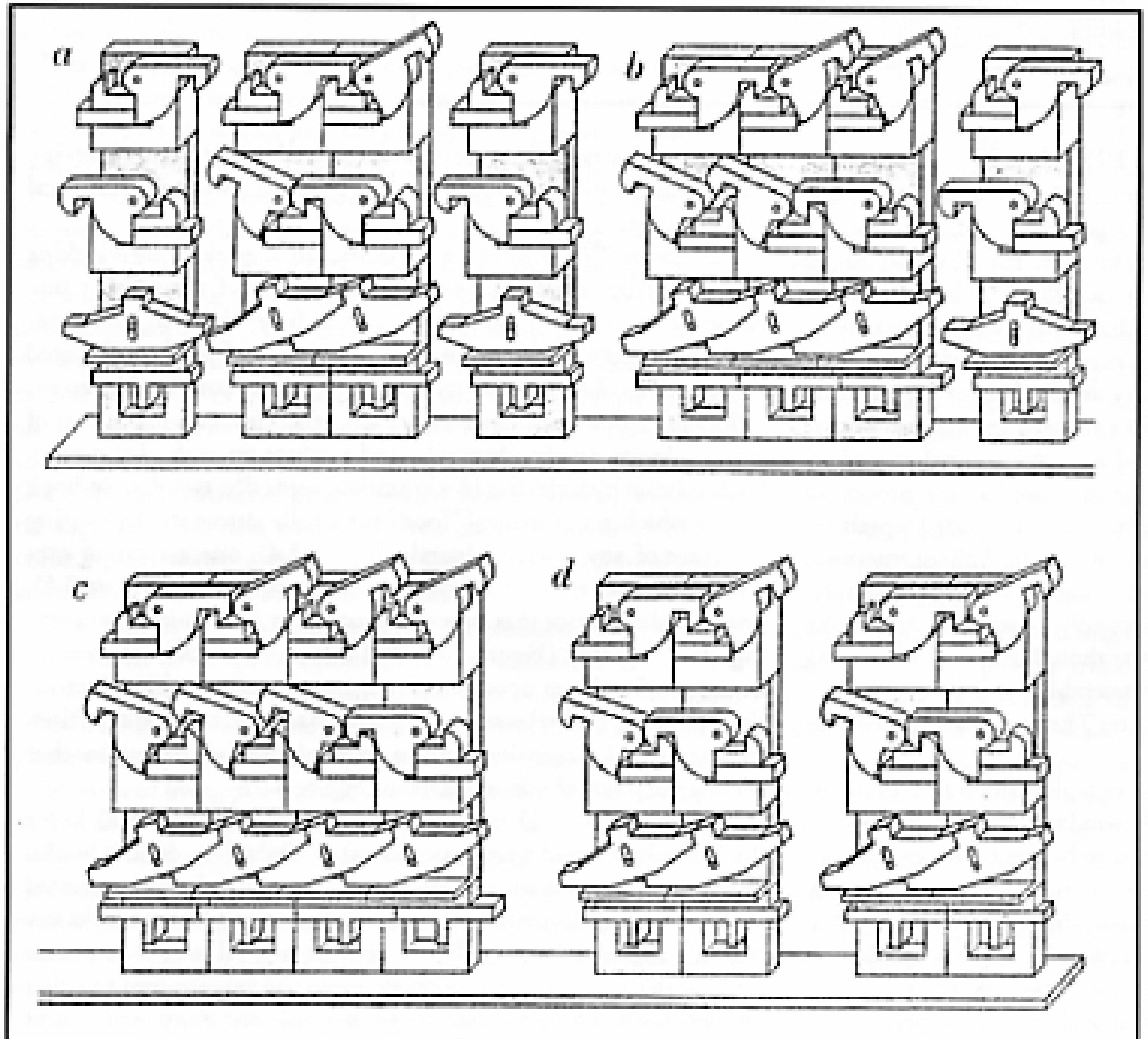


Figure 3.8. One replication cycle of the Penrose block replicator, from Penrose.[683]

The Penrose Block Replicator

Movie:

