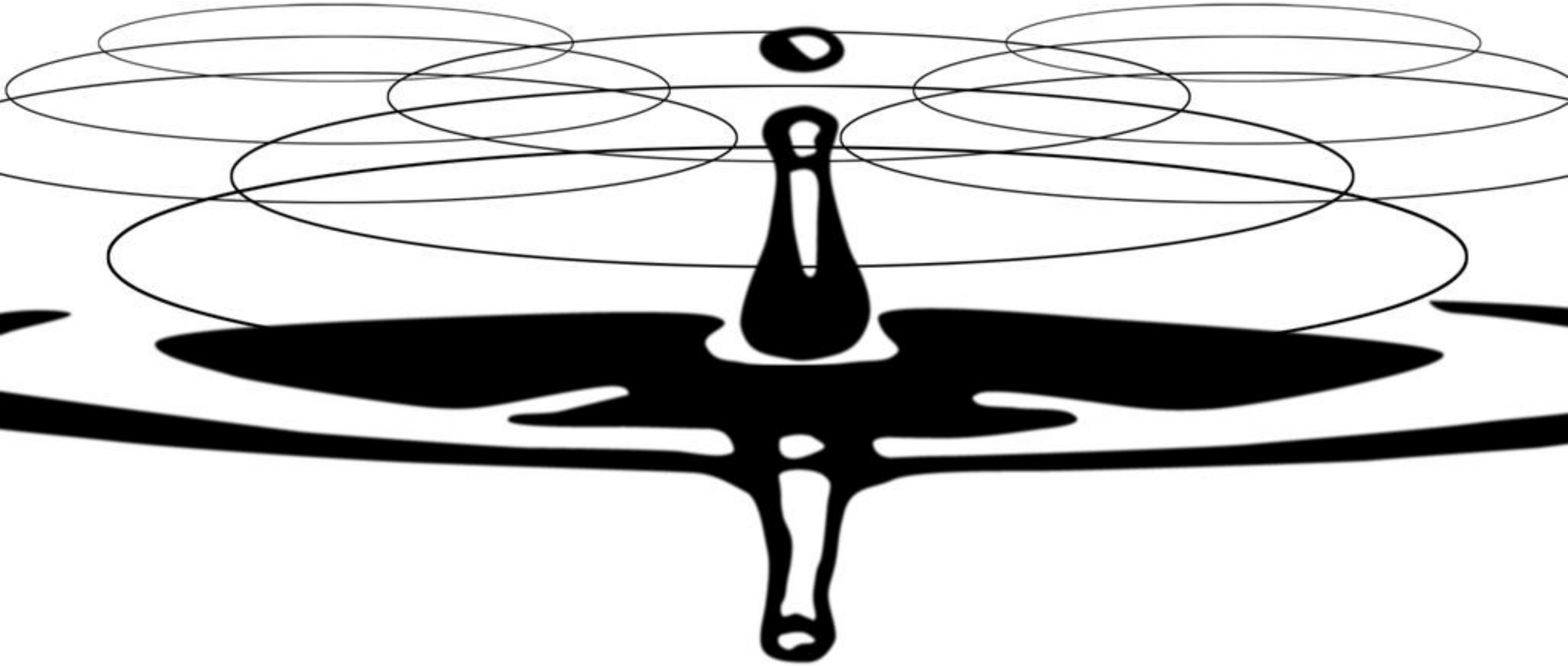
From Digitalism to Observer

Jake
School of Systems Science



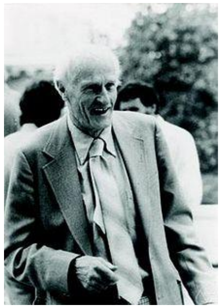
Outline

- Digitalism
- Wolfram’s NKS
- Tierra liked simulations
- Observer
- Clickstream

Triumph of Computation Theory

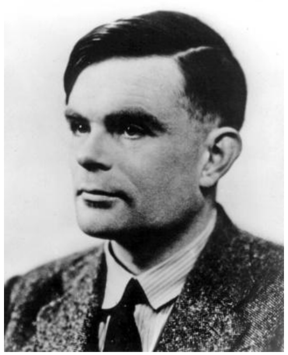
David Hilbert
（ 1862-1943 ）

Alonzo Church(1903-1995)

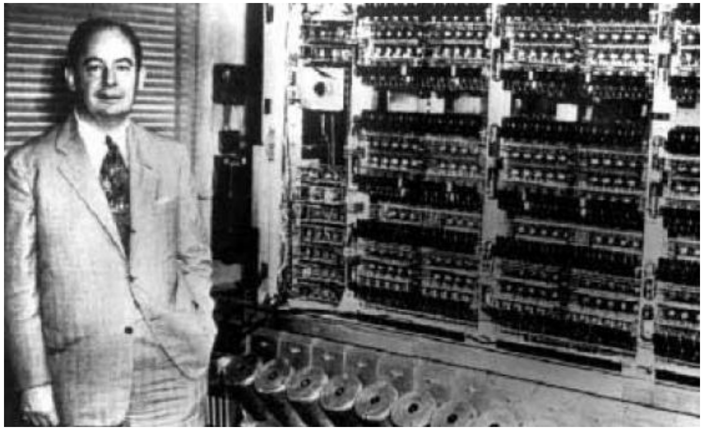


Stephen Kleene (1909-1994)

Kurt Godel(1906-1978)

