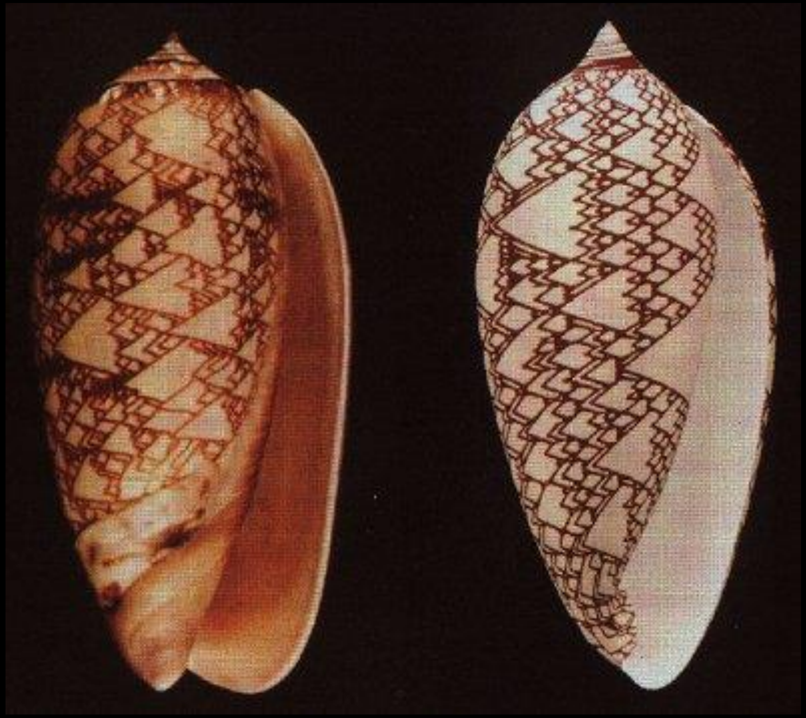
Symulacje komputerowe w fizyce



17-18.01.2023

Wykład 13: Cztery linijki kodu

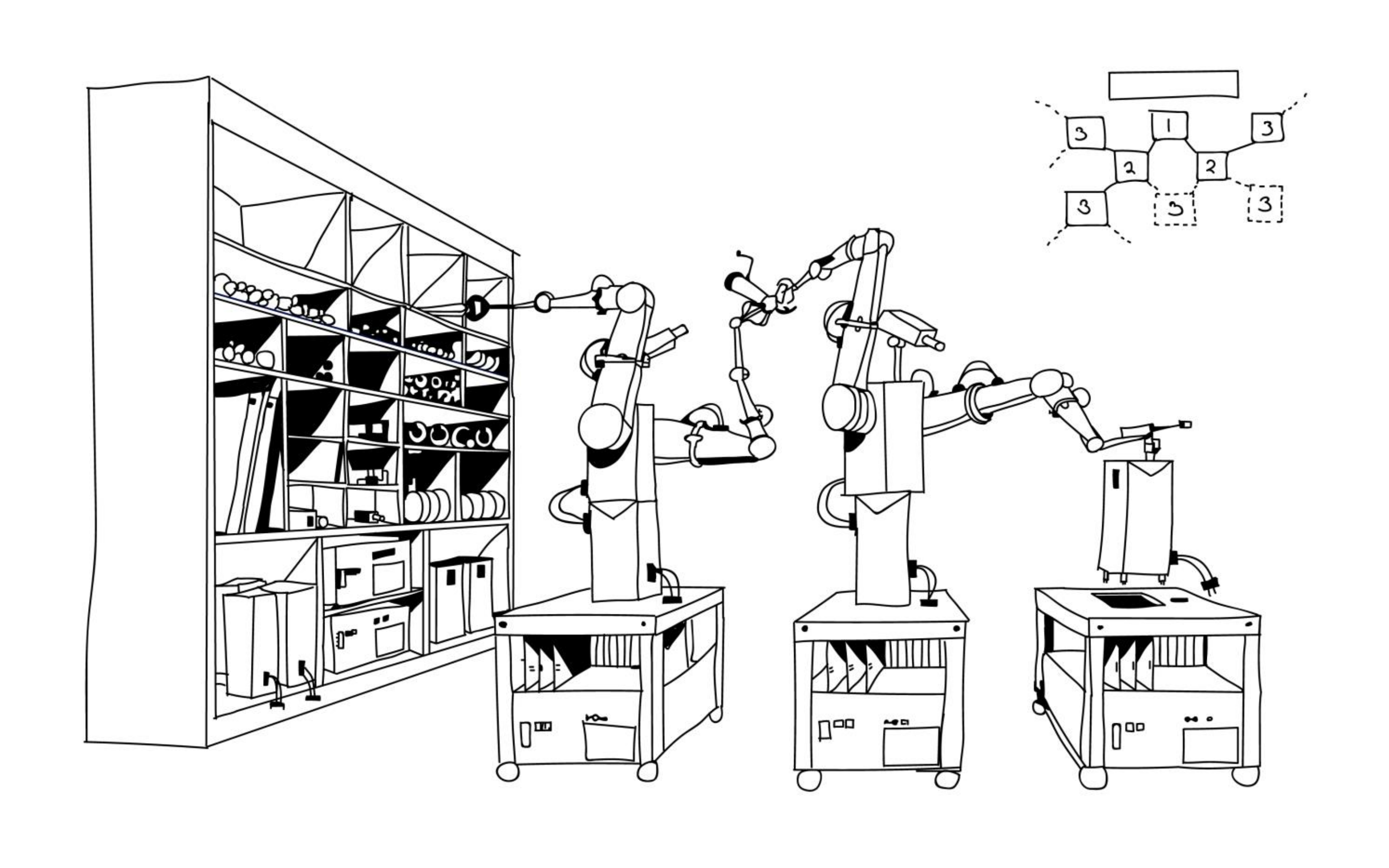
Automaty komórkowe

Historia

Stanisław Ulam, 1909-1984

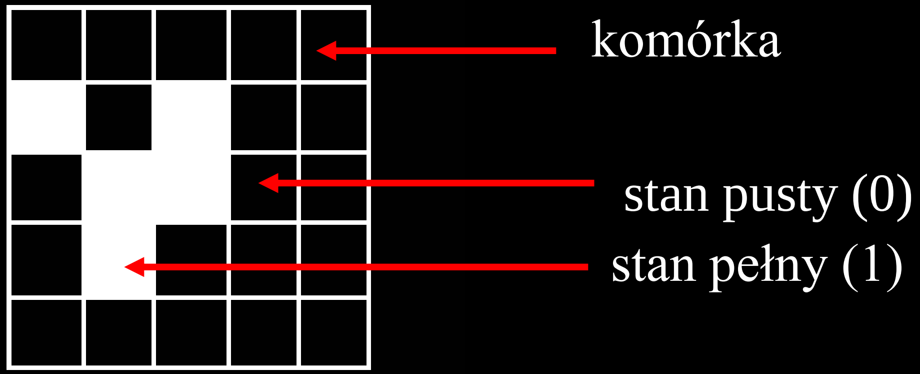
John von Neumann, 1903-1957

Historia



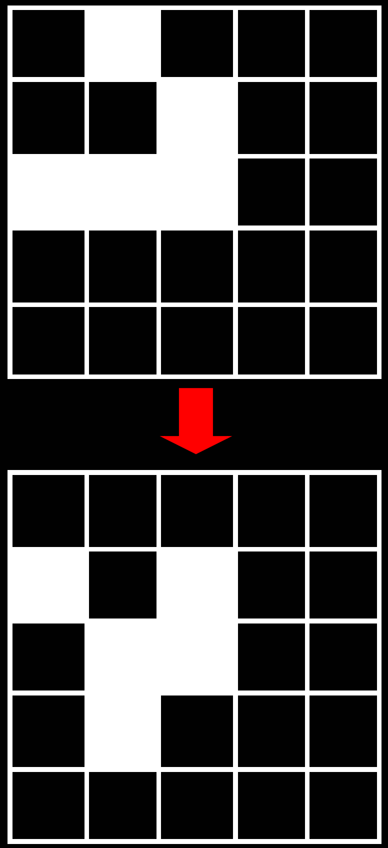
maszyna samoreplikująca

Automaty komórkowe



Automat składa się z siatki komórek, z których każda może znajdować się w jednym ze skończonej liczby stanów.

Ewolucja



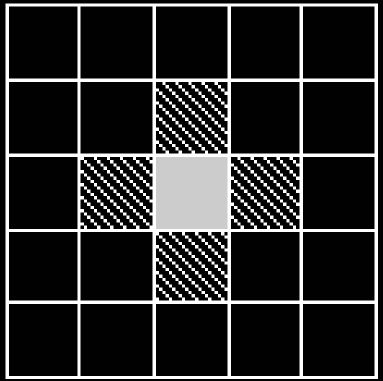
T=1

T=2

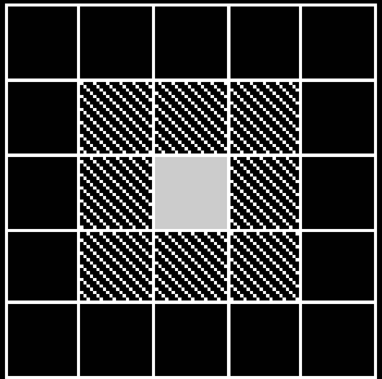
Automat komórkowy ewoluuje zmieniając stany swoich komórek zgodnie z *regułami przejścia*, które określają przyszły stan komórki w zależności od stanu obecnego jej i jej sąsiedztwa.