Automatic mechanical self-replication

Modern Version of Mechanical Replicator

Movie: mechrep_cd.mov

Griffith et al. (2004)

http://alumni.media.mit.edu/~saul/PhD/

Self-Replication in the Real World

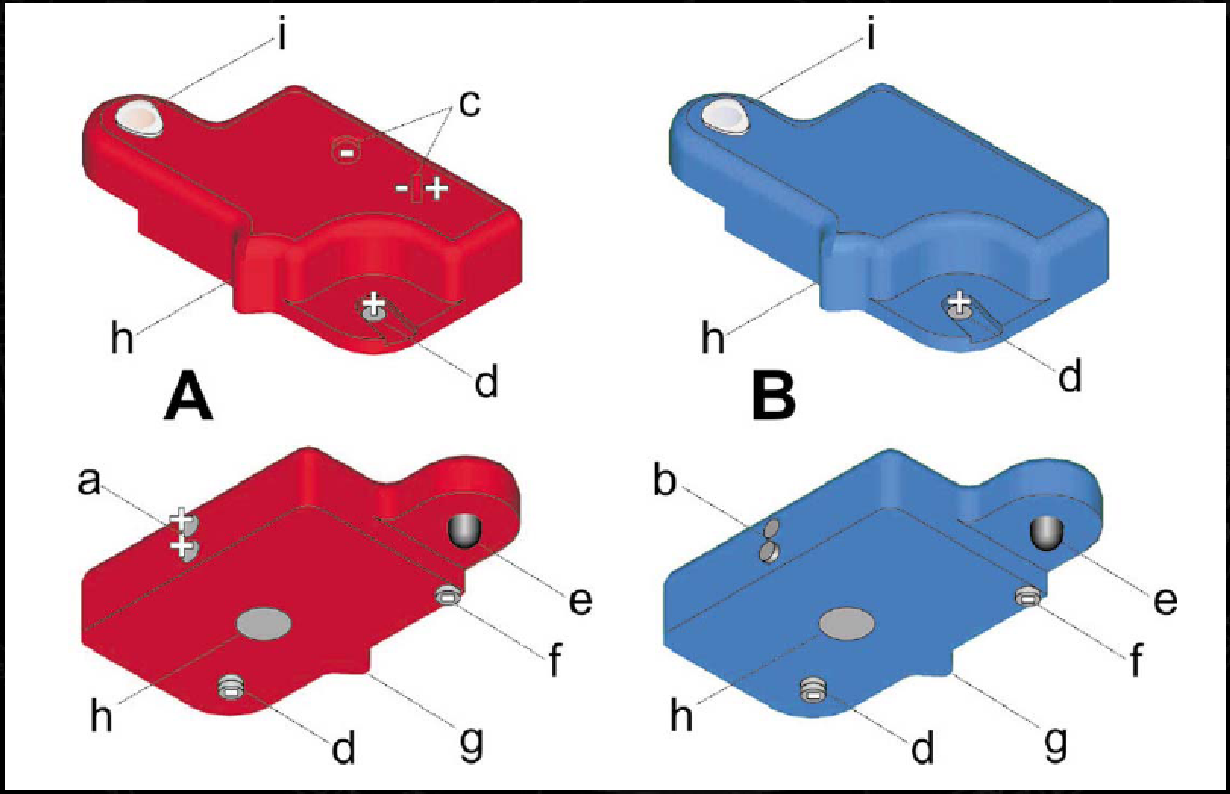
Self-Organization of Selective Replication

Self-organization of template-replicating polymers due to environmental fluctuations in temperature

Main idea: Initially random sequences of monomers direct the formation of complementary sequences, and structural information is inherited from one structure to the other (inspired by the Penrose block-replicator)

Breivik (2001)

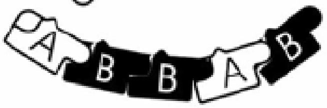
Monomeric Objects



Breivik (2001)