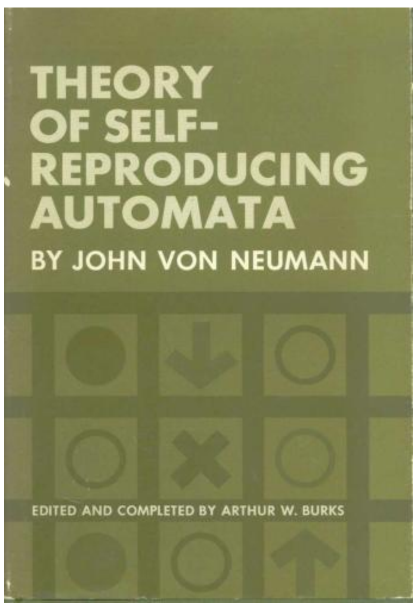


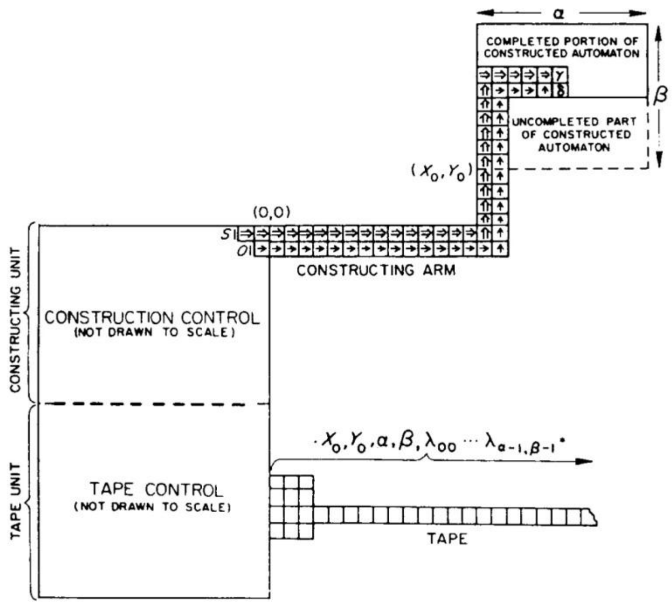
Alan Turing, 1912-1954

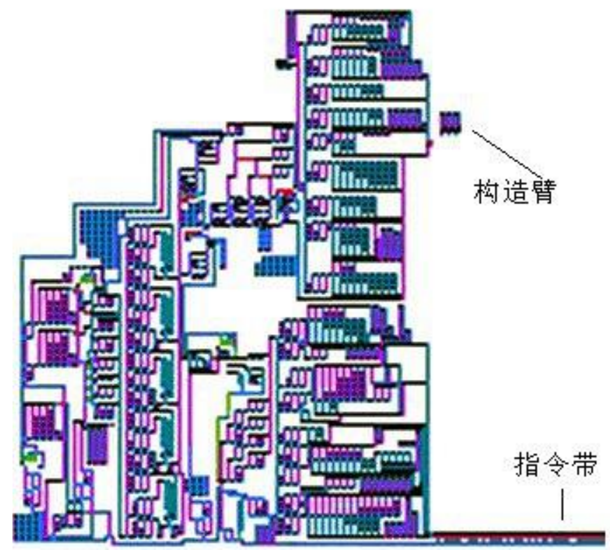


Von Neumann, 1903-1957

Self-reproducing Automata







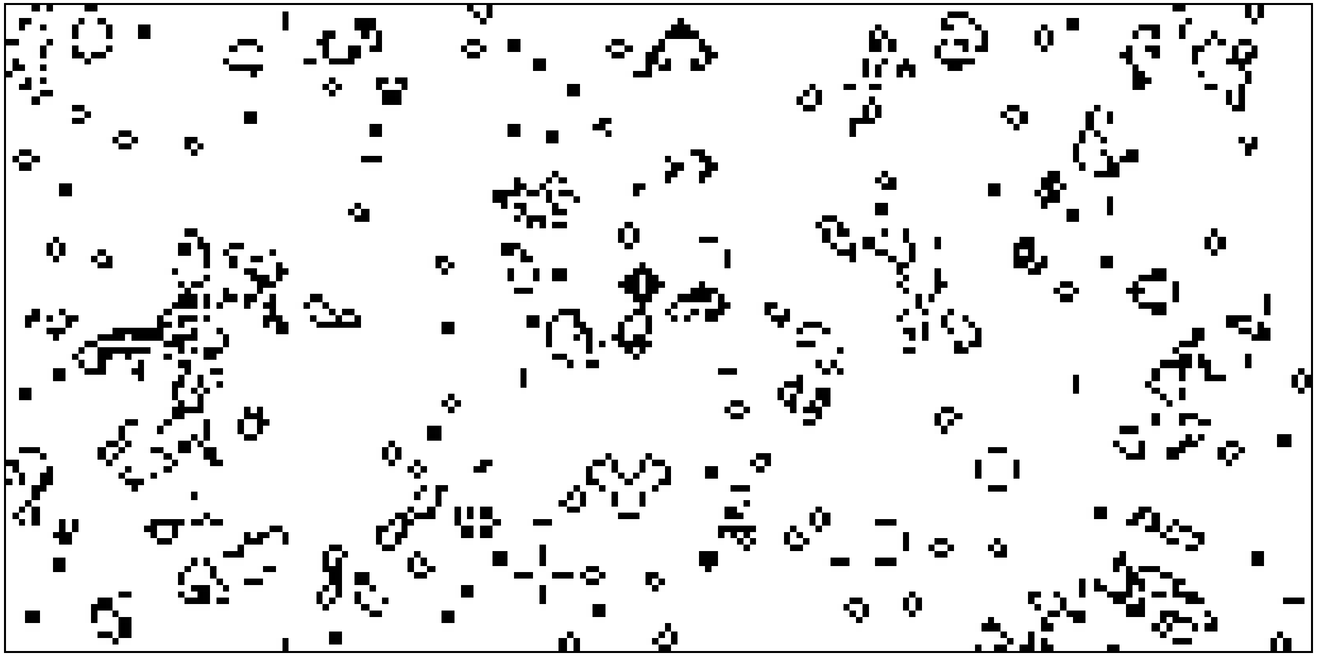
Self-reference

这句话是错的

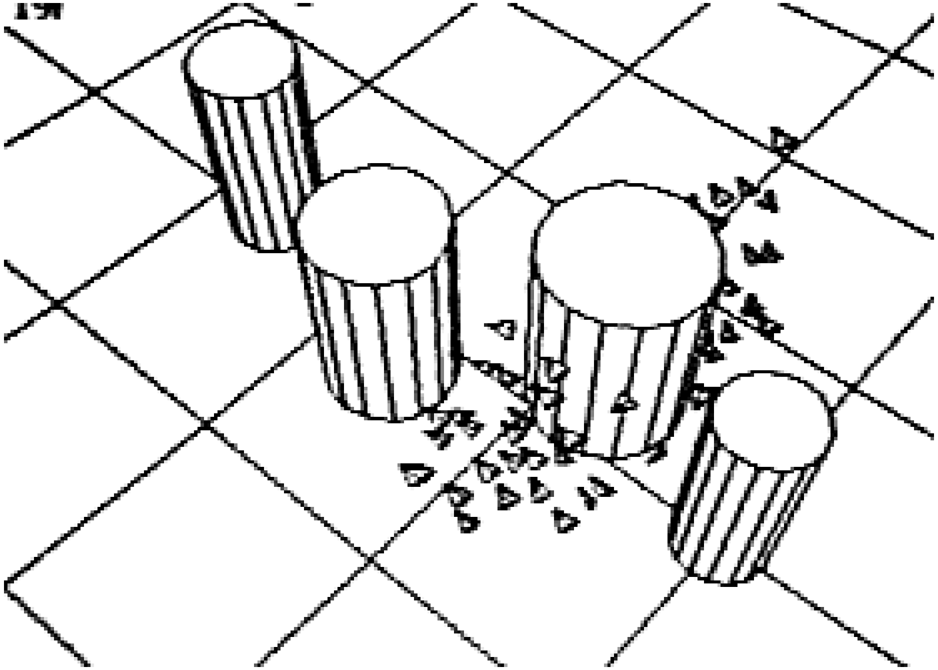
这句话是对的

自颠覆的哥德尔语句，
自创生的冯纽曼自动机

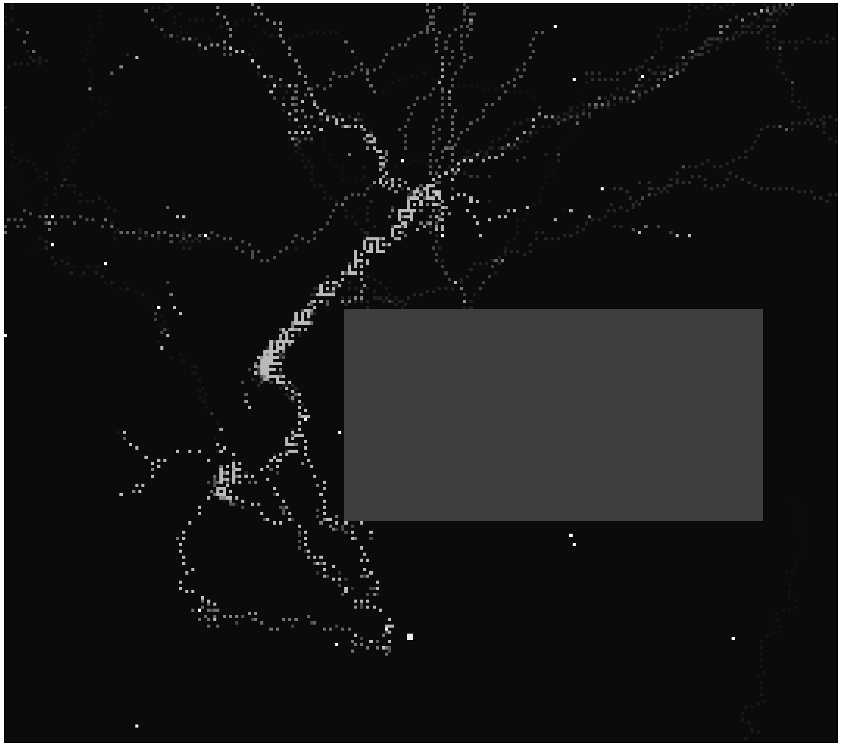
Studies of Simple Programs

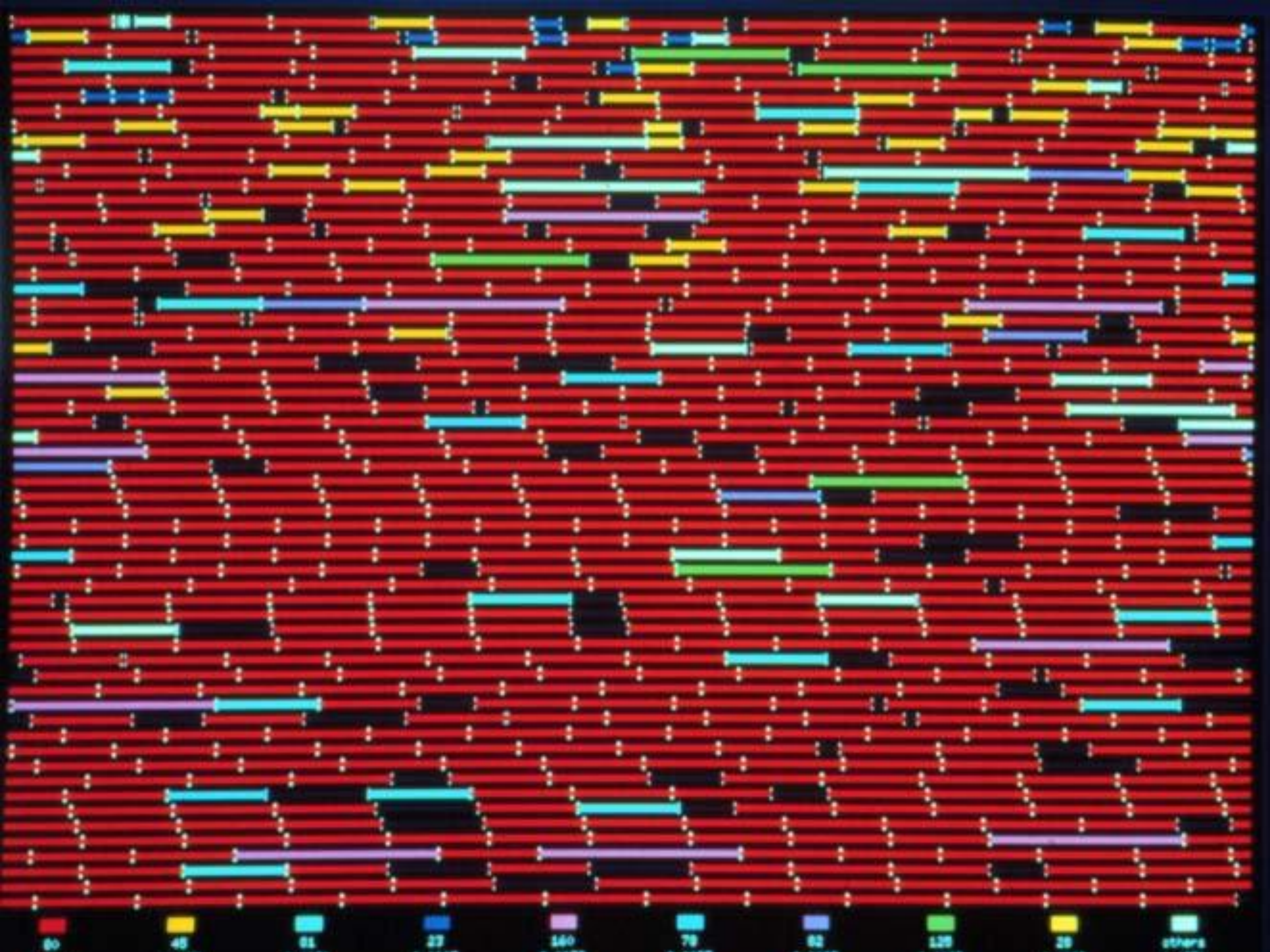


Studies of Simple Programs



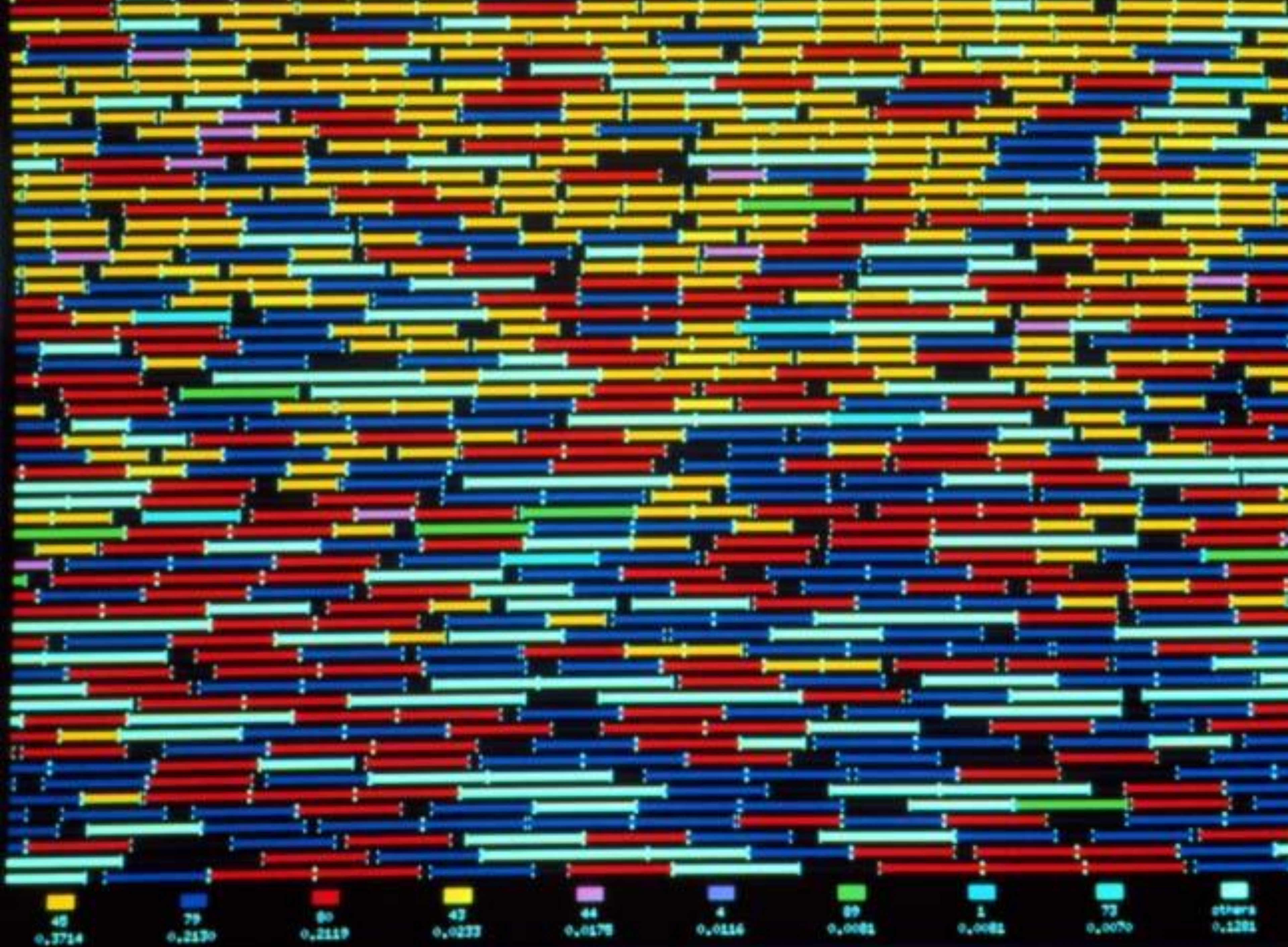
Studies of Simple Programs





27
160
125
others


45
80
79
58
81
46
106
79
43
others

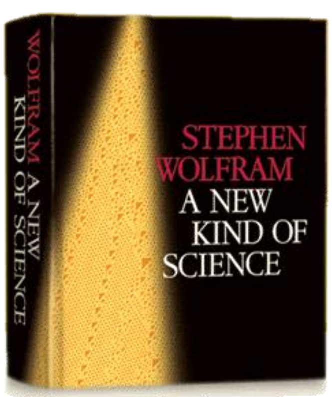


45
0,3714
79
0,2130
80
0,2119
47
0,0233
44
0,0175
4
0,0116
89
0,0081
1
0,0081
72
0,0070
others
0,1281