Sąsiedztwo

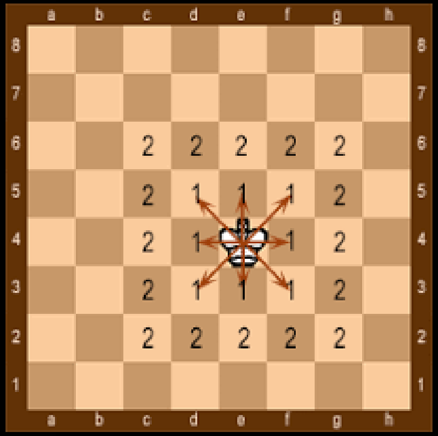
- określa stopień sprzężenia między komórkami wewnątrz siatki
- dwa najpopularniejsze sąsiedztwa (w 2 wymiarach):



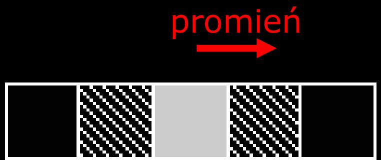
von Neumann Neighbourhood



Moore Neighbourhood



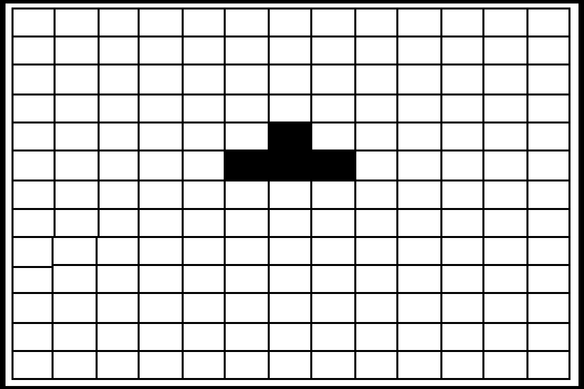
-w jednym wymiarze:



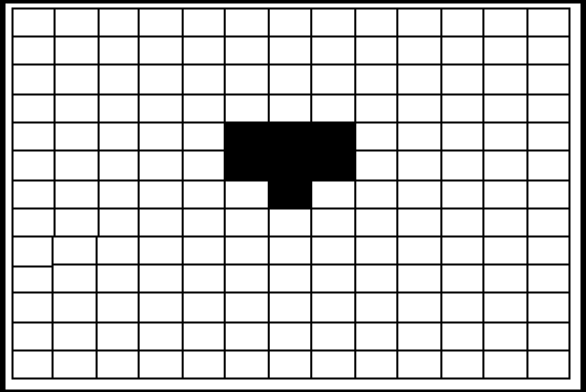
Przykład (gra w życie)

Rozpatrzmy następujące reguły przejścia:

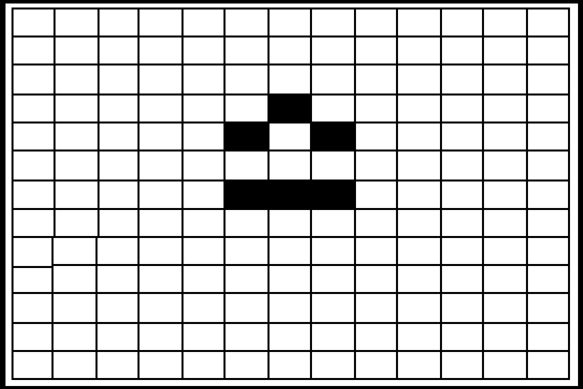
- komórka pozostaje w stanie 1 (czarnym), jeśli ma dwóch lub trzech czarnych sąsiadów (Moore'a),
- komórka zmienia się na czarną, jeśli ma dokładnie 3 czarnych sąsiadów,
- w pozostałych przypadkach komórka będzie biała.



stan początkowy

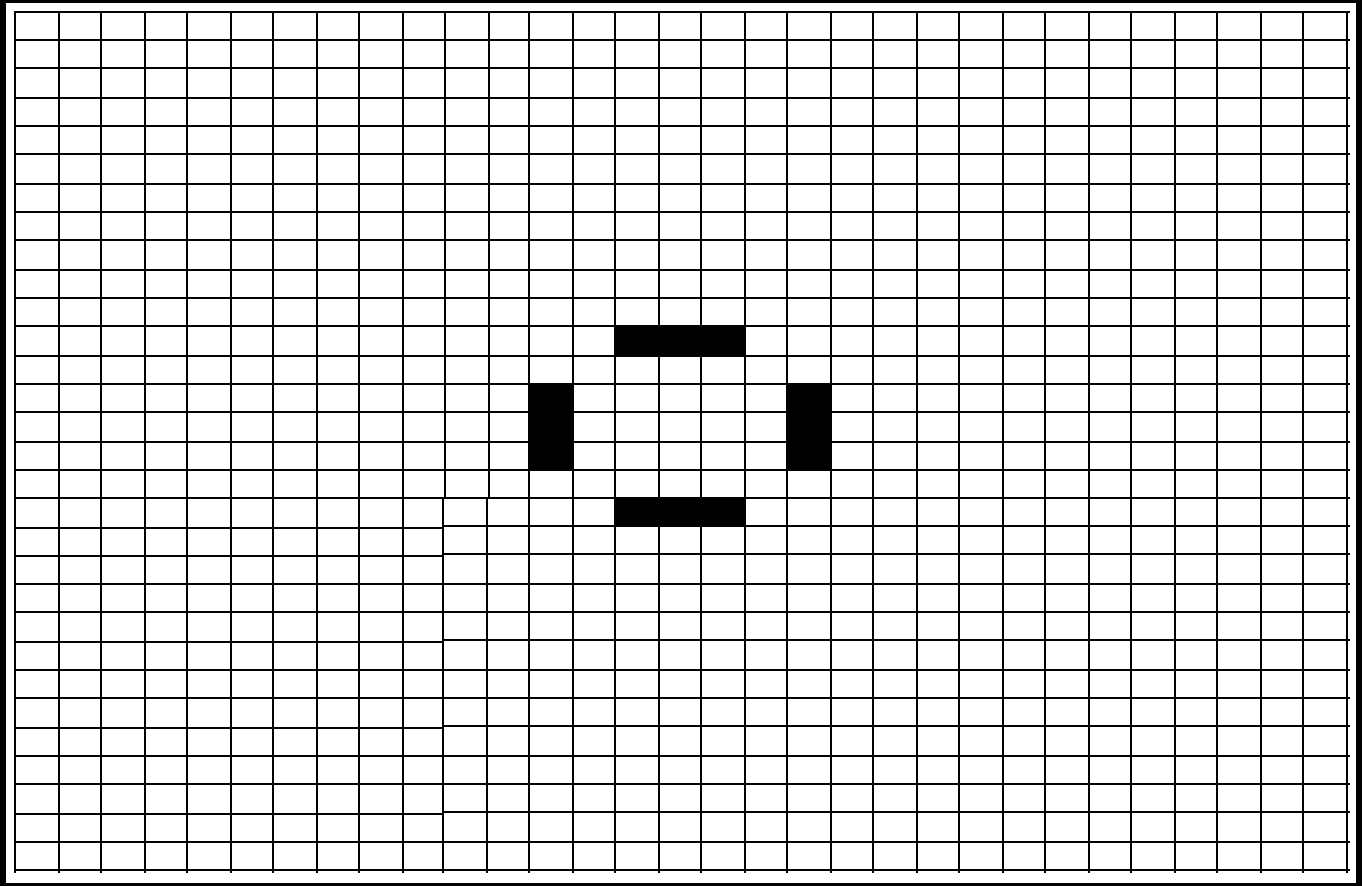


krok 1

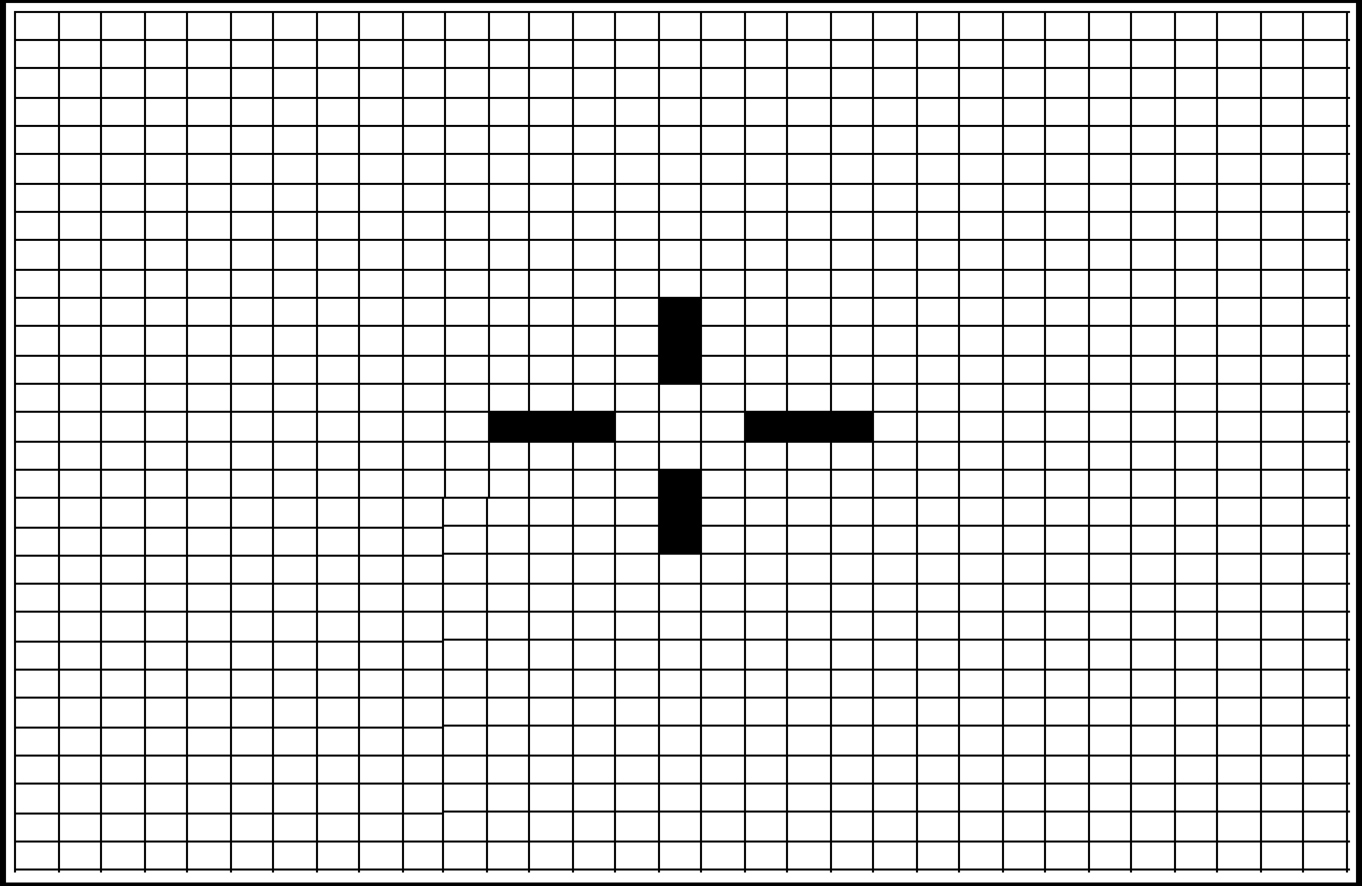


krok 2

Ewolucja...