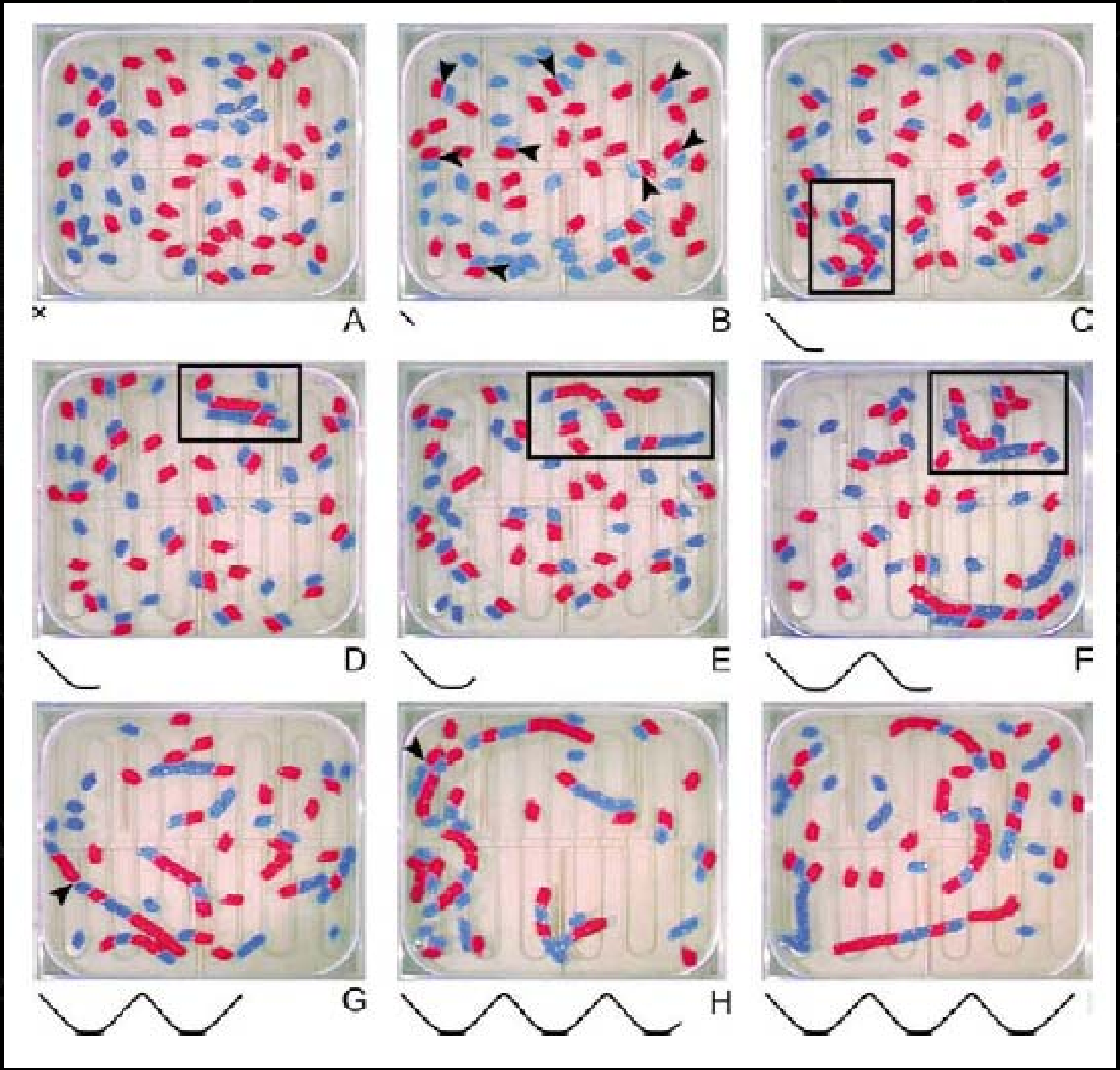
Self-Organization of Selective Replication

Self-organization of template-replicating polymers

	General concept:	Biologic correlate:	Presented system:
a	At least two types of objects (A, B) that form two types of bindings (I and II).	The 4 nucleotides of DNA (A, G, T, C).	
b	Binding I forms specific pairs (A:B) in response to a cyclic variable.	Watson-Crick base pairs (A:T, G:C) in response to the cell cycle.	 in response to temperature cycles.
c	Binding II forms continuous polymers (-A-B-B-A-B-).	Phosphodiester bridges of the DNA backbone.	
d	Binding I more probable than Binding II.	Low vs. high activation energy.	Exposed vs. concealed surfaces.
e	Binding II more stable than Binding I.	Covalent vs. hydrogen bindings.	High T_c vs. low T_c.
f	**IF [A_1-B_1 AND A_1:B_2 AND B_1:A_2] THEN A_2-B_2**	Protein dependent DNA replication.	