Artificial life & Open Ended Evolution

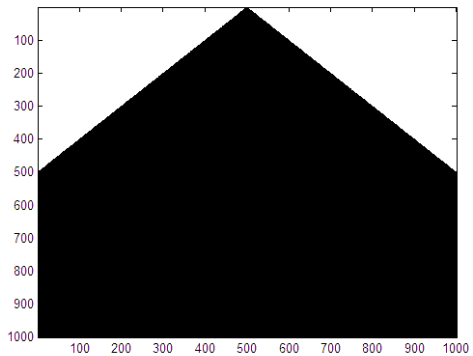
- Many Tierra-like systems
- All of them are not open-ended
- There seems a law like 2nd thermodynamic law

Wolfram's NKS

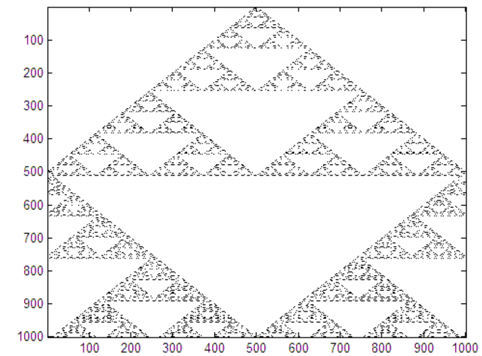


- Published his first paper in 15 years old, the youngest recipient of MacArthur Prize Fellowship (22 years old)
- Worked for Princeton, Illinois university
- Launched Wolfram Research Inc. in 1986
- Shifted from physics to complexity, began to study CA in mid 1980's
- Started to write NKS book from 1991
- Published the book in May, 2002