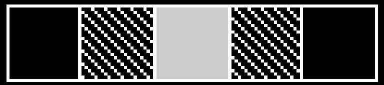
Cykl graniczny...



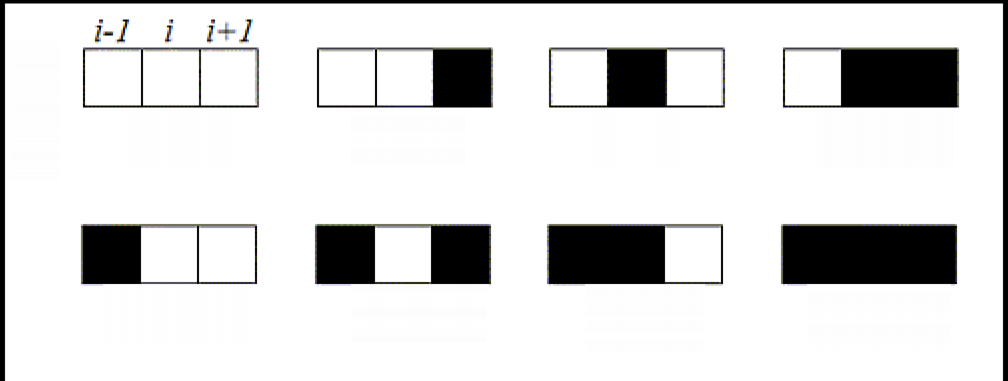
istnienie cykli granicznych $\Rightarrow$ nieodwracalność!

Automaty jednowymiarowe

Rozpatrzmy automaty jednowymiarowe z sąsiedztwem o promieniu 1.



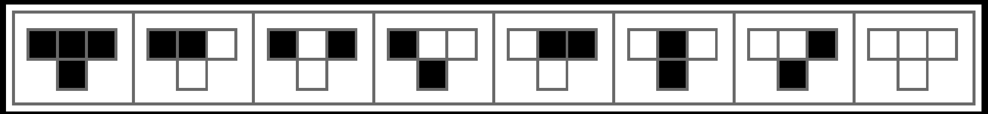
Sąsiedztwo takie może być w jednym z 8 stanów:



Reguła przejścia musi mówić, co zrobić w każdym z tych przypadków.

$$c_i(t+1) = \varphi(c_{i-1}(t), c_i(t), c_{i+1}(t))$$

Przykład

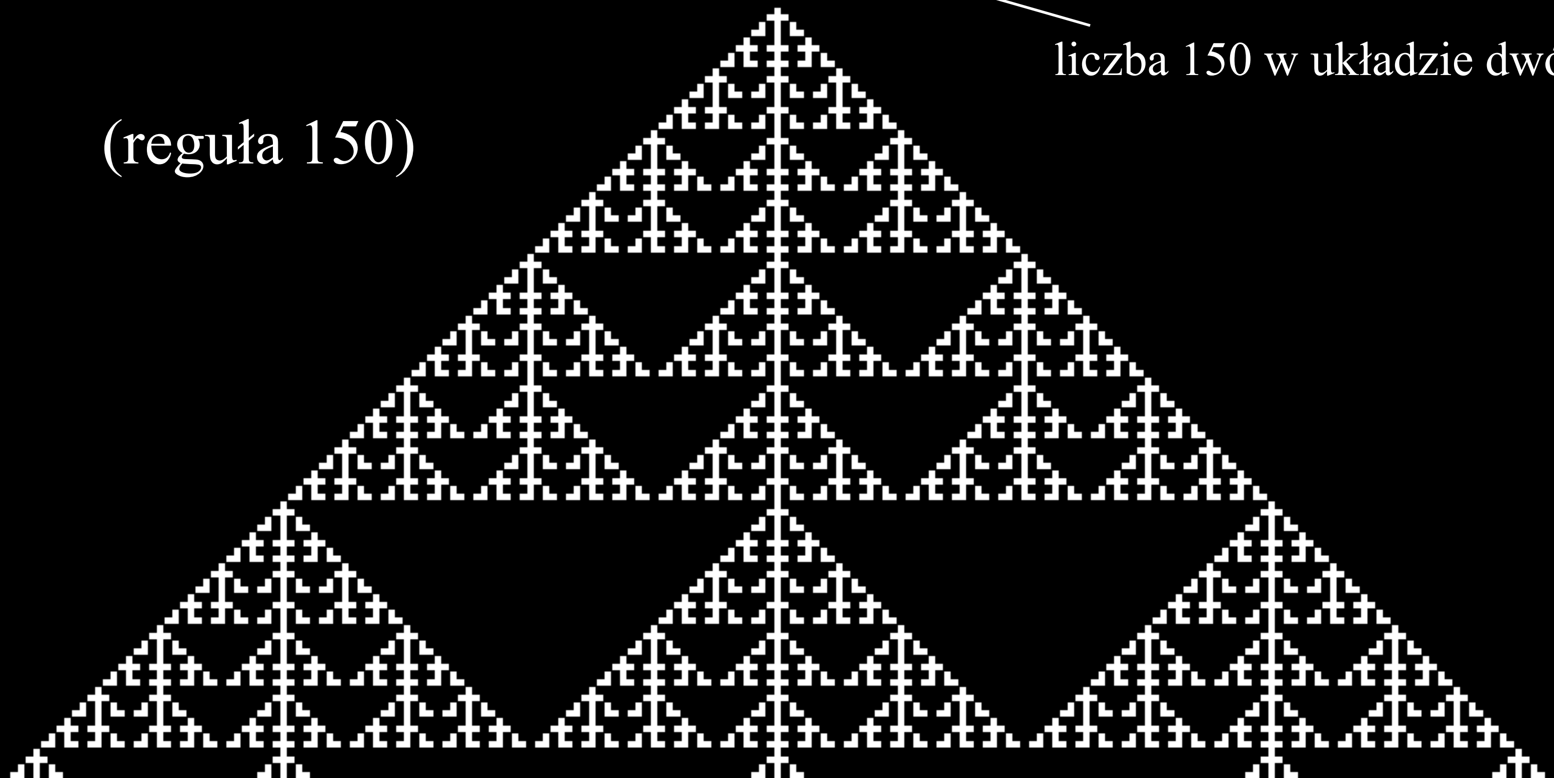


w sumie 256 różnych reguł!

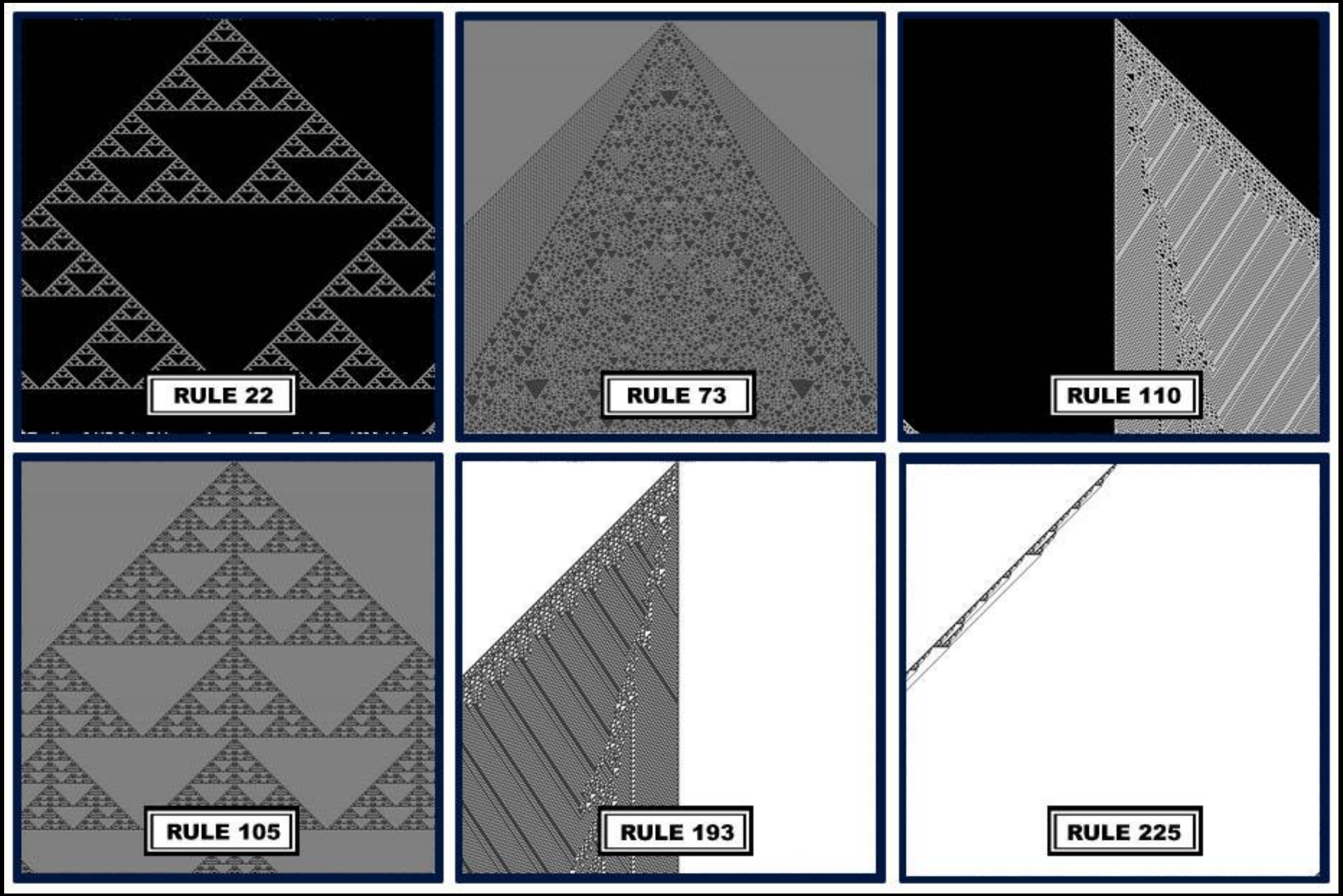
111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

liczba 150 w układzie dwójkowym

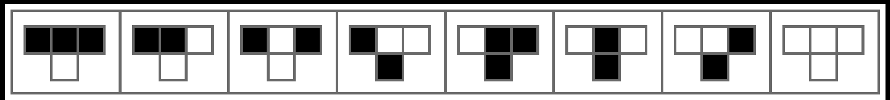
(reguła 150)