Breivik (2001)

Self-Organization Through Thermocycles



Breivik (2001)

Self-Replicating Machines

Modular robots (composed of modules or cells)

Self-reconfigurable robots (capable of changing shape and form)

Self-repairing machines

Self-Replicating Machines

Constructs a 3-module replica in 1 minute; a 4-module replica in 2.5 minutes

Zykov et al. (2005) “Self-reproducing machines” Nature 435

Self-Replication from Random Inputs

Goal: autonomous self-replication of a reconfigurable string of parts from randomly positioned input components

Griffith et al. (2005): Self-replication of 5-bit string

Parts float on 2-D air table

Self-Replication from Random Inputs

Movie:

"re5mer"

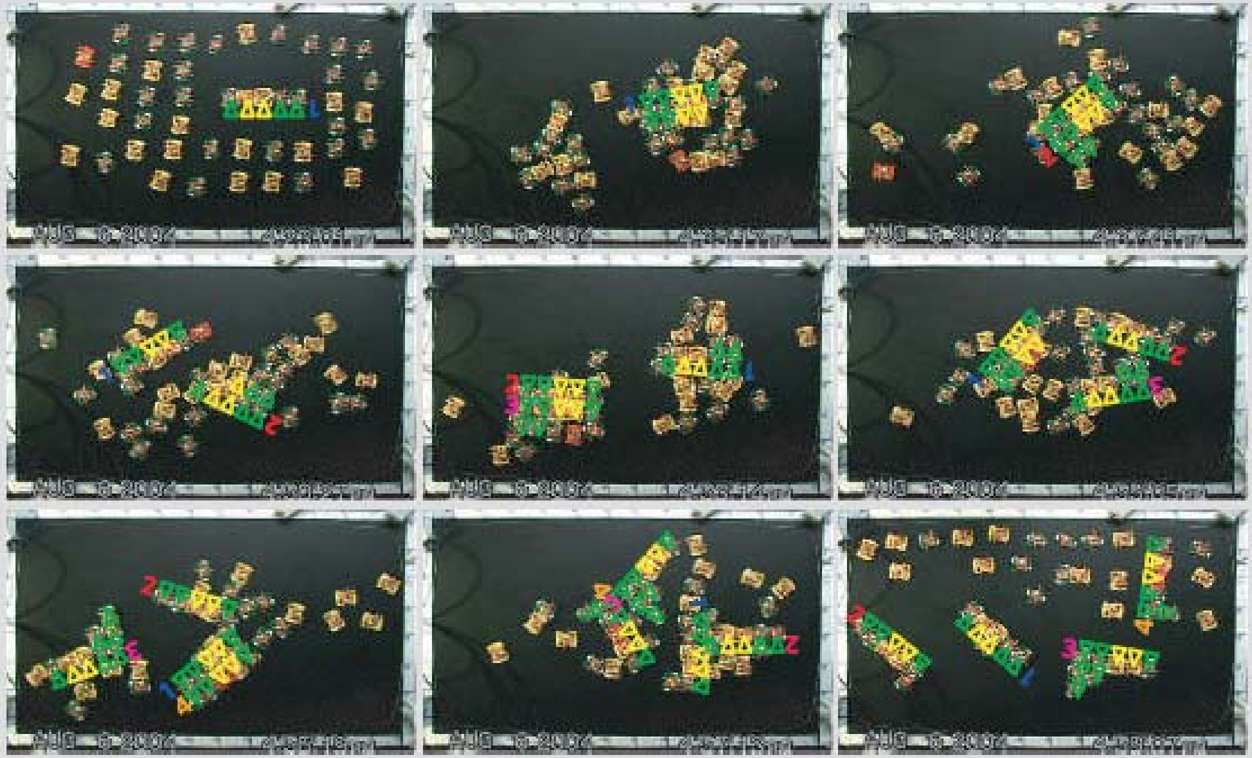


Figure 1 | Self-replication of a 5-bit string. Frames 1-9 (from left to right in each row): time sequence of photographs showing the autonomous replication of a 5-bit string of electromechanical units, starting from a single initial input string (number 1: green, green, yellow, yellow, green); frame 3: multiple replicants (numbered 1,2) assembling on a single substrate. Addition of building blocks is purely sequential along the string, as governed by a rule running in each block's state machine; frame 9: four independent strings (numbered 1-4) result from the action of templating and division. For movie, see supplementary information.

Griffith et al. (2005)