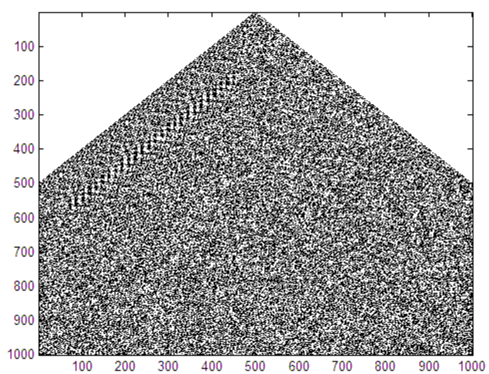
Wolfram's NKS



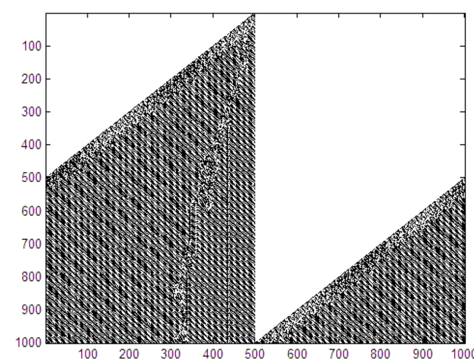
固定值：254



周期结构：90



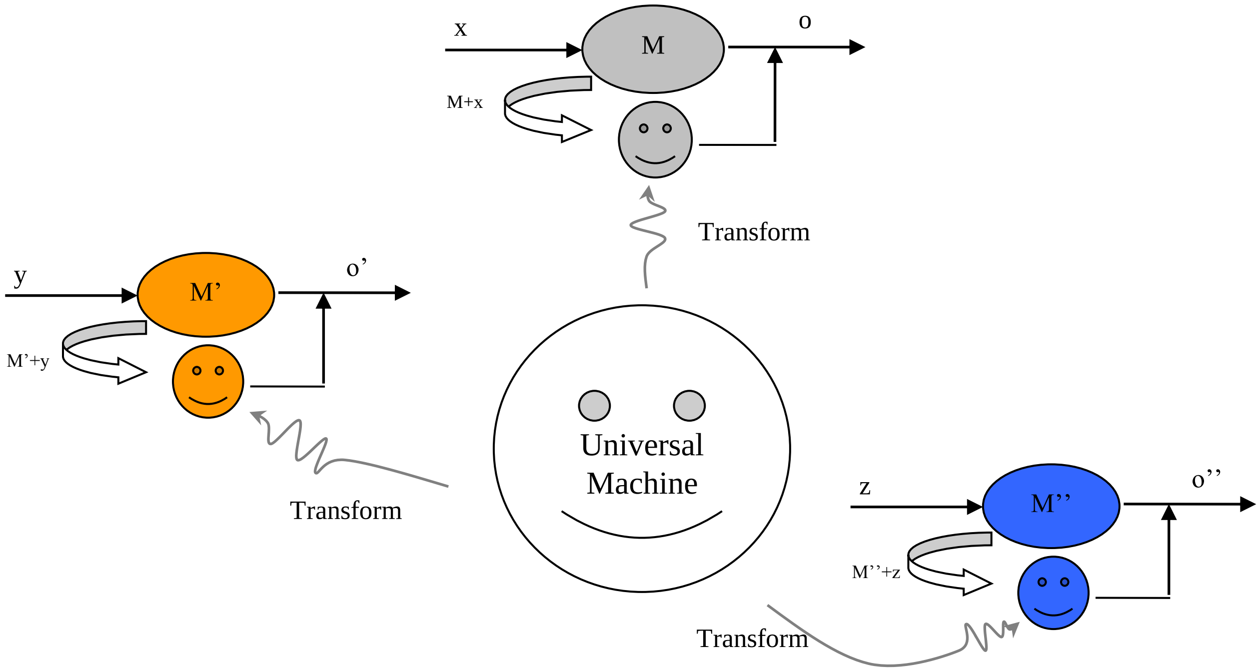
随机：30



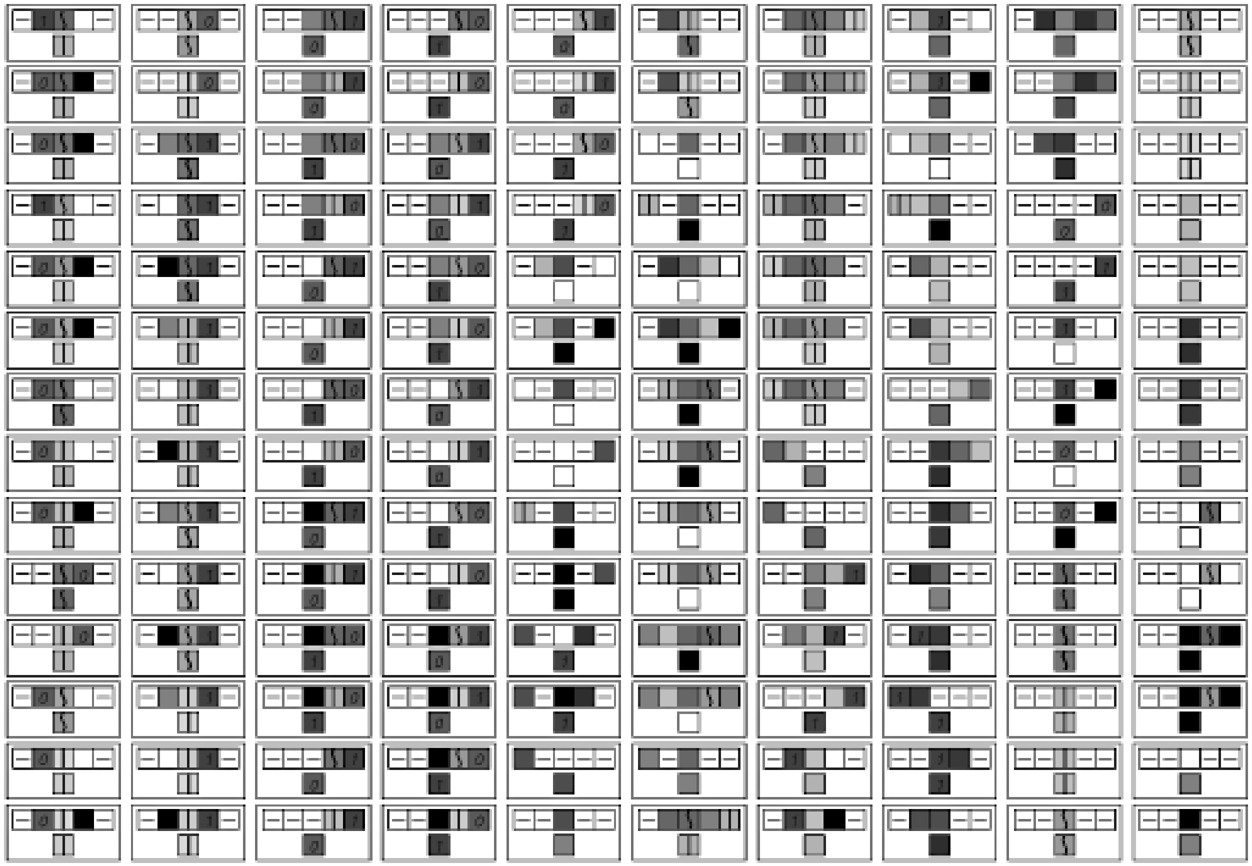
复杂：110
（局域的不规则结构）

Universal Machine

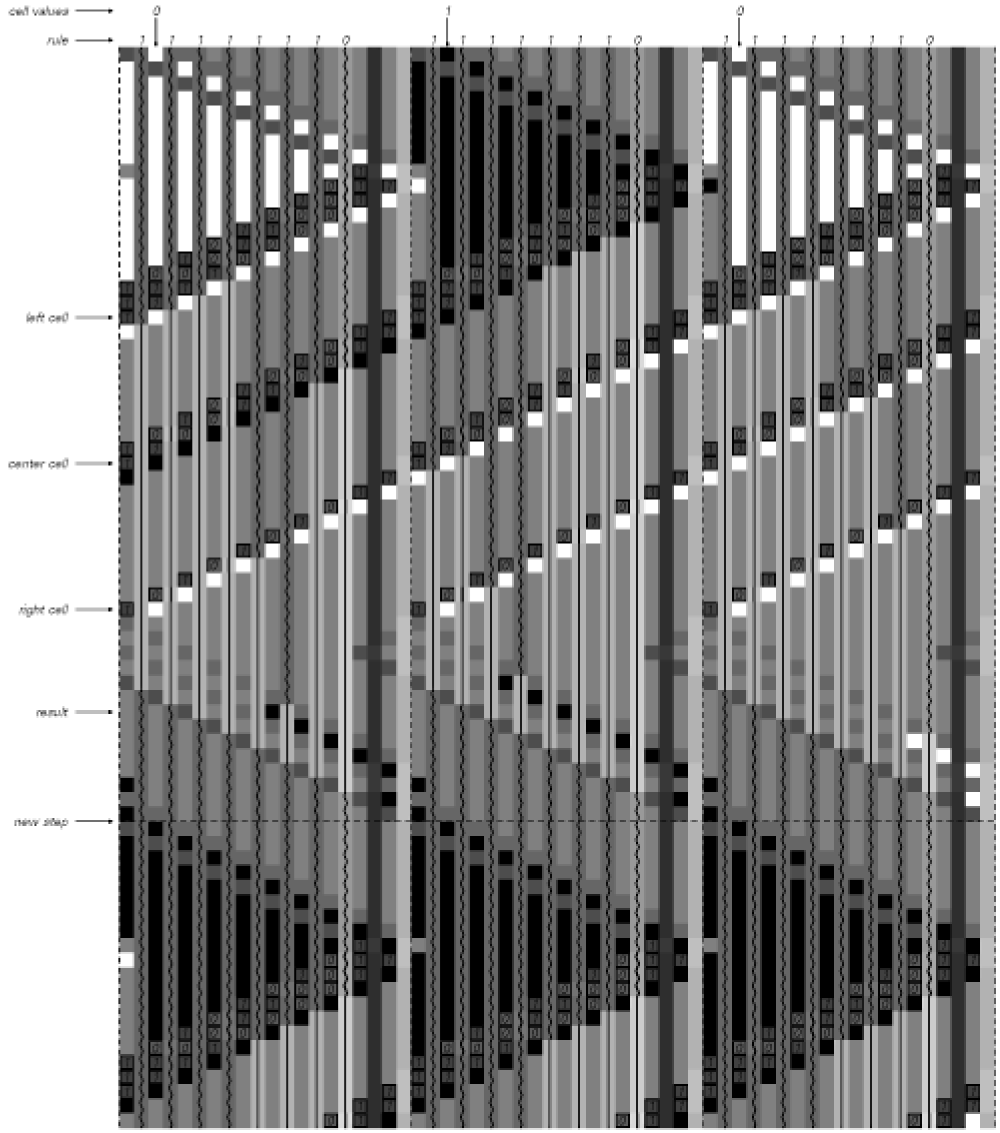
A universal machine can emulate any other machines by right initial configure



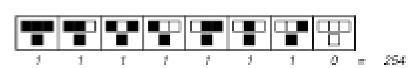
Universal Cellular Automaton



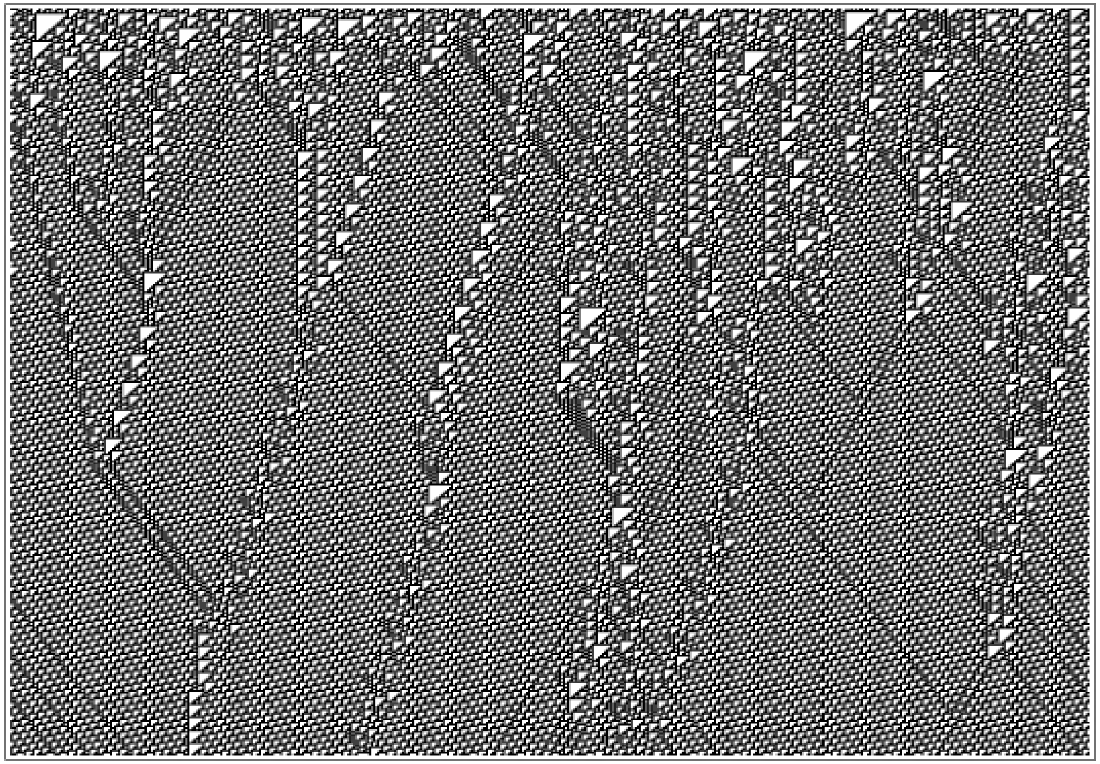
The rules for the universal cellular automaton. There are 19 possible colors for each cell, represented here by 19 different icons. Since the new color of each cell depends on the previous colors of a total of five cells, there are in principle 2,476,099 cases to cover. But by using ⊟ to stand for a cell with any possible color, many cases are combined. Note that the cases shown are in a definite order reading down successive columns, with special cases given before more general ones. With the initial conditions used, there are some combinations of cells that can never occur, and these are not covered in the rules shown.



Details of how the universal cellular automaton emulates rule 254. Each of the blocks in the universal cellular automaton represents a single cell in rule 254, and encodes both the current color of the cell and the form of the rule used to update it.



CA 110



CA 110 is universal, it is really a non-trivial discovery!!!

Skill: Emulation by emergent behavior not by the rules

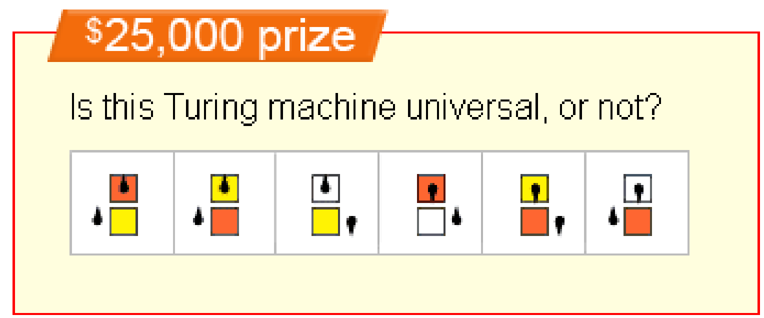
The proof of CA110 is universal

```
CTToR110[rules_ /; Select[rules, Mod[Length[#], 6] ≠ 0 &] == {}, init_] :=
 Module[{g1, g2, g3, nr = 0, x1, y1, sp},
  g1 = Flatten[Map[If[# === {}, {{{2}}}, {{{1, 3, 5 - First[#]}}, Table[{4, 5 - #[[n]]}, {n, 2, Length[#]}]}] &, rules] /.
     a_Integer ↦ Map[{d[#[[1]], #[[2]]], s[#[[3]]]} &, Partition[c[a], 3]], 4];
  g2 = g1 = MapThread[If[#1 === #2 === {d[22, 11], s3}, {d[20, 8], s3}, #1] &, {g1, RotateRight[g1, 6]}];
  While[Mod[Apply[Plus, Map[#[[1, 2]] &, g2]], 30] ≠ 0, nr++; g2 = Join[g2, g1]]; y1 = g2[[1, 1, 2]] - 11; If[y1 < 0, y1 += 30];
  Cases[Last[g2][[2]], s[d[x_, y1], _, _, a_] ↦ (x1 = x + Length[a])]; g3 = Fold[sadd, {d[x1, y1], {}}, g2]; sp = Ceiling[5 Length[g3[[2]]] / (28 nr) + 2];
  {Join[Fold[sadd, {d[17, 1], {}}, Flatten[Table[{{d[sp 28 + 6, 1], s[5]}, {d[398, 1], s[5]}, {d[342, 1], s[5]}, {d[370, 1], s[5]}}, {3}], 1]][[2]], bg[4, 11]],
   Flatten[Join[Table[bgi, {sp 2 + 1 + 24 Length[init]}], init /. {0 → init0, 1 → init1}, bg[1, 9], bg[6, 60 - g2[[1, 1, 1]] + g3[[1, 1]] + If[g2[[1, 1, 2]] < g3[[1, 2]], 8, 0]]]], g3[[2]]}]

s[1] = struct[{3, 0, 1, 10, 4, 8}, 2];

s[2] = struct[{3, 0, 1, 1, 619, 15}, 2];

s[3] = struct[{3, 0, 1, 10, 4956, 18}, 2];

s[4] = struct[{0, 0, 9, 10, 4, 8}];

s[5] = struct[{5, 0, 9, 14, 1, 1}];

{c[1], c[2]} = Map[Join[{22, 11, 3, 39, 3, 1}, #] &, {{63, 12, 2, 48, 5, 4, 29, 26, 4, 43, 26, 4, 23, 3, 4, 47, 4, 4}, {87, 6, 2, 32, 2, 4, 13, 23, 4, 27, 16, 4}}];

{c[3], c[4], c[5]} = Map[Join[#, {4, 17, 22, 4, 39, 27, 4, 47, 4, 4}] &, {{17, 22, 4, 23, 24, 4, 31, 29}, {17, 22, 4, 47, 18, 4, 15, 19}, {41, 16, 4, 47, 18, 4, 15, 19}}]

{init0, init1} = Map[IntegerDigits[216 (# + 432 10^49), 2] &,
  {246 005 560 154 658 471 735 510 051 750 569 922 628 065 067 661, 1 043 746 165 489 466 852 897 089 830 441 756 550 889 834 709 645}]

bgi = IntegerDigits[9976, 2]

bg[s_, n_] := Array[bgi[[1 + Mod[# - 1, 14]]] &, n, s]

ev[s[d[x_, y_], pl_, pr_, b_]] := Module[{r, pl1, pr1}, r = Sign[BitAnd[2^ListConvolve[{1, 2, 4}, Join[bg[pl - 2, 2], b, bg[pr, 2]]], 110]];
  pl1 = (Position[r - bg[pl + 3, Length[r]], 1 | -1] /. {} → {{Length[r]}})[[1, 1]]; pr1 = Max[pl1, (Position[r - bg[pr + 5 - Length[r], Length[r]], 1 | -1] /. {} → {{1}})[[-1, 1]]];
  s[d[x + pl1 - 2, y + 1], pl1 + Mod[pl + 2, 14], 1 + Mod[pr + 4, 14] + pr1 - Length[r], Take[r, {pl1, pr1}]]]

struct[{x_, y_, pl_, pr_, b_, bl_}, p_Integer : 1] :=
 Module[{gr = s[d[x, y], pl, pr, IntegerDigits[b, 2, bl]], p2 = p + 1}, Drop[NestWhile[Append[#, ev[Last[#]]] &, {gr}, If[Rest[Last[#]] === Rest[gr], p2--]; p2 > 0 &], -1]]

sadd[{d[x_, y_], b_}, {d[dx_, dy_], st_}] := Module[{x1 = dx - x, y1 = dy - y, b2, x2, y2}, While[y1 > 0, {x1, y1} += If[Length[st] == 30, {8, -30}, {-2, -3}]];
  b2 = First[Cases[st, s[d[x3_, -y1], pl_, _, sb_] ↦ Join[bg[pl - x1 - x3, x1 + x3], x2 = x3 + Length[sb]; y2 = -y1; sb]]]; {d[x2, y2], Join[b, b2]}]
```