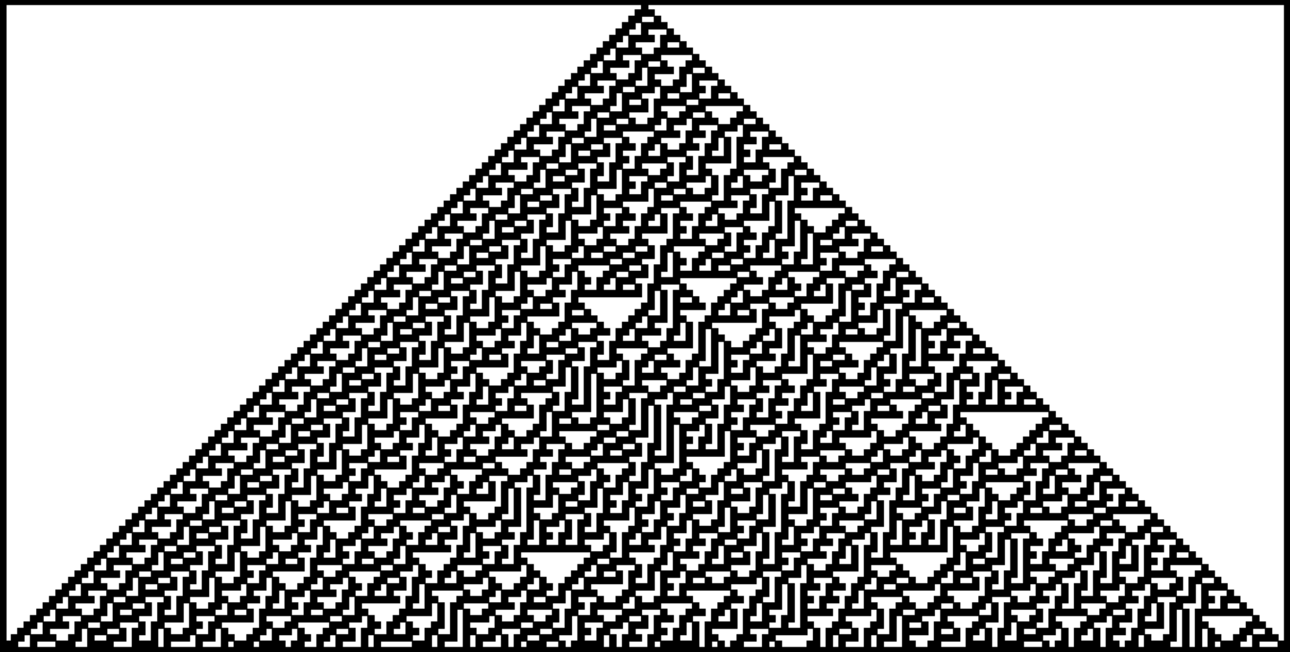
Dalsze przykłady



Reguła 30



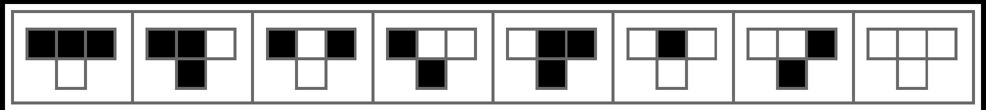
$$c_i(t+1) = c_{i-1}(t) + c_i(t) + c_{i+1}(t) + c_i(t)c_{i+1}(t) \pmod 2$$



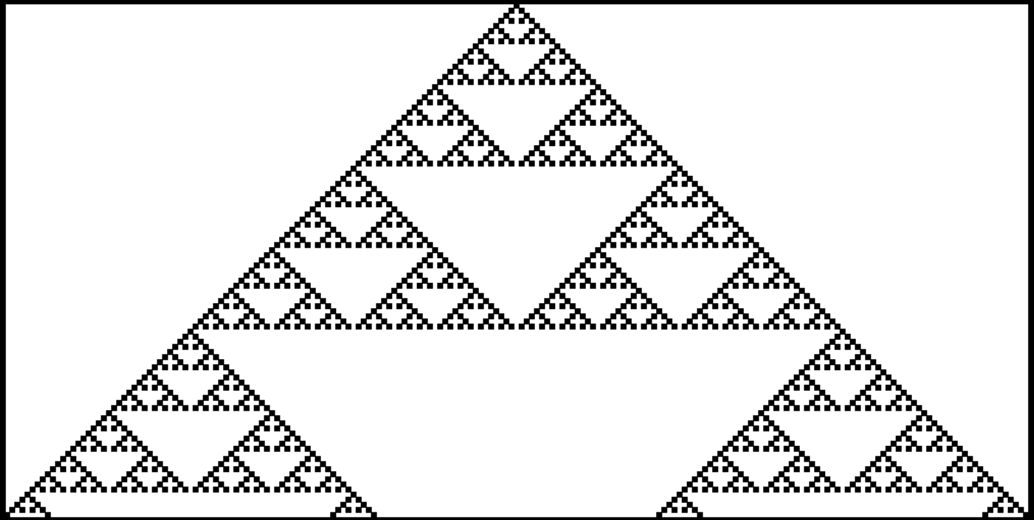
centralna kolumna: 1001110011... ← znakomicie przypadkowa

Perrenoud, M. et al., *On the generation of high-quality random numbers by two-dimensional cellular automata (2000).*

Reguła 90



$$c_i(t+1) = c_{i-1}(t) + c_{i+1}(t) \pmod 2$$

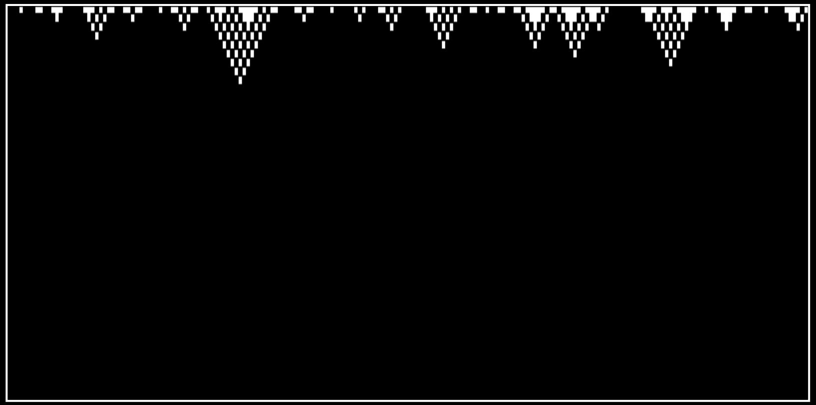


trójkąt Pascala:

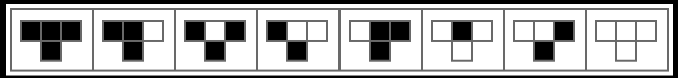
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