Finding Minimum universal machine

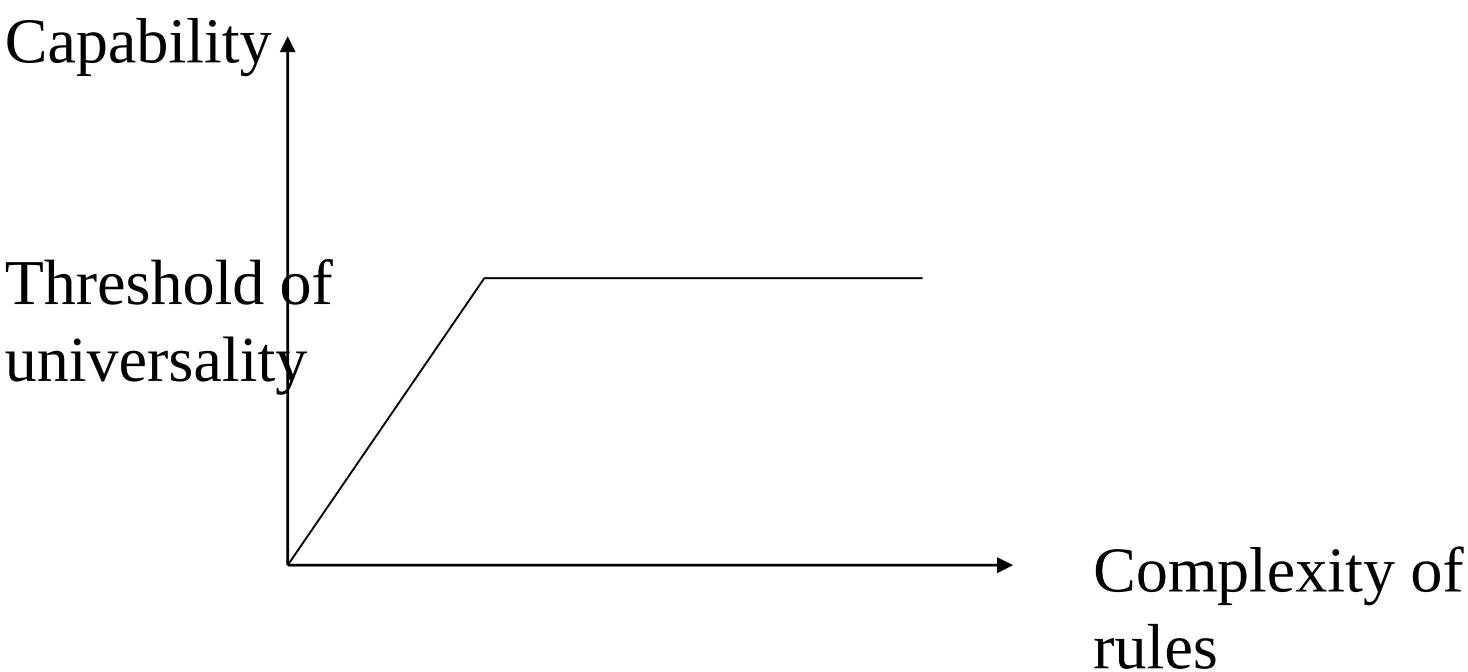
- 1962: TM (7 states, 4 colors)
- 2002: CA110
- 2002: Turing machine (2 states, 5 colors)
- Wolfram prize:

THE WOLFRAM
2,3 TURING MACHINE
RESEARCH PRIZE



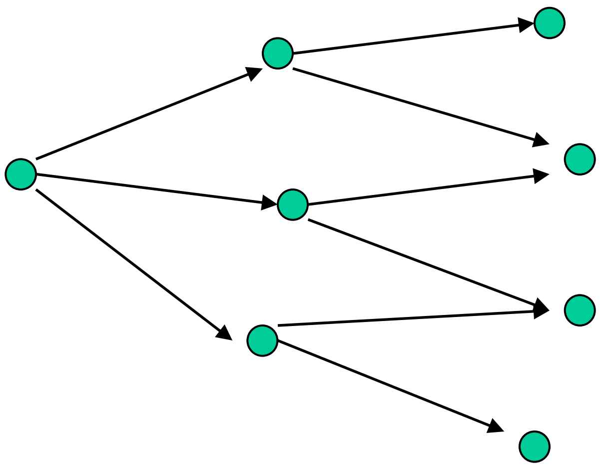
Computational equivalence principle

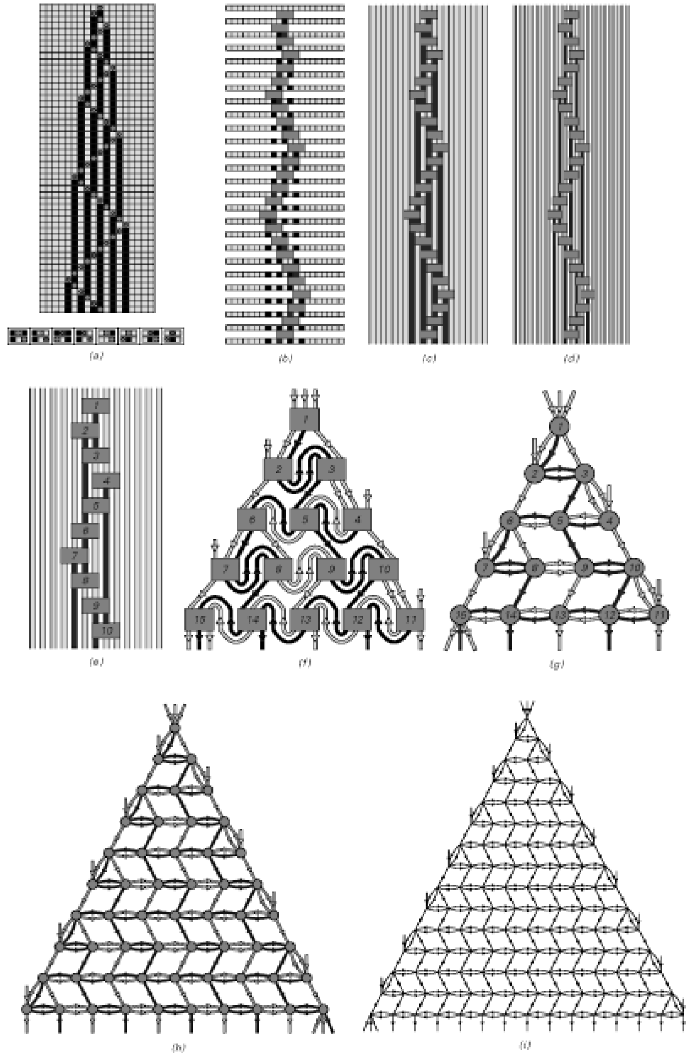
- Any class 4 system is universal
- There is no random class
- Universality instead of complexity

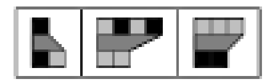


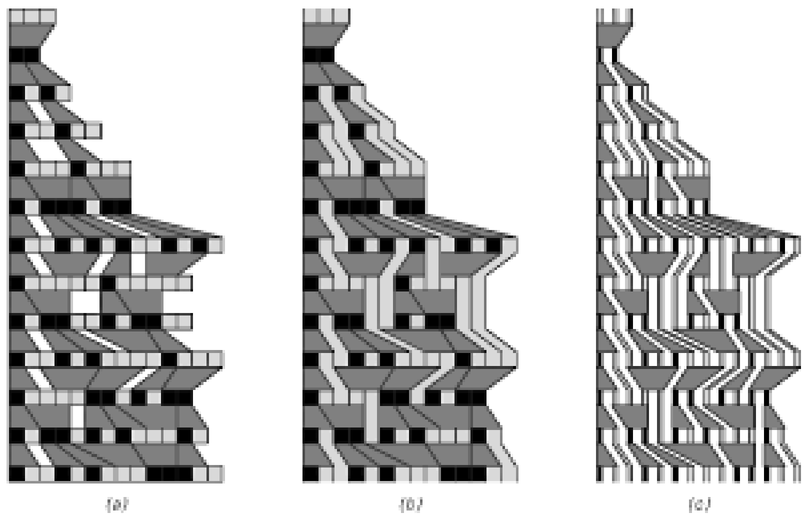
Causal network

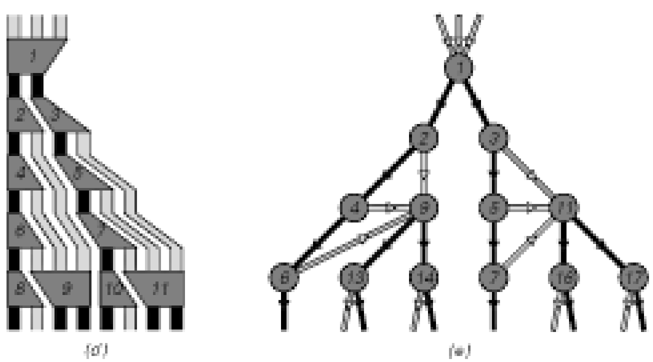
- Every thing is causal
- Event is node, causal effect is edge





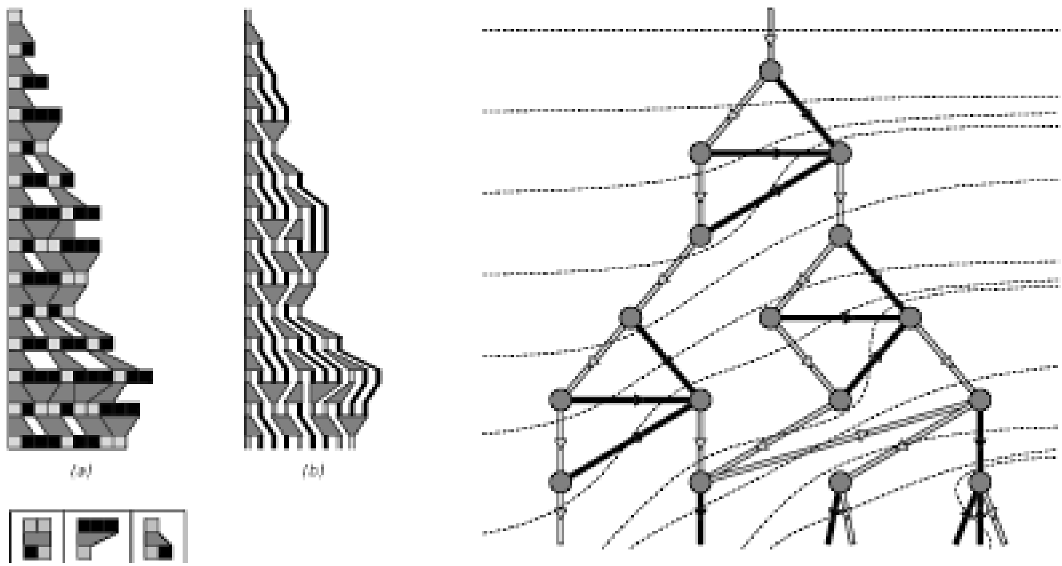






Steps in the construction of a causal network from a general substitution system. The substitution system works by replacing blocks of elements at each step according to the rule shown. Each such updating event becomes a node in the causal network. In the case shown here, all the replacements found to fit in a left-to-right scan are carried out at each step.