reguła 90 pokazuje nieparzyste współczynniki

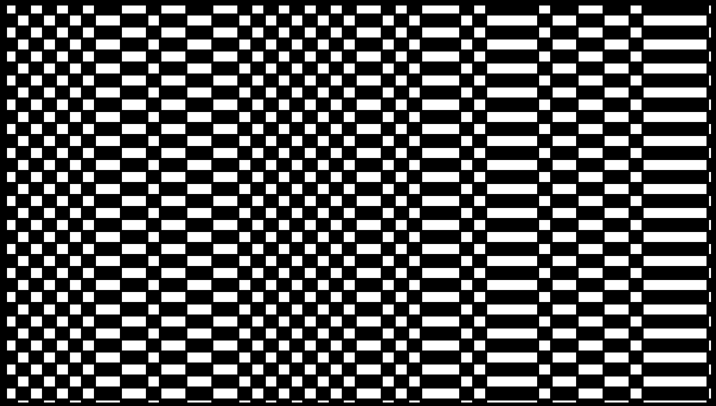
Nieodwracalność ewolucji



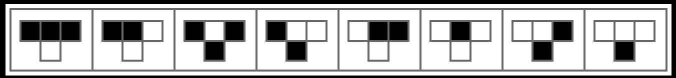
reguła 250



wszystko zamienia się w czerń – nieodwracalny

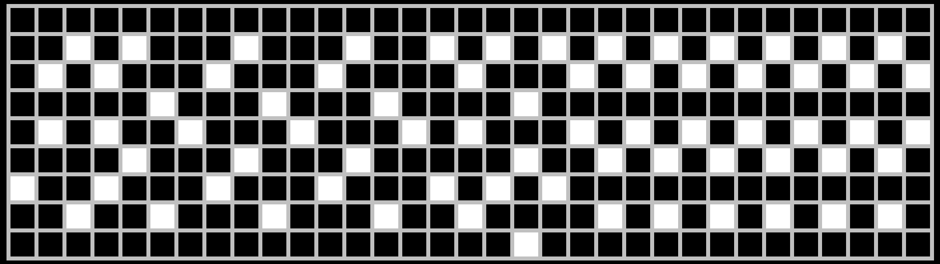


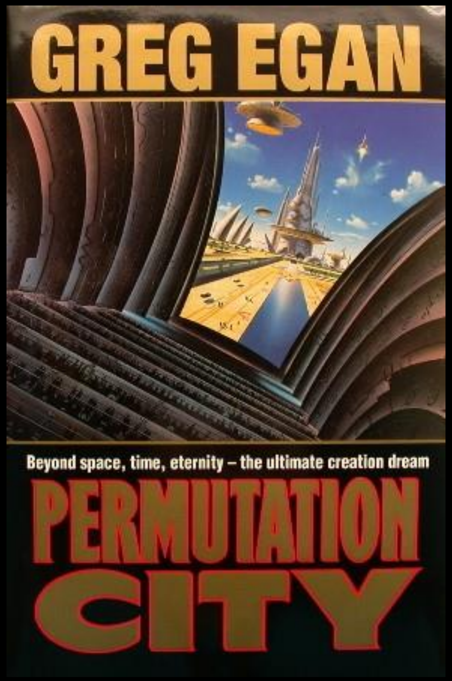
reguła 51



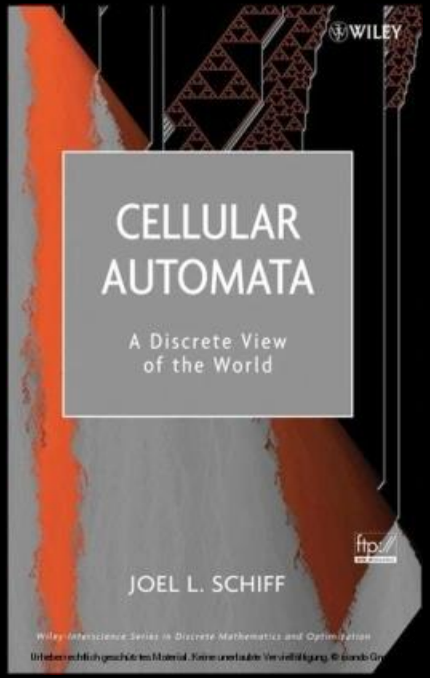
każda komórka zmienia kolor w kolejnym kroku – trywialnie odwracalny

Ogrody Edenu





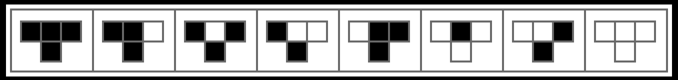
Jak ustalić, czy żyjemy w symulacji?

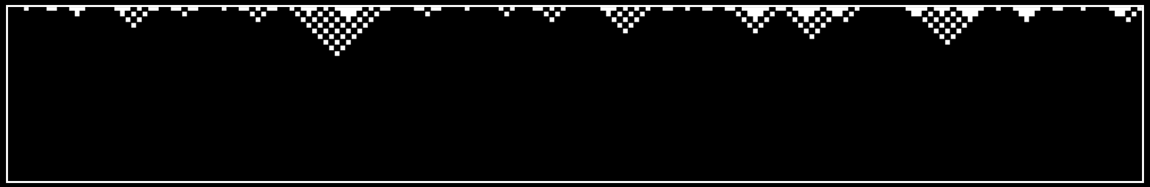


Klasyfikacja automatów jednowymiarowych

Klasa I – kończą ewolucję w stanie jednorodnym (same 0 lub 1)

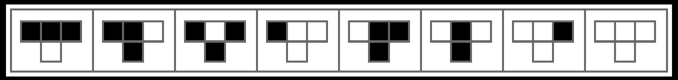
reguła 250:

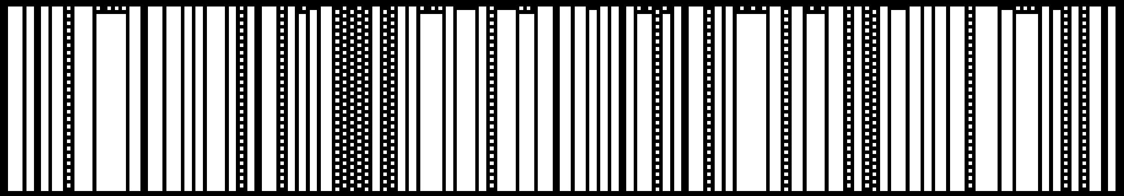




Klasa II – kończą w stanie ustalonym lub okresowym

reguła 108:

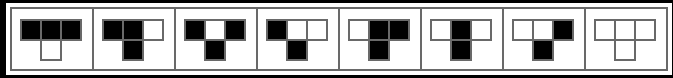




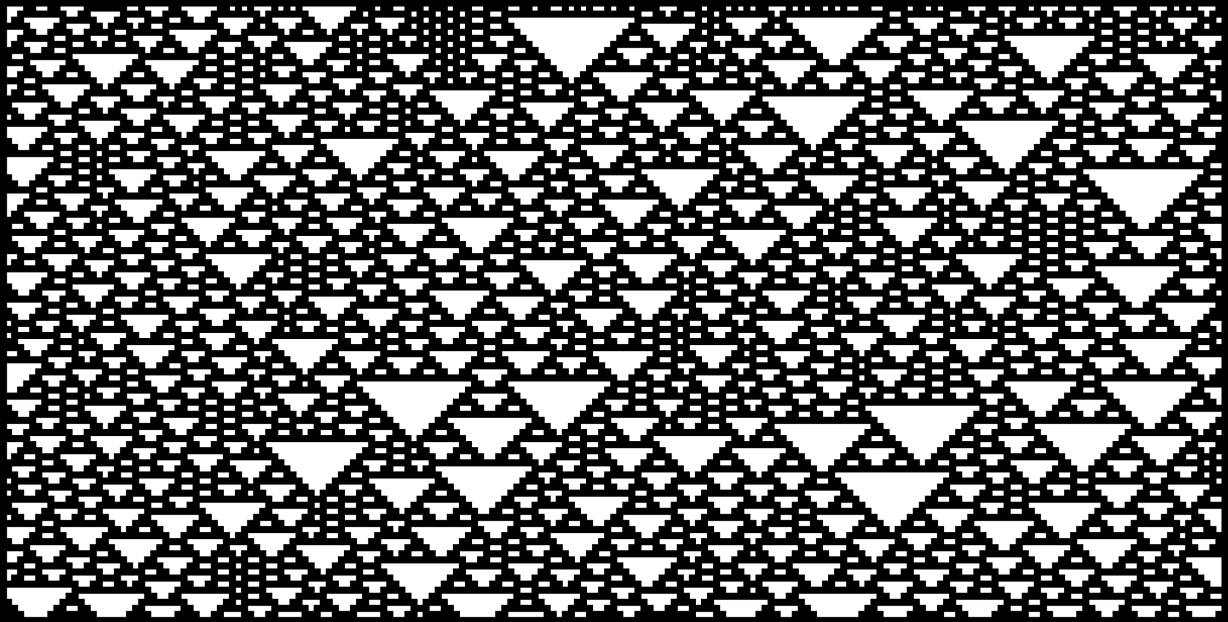
Klasyfikacja automatów jednowymiarowych

Klasa III – wykazują zachowanie chaotyczne (choć deterministyczne)

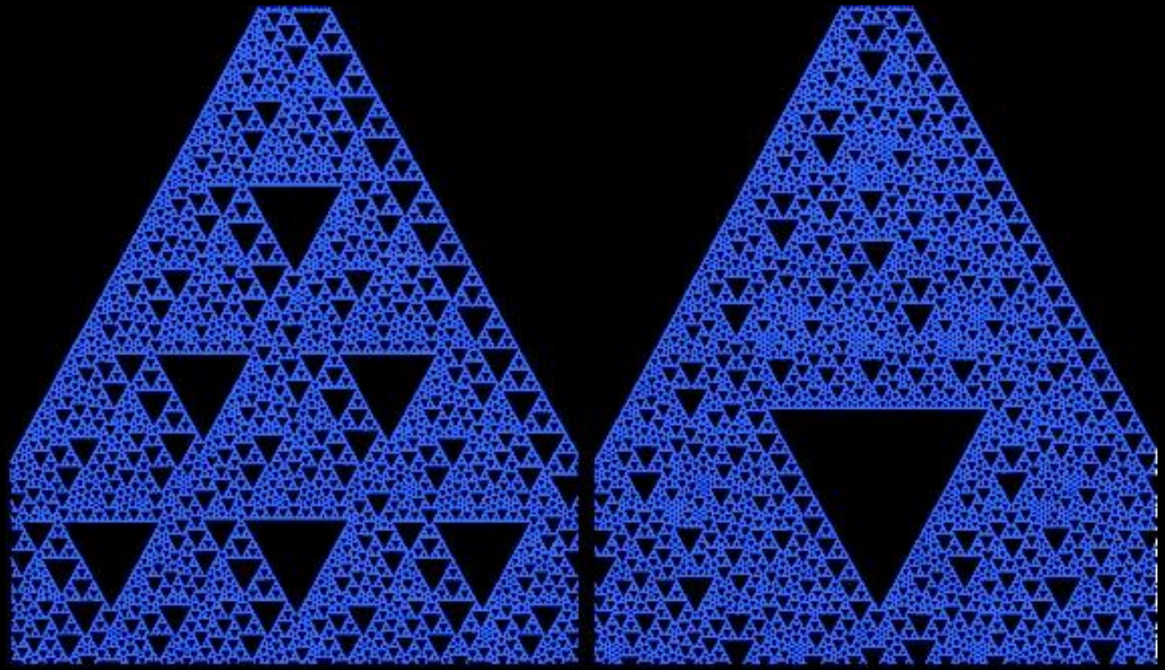
reguła 126:



przypadkowe warunki początkowe

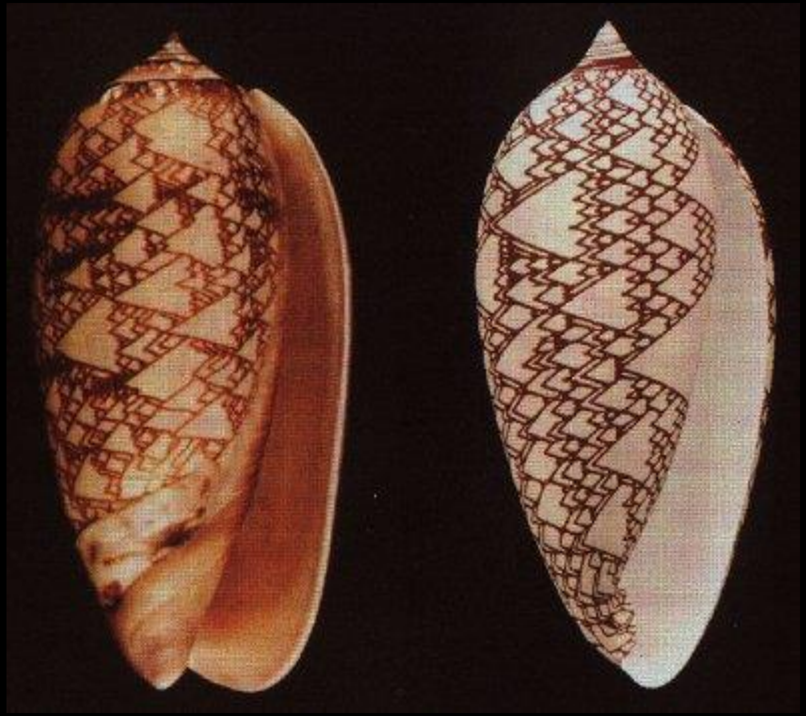


Czułość na warunki początkowe



Ewolucja wg. reguły 126 z warunków początkowych różniących się o stan jednej komórki.

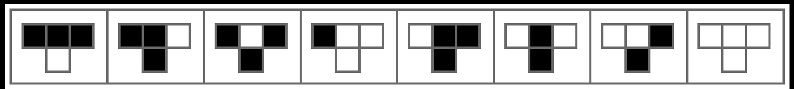
Automaty klasy III na muszlach

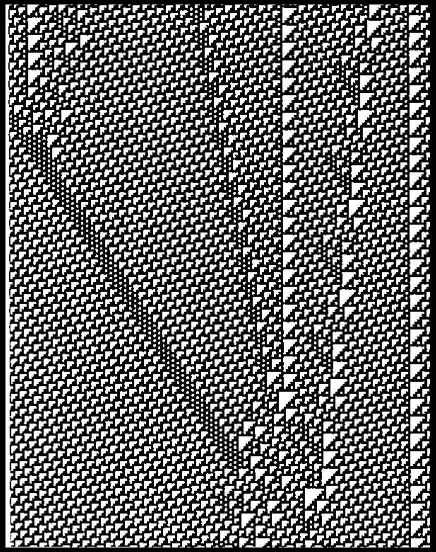


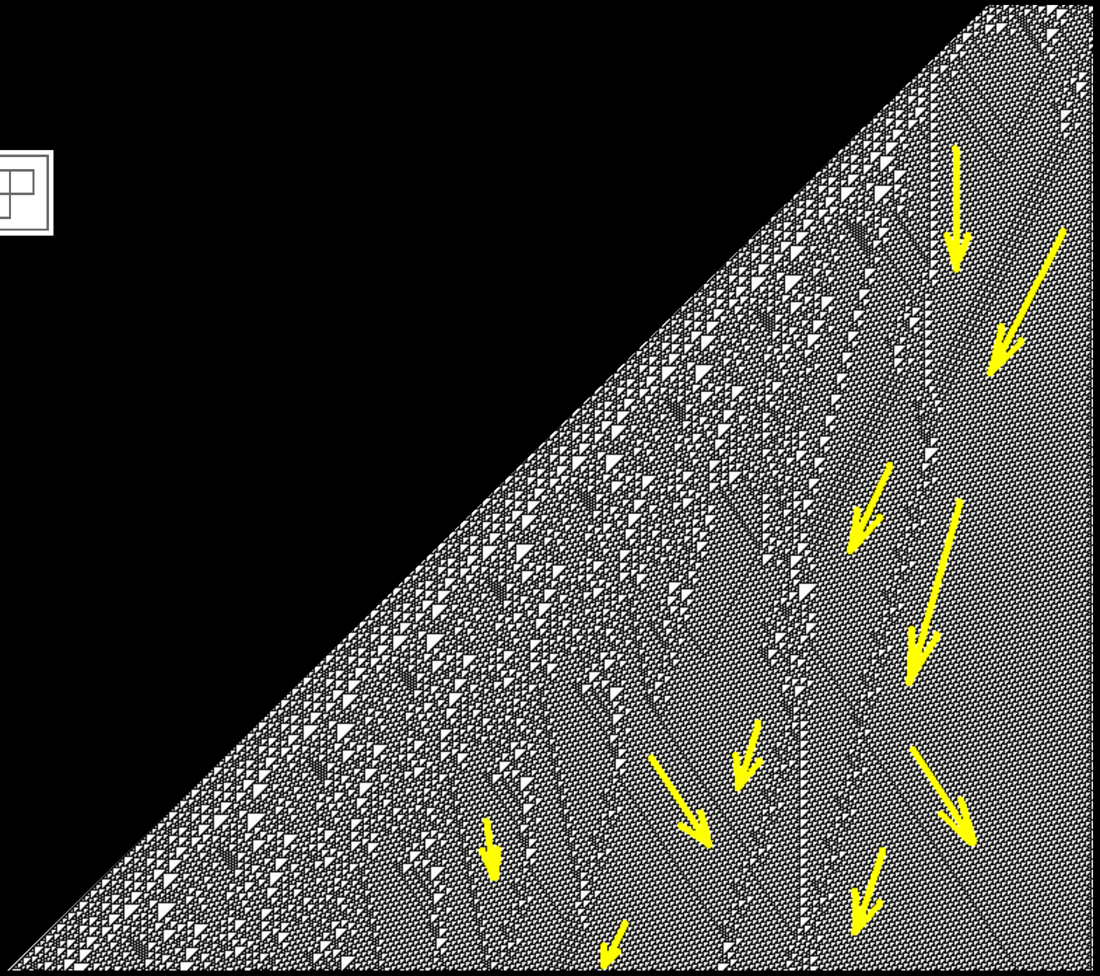
Klasyfikacja automatów jednowymiarowych

Klasa IV – „na granicy chaosu”, pojawiają się propagujące i oddziałujące ze sobą zlokalizowane struktury

reguła 110:





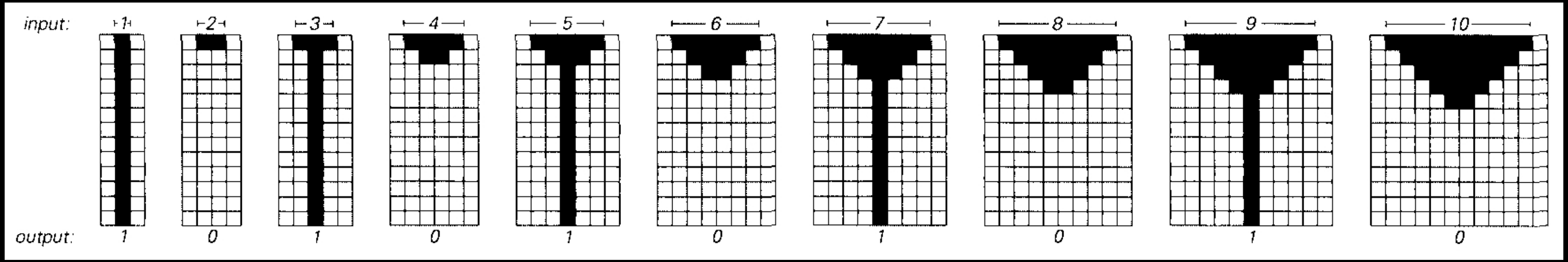


Obliczenia na automatach

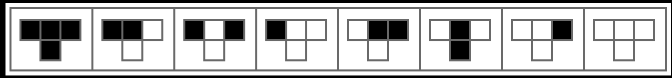
stan początkowy – input

stan końcowy – output

Przykład – sprawdzanie parzystości liczb:

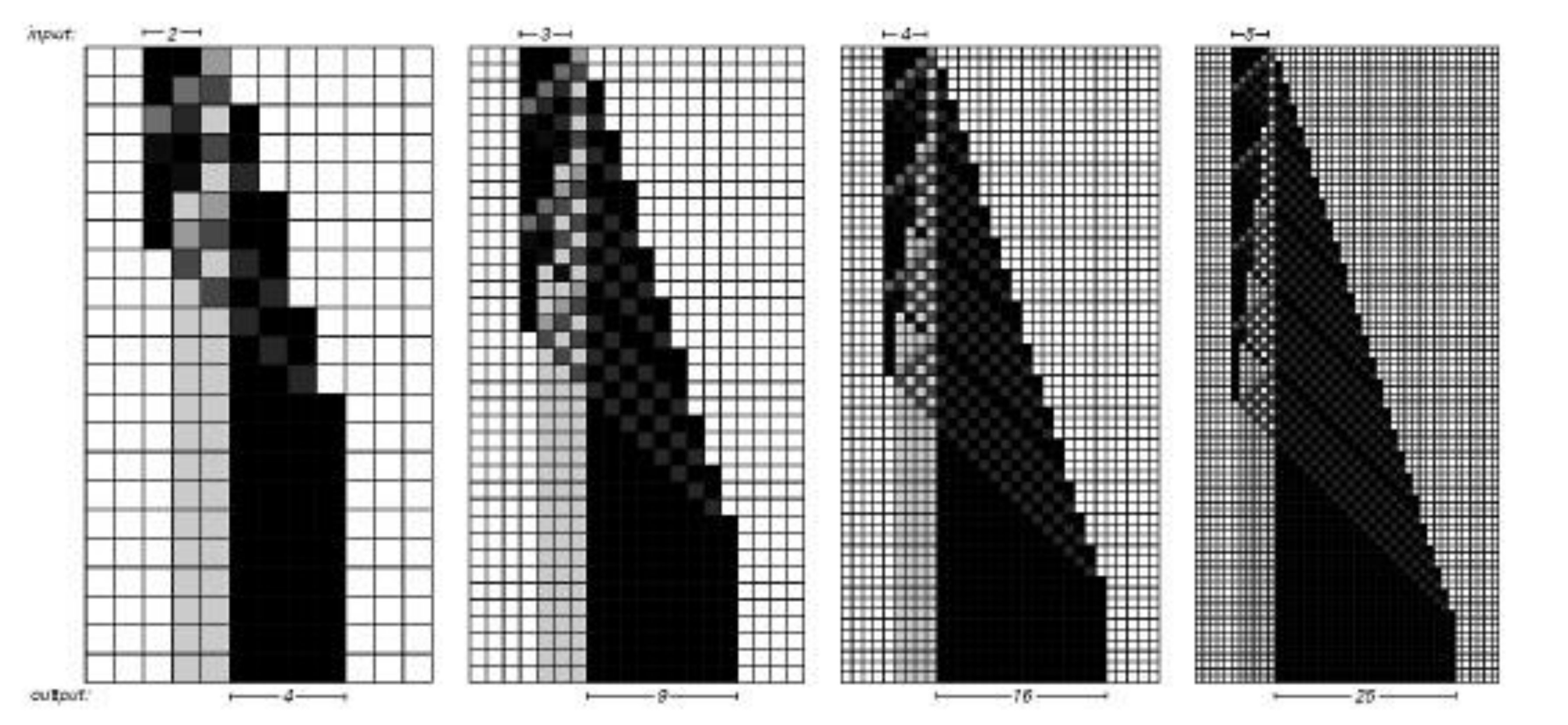


reguła 132:



Podnoszenie do kwadratu...

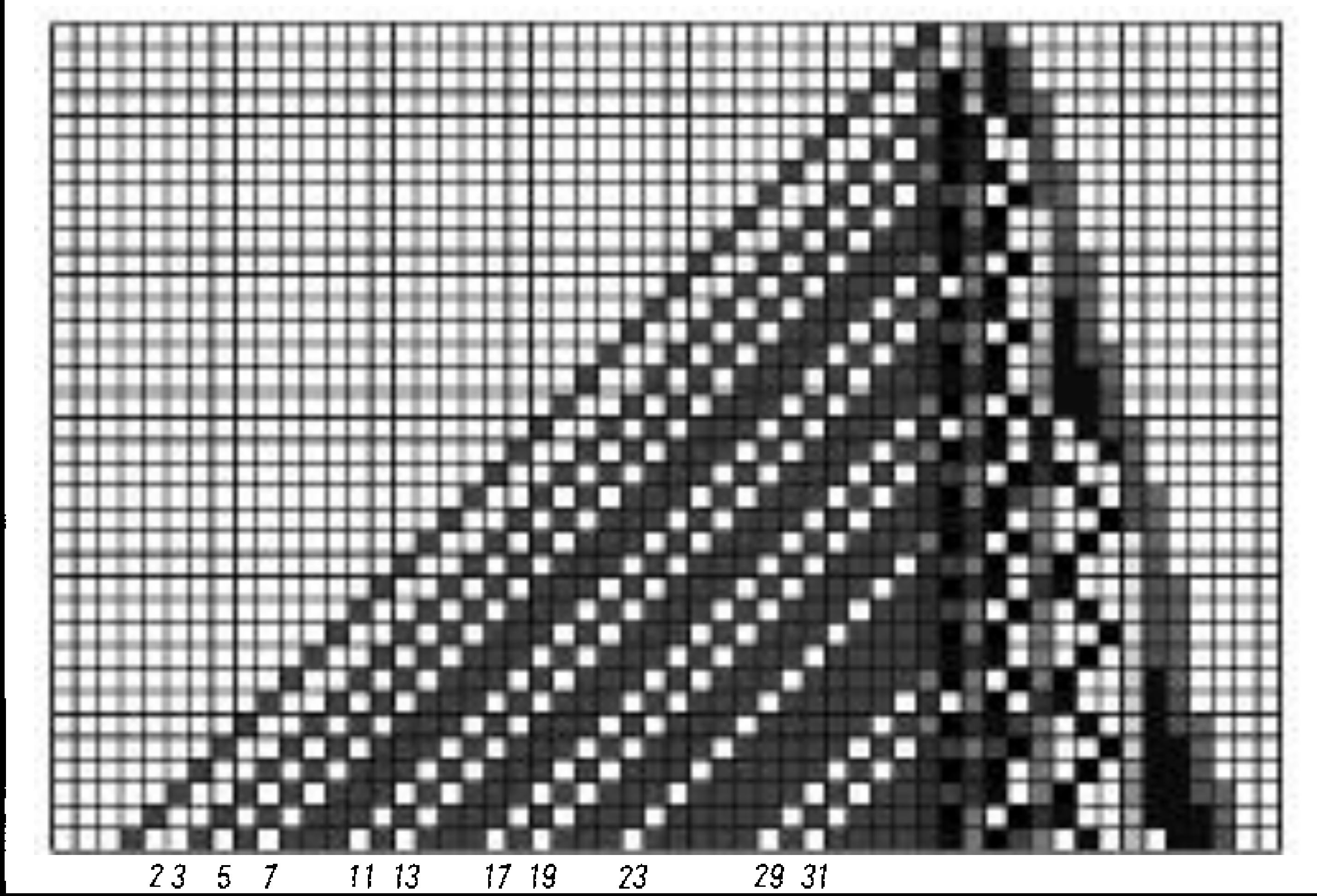
2 3 4 5



4 9...

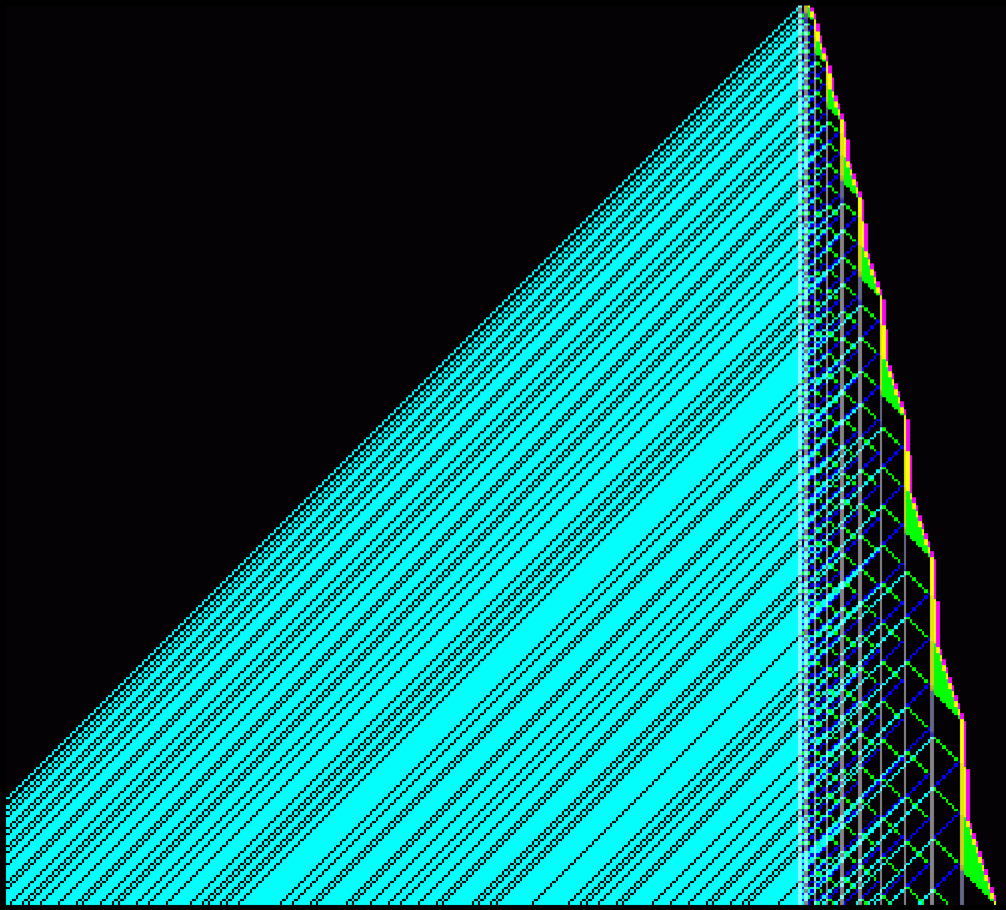
(8 kolorów na komórkę)

Wyliczanie liczb pierwszych



do 31...