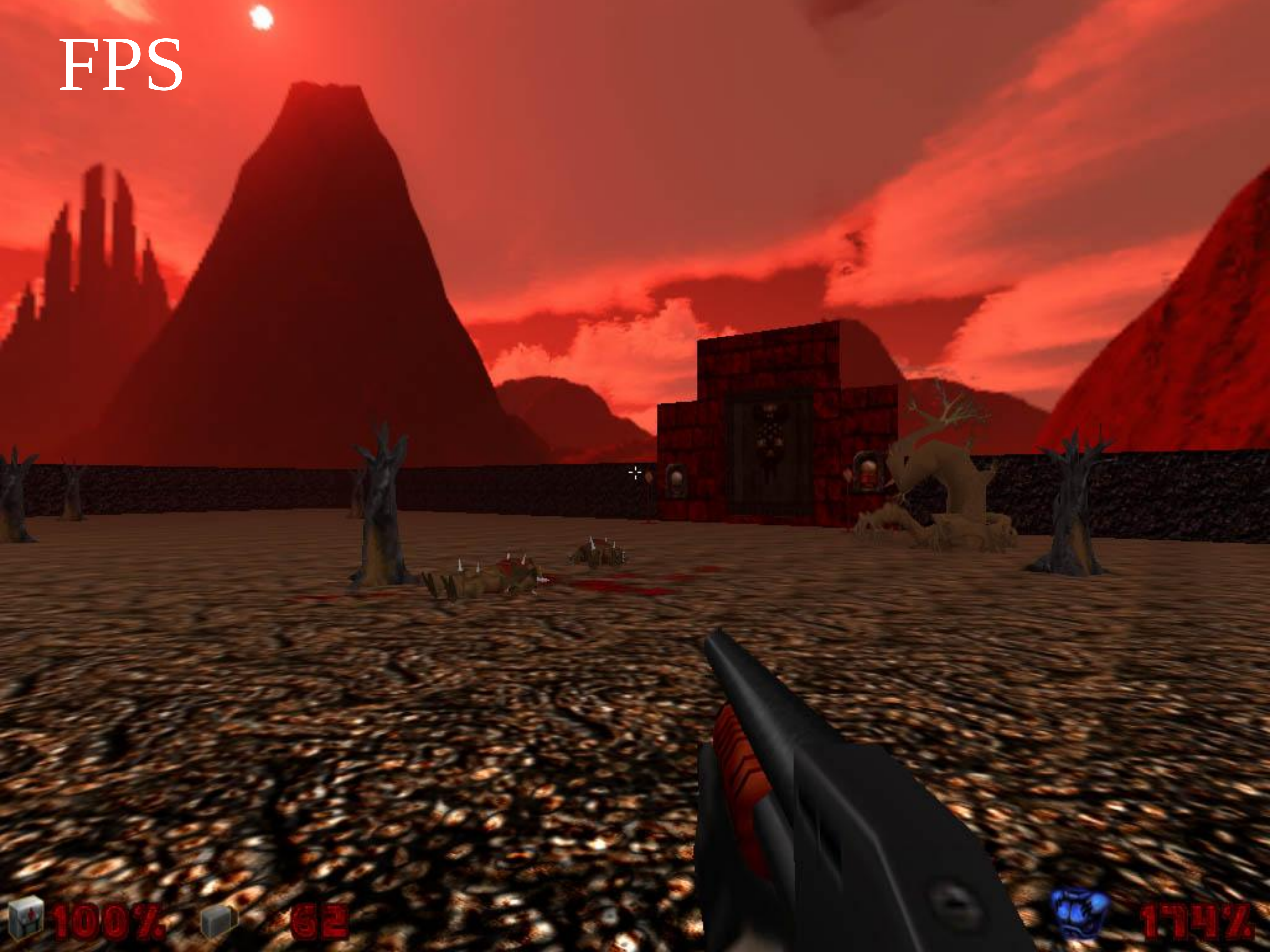
Different ways to view causal network



Problems

- Digitalism or reductionism?
- All computational models are closed
- 2nd Thermodynamic law?
- Open – observer – computer games

FPS



RPG

RTS

ALT

New field emerging

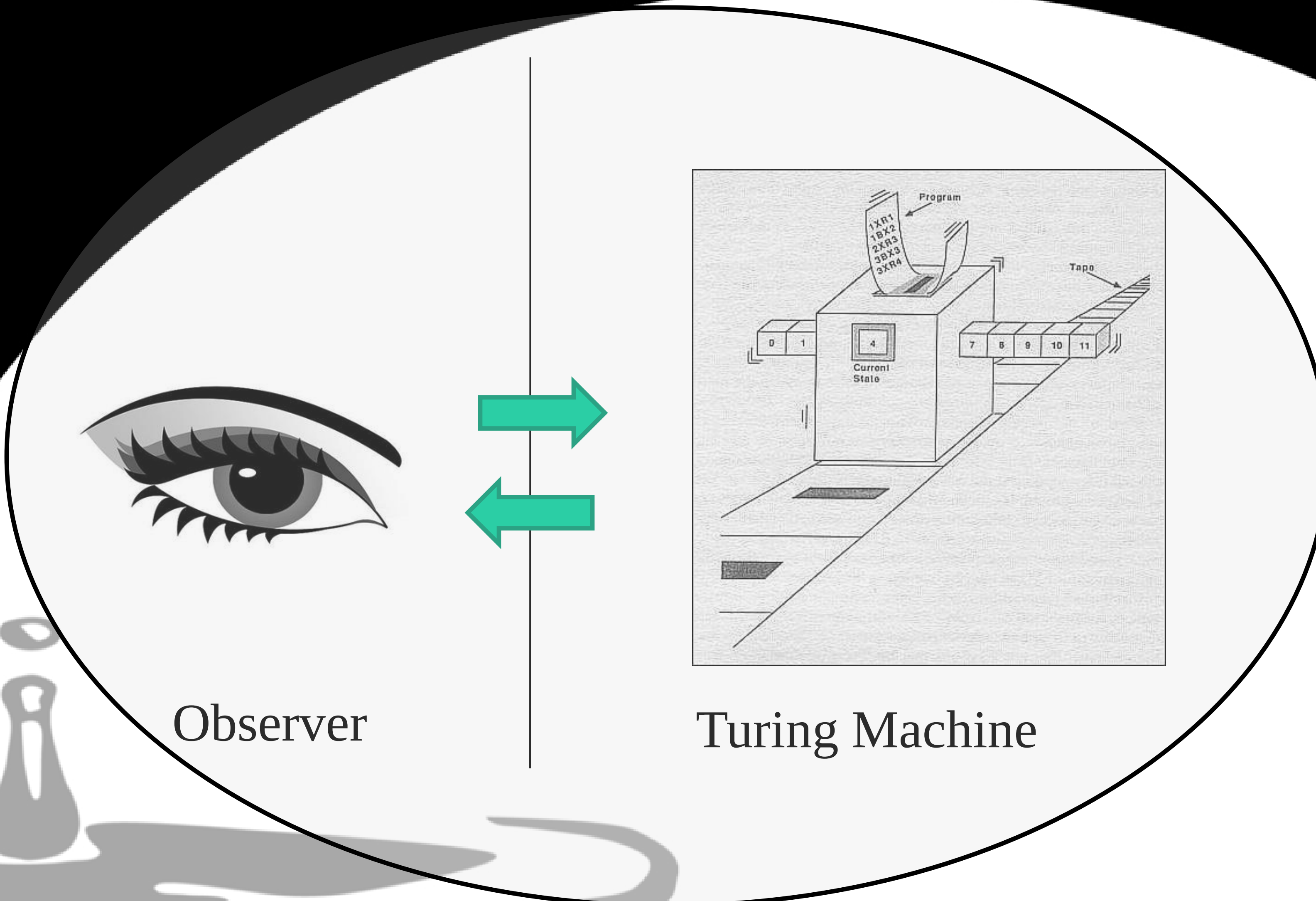
World of Warcraft

Second life

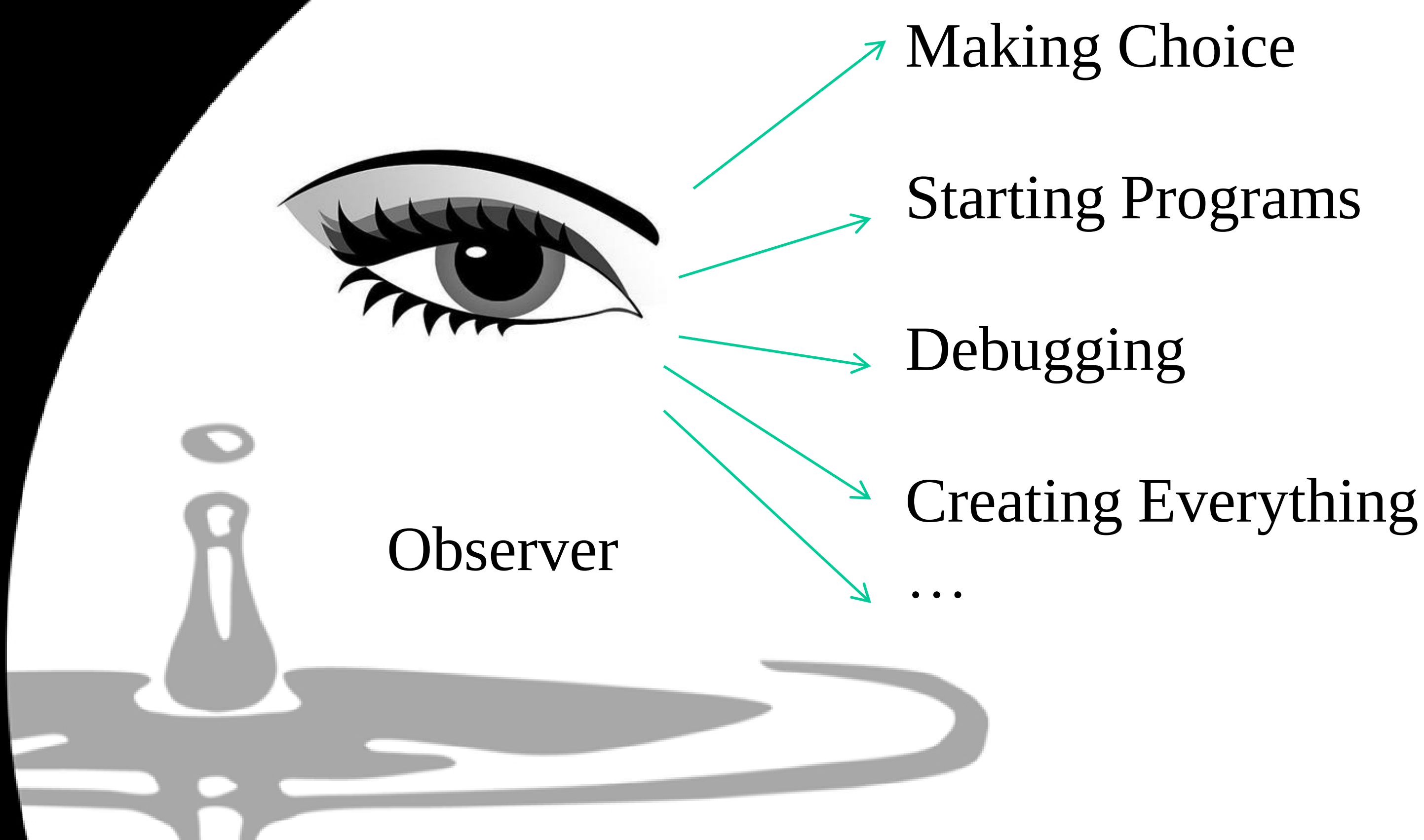
W.S. Bainbridge: The Scientific Research Potential of VIRTUAL WORLDs, Science, vol 317, 2007

Jim Giles, Social Sciences: Life's A Game , Nature 445, 18-20, 2007/01/04

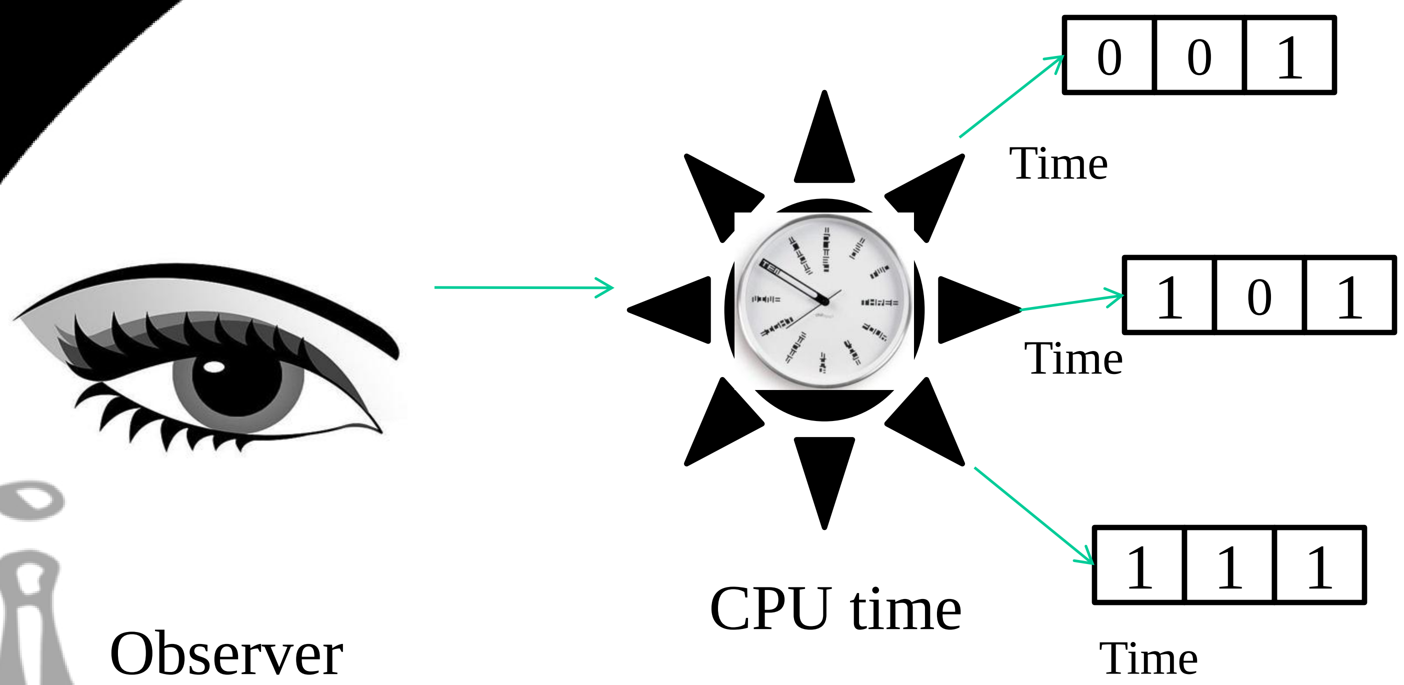
Turing Machine – Observer Model



Observer – The Engine of Machine



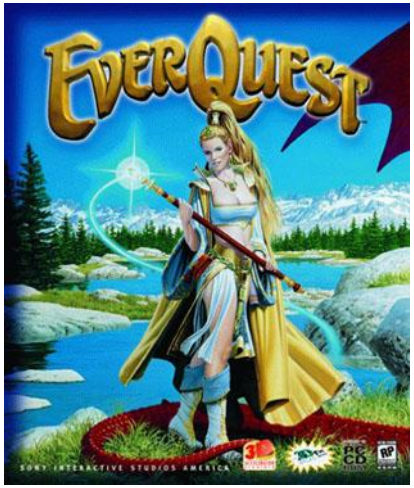
Play in Time



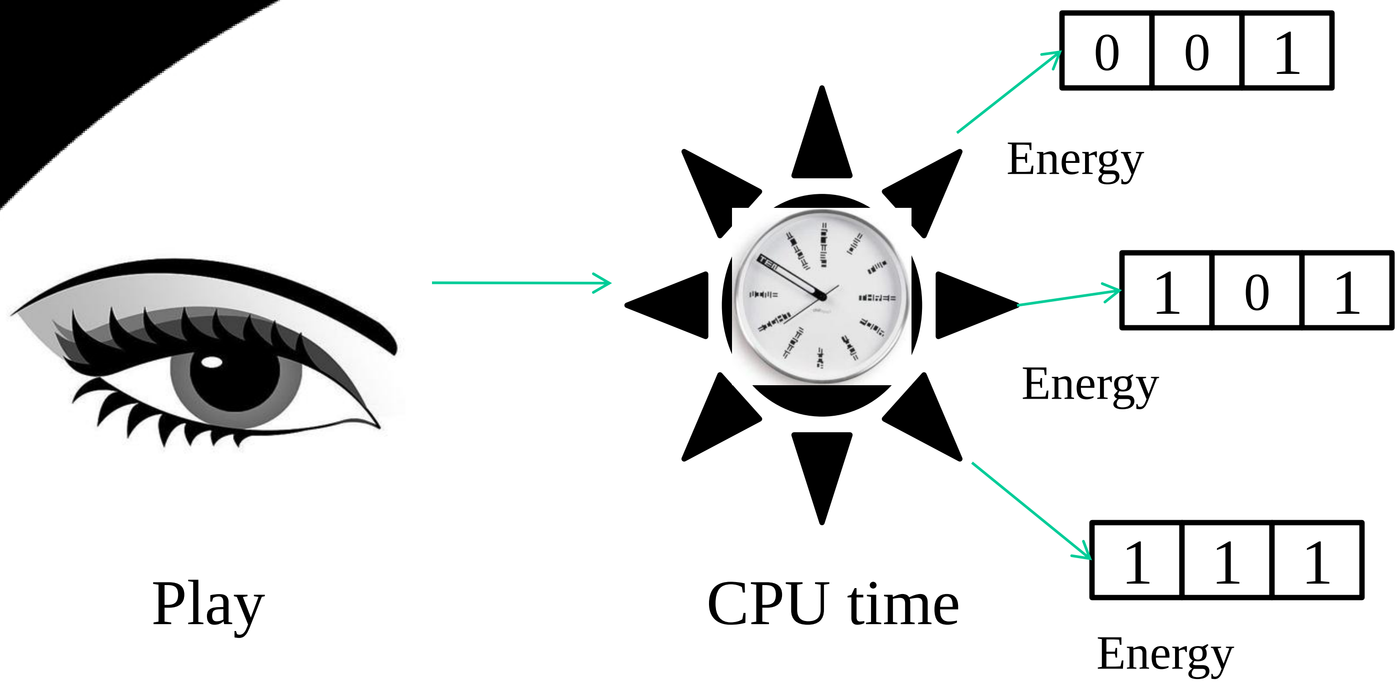
Example: GDP of Everquest

- Edward Castronova: Economist at Indiana University in Bloomington
- 2001:
- A First-Hand Account of Market and Society on the Cyberian Fontier

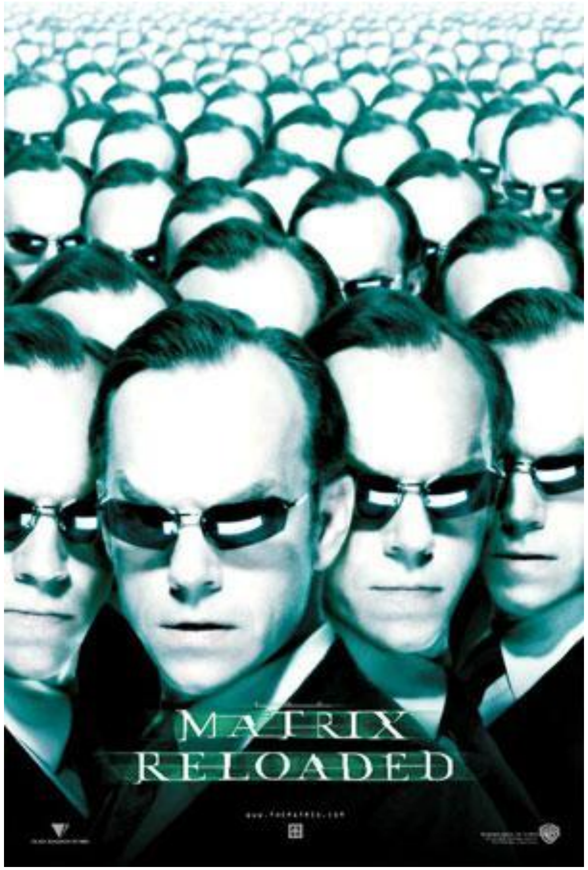
"I would venture to claim that this is the first time in human history that a distinct macrosocial phenomenon has actually been verified experimentally."



Play = Production



A "Real" Metaphor

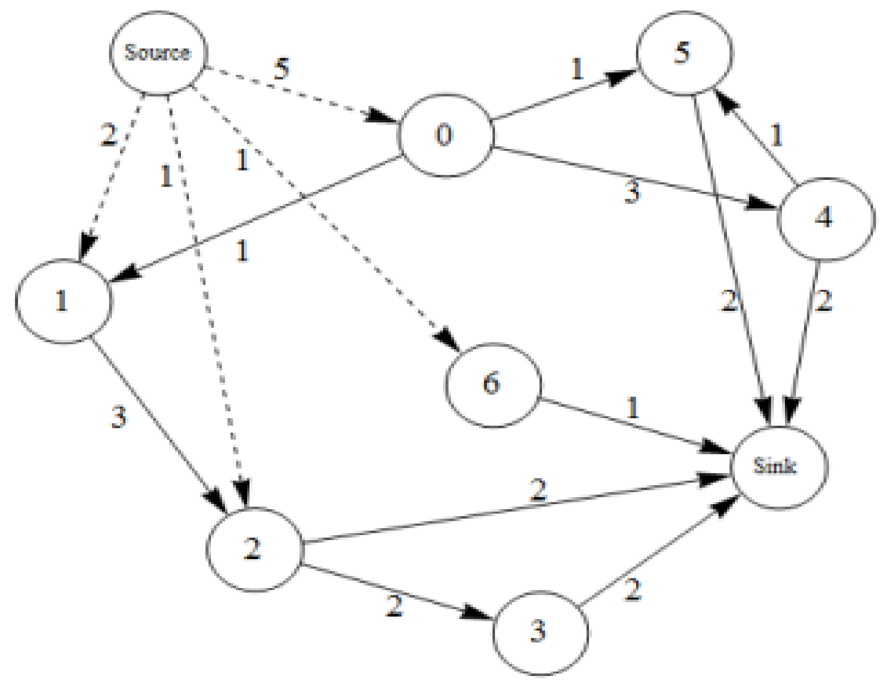


Clickstream

Network Construction

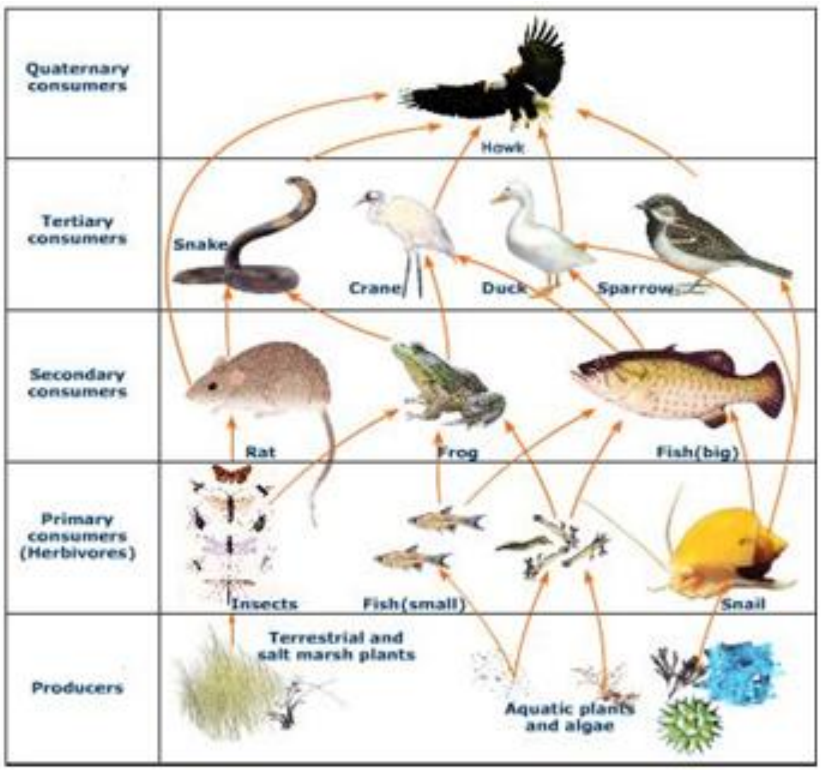
a	1
a	2
b	1
b	2
c	1
c	2
c	3
d	2
d	3
e	0
e	4
f	0
f	4
g	0
g	4
g	5
h	0
h	5

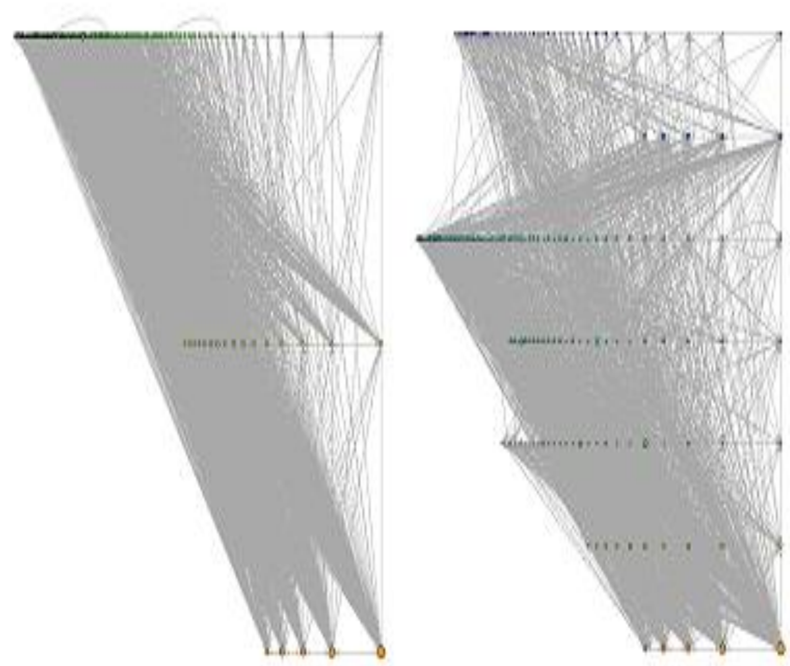
(a)



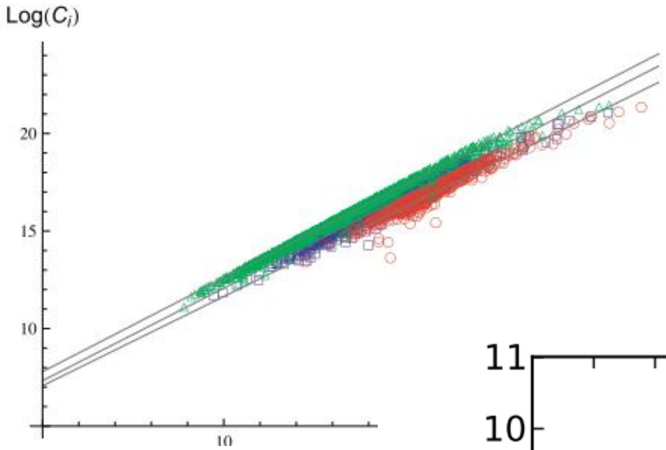
(b)

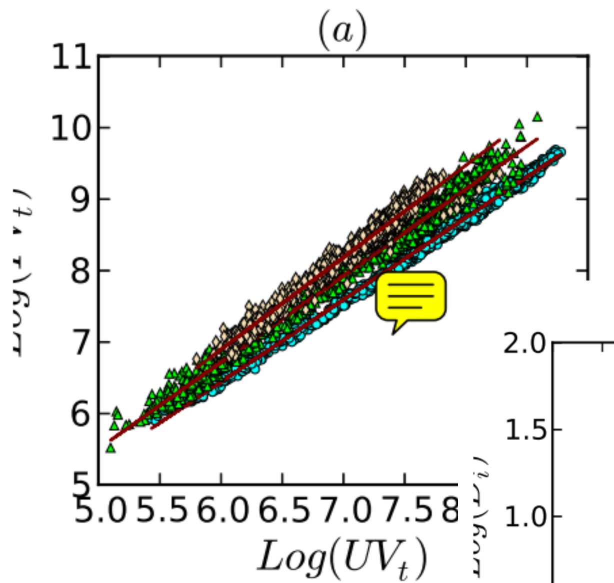
Digital Ecosystem

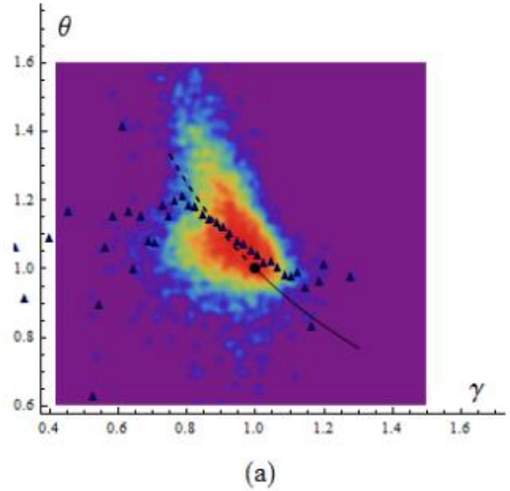


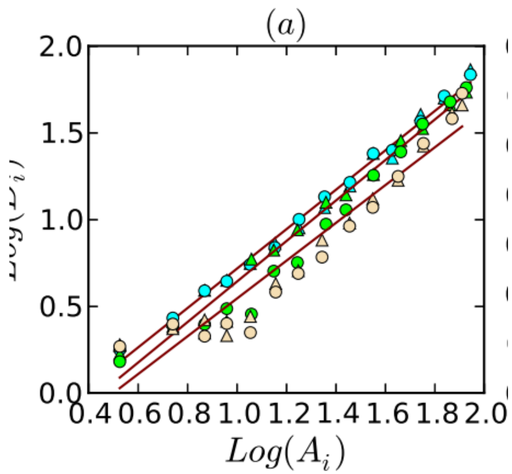


Common laws



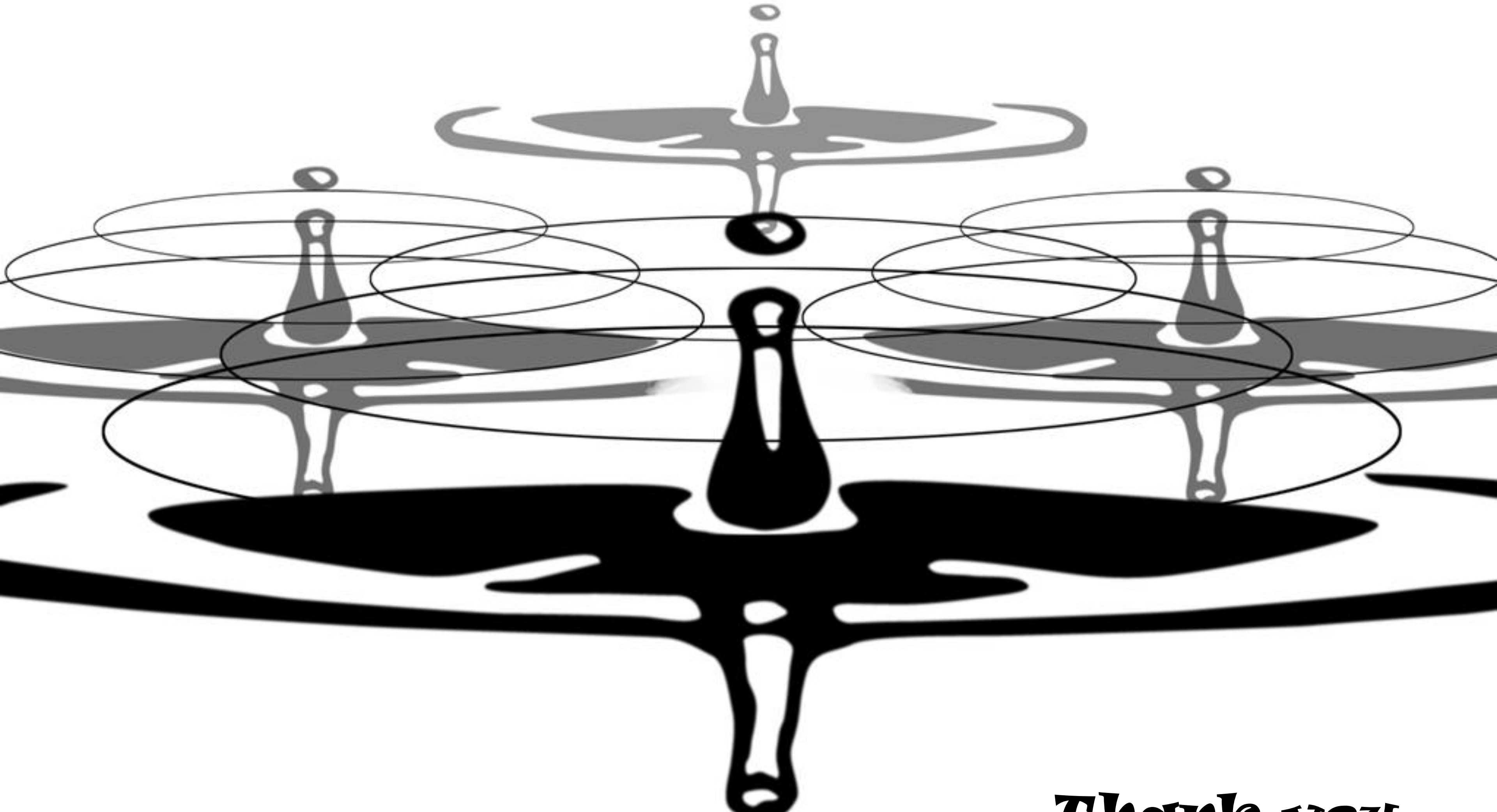






Allometric Scaling
Allometric Growth
Dissipation law
Negative correlation

Thank you



Thank you