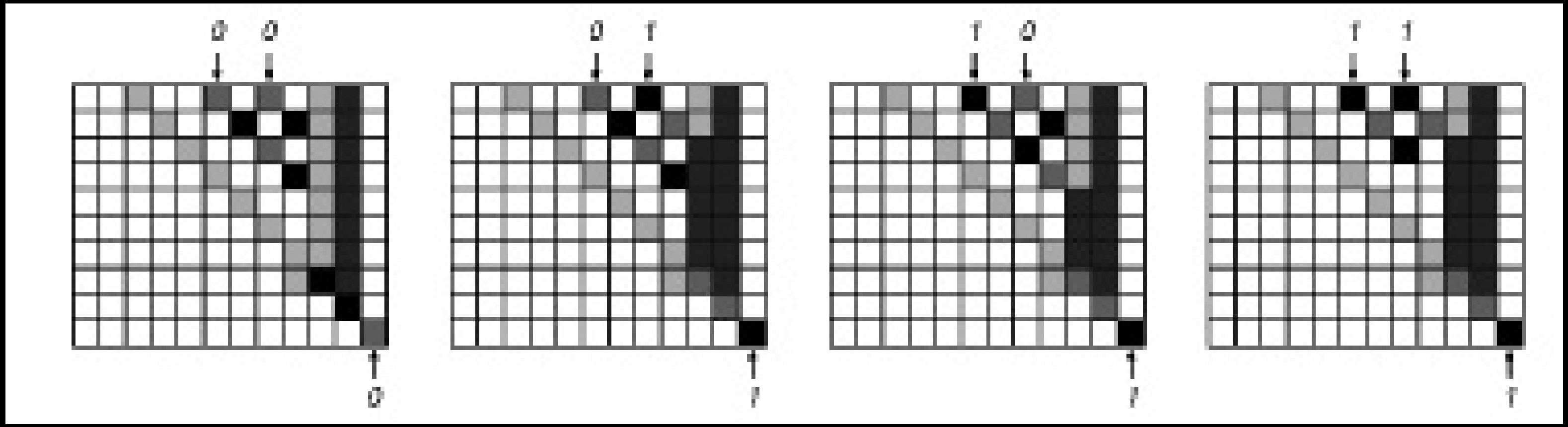
Wyliczanie liczb pierwszych



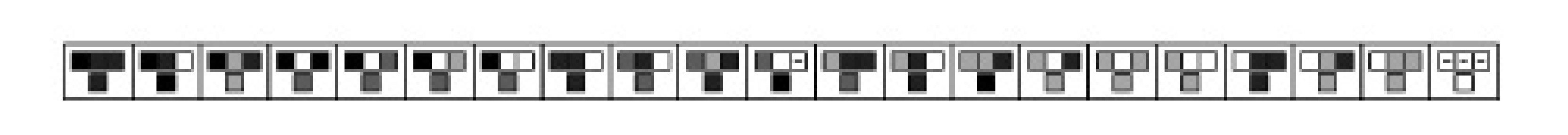
.. i do 821

Operacje logiczne

OR:



reguły (działają też dla innych operacji logicznych):

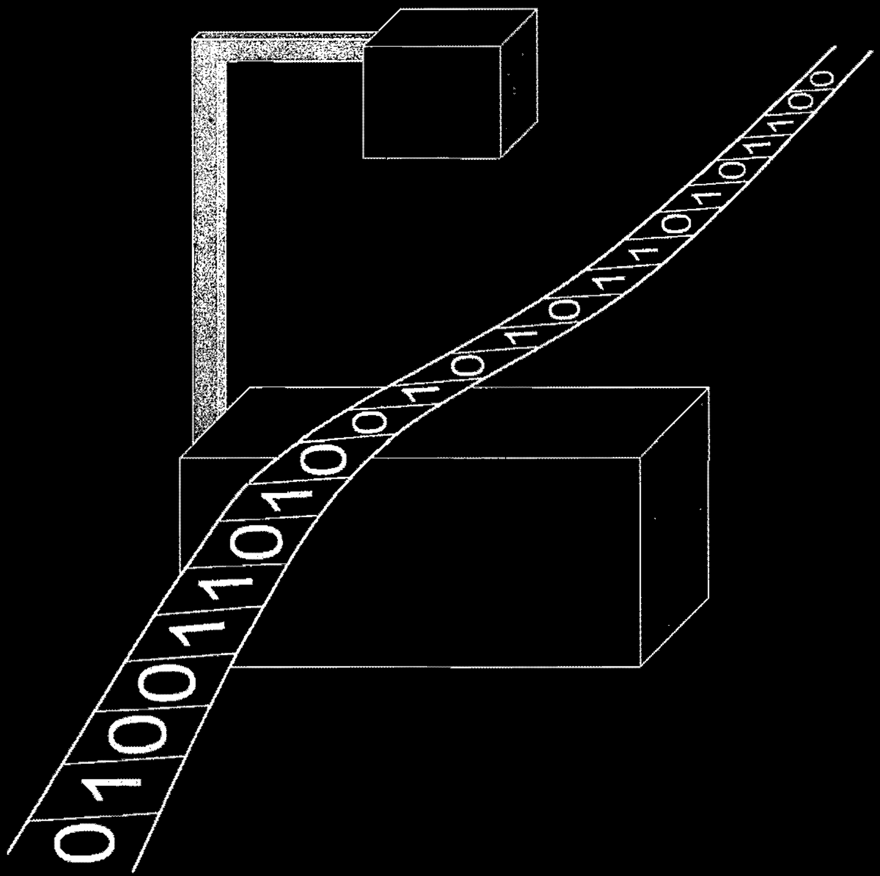


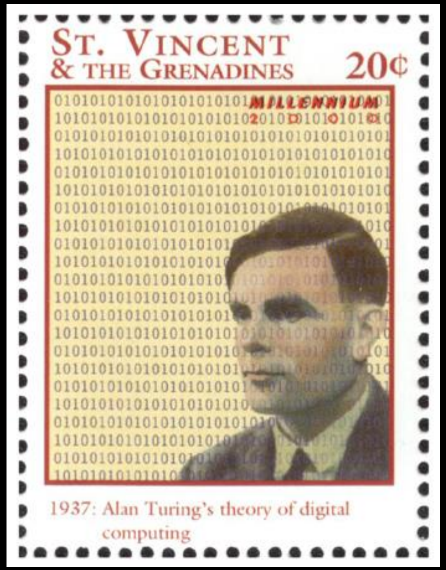
Uniwersalny komputer?

W powyższych przykładach do różnego rodzaju obliczeń używaliśmy różnych automatów.

Chciałoby się mieć jednak automat uniwersalny „do wszystkiego”.

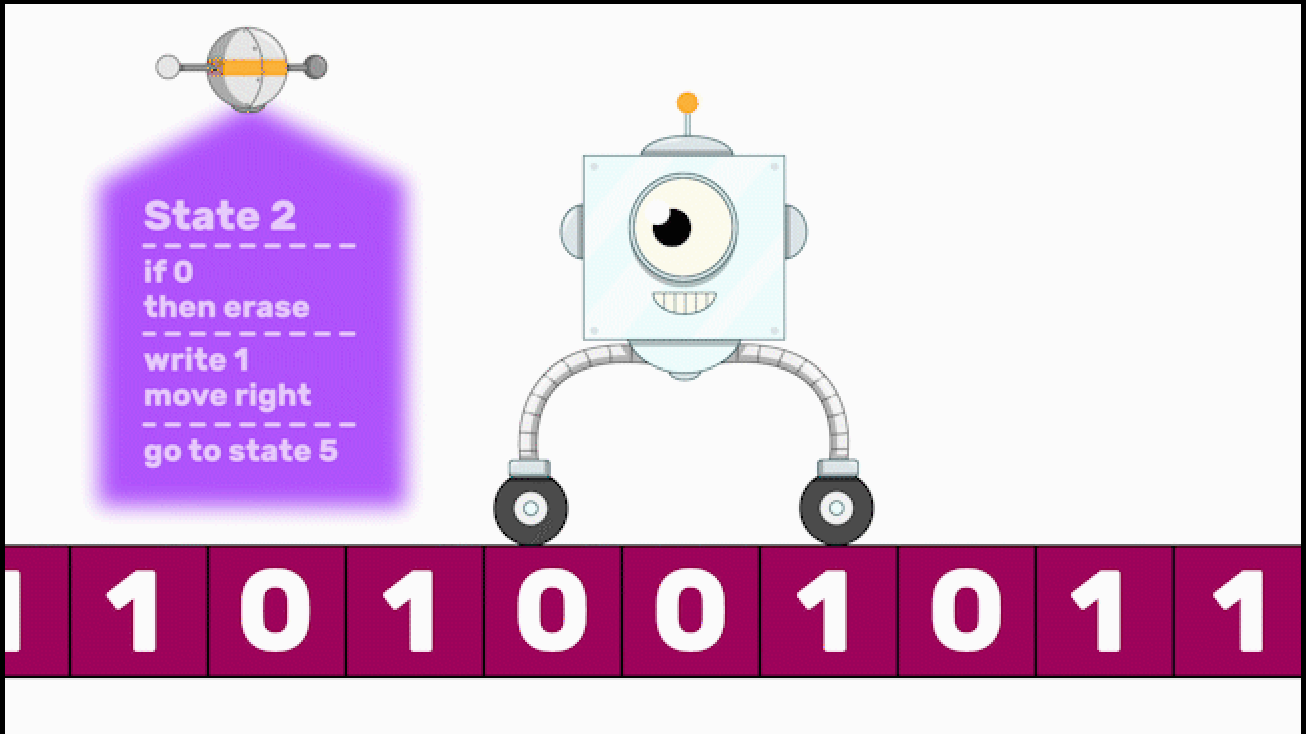
Maszyna Turinga





- głowica w jednym z M stanów
- taśma z pól w jednym z N stanów
- kombinacja stanów głowicy i pola może zapisać nową wartość w polu, zmienić stan głowicy i przesunąć taśmę o pole w prawo lub lewo

Maszyna Turinga



Problem stopu

Określić, czy maszyna Turinga z danym programem zatrzyma się po skończonej liczbie operacji czy nie.

Przykład:

problem Collatza

```
x:=x0
repeat
        if x jest parzyste then
            x := x/2
        else
         x := 3*x + 1
until x = 1
```

27, 82, 41, 124, 62, 31, 94, 47, 142, 71, 214, 107, 322, 161, 484, 242, 121, 364, 182, 91, 274, 137, 412, 206, 103, 310, 155, 466, 233, 700, 350, 175, 526, 263, 790, 395, 1186, 593, 1780, 890, 445, 1336, 668, 334, 167, 502, 251, 754, 377, 1132, 566, 283, 850, 425, 1276, 638, 319, 958, 479, 1438, 719, 2158, 1079, 3238, 1619, 4858, 2429, 7288, 3644, 1822, 911, 2734, 1367, 4102, 2051, 6154, 3077, 9232, 4616, 2308, 1154, 577, 1732, 866, 433, 1300, 650, 325, 976, 488, 244, 122, 61, 184, 92, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Church i Turing pokazali, że w ogólności problem stopu nie jest rozstrzygalny, co oznacza, że nie istnieje uniwersalny algorytm rozstrzygający o innych algorytmach, czy mają własność stopu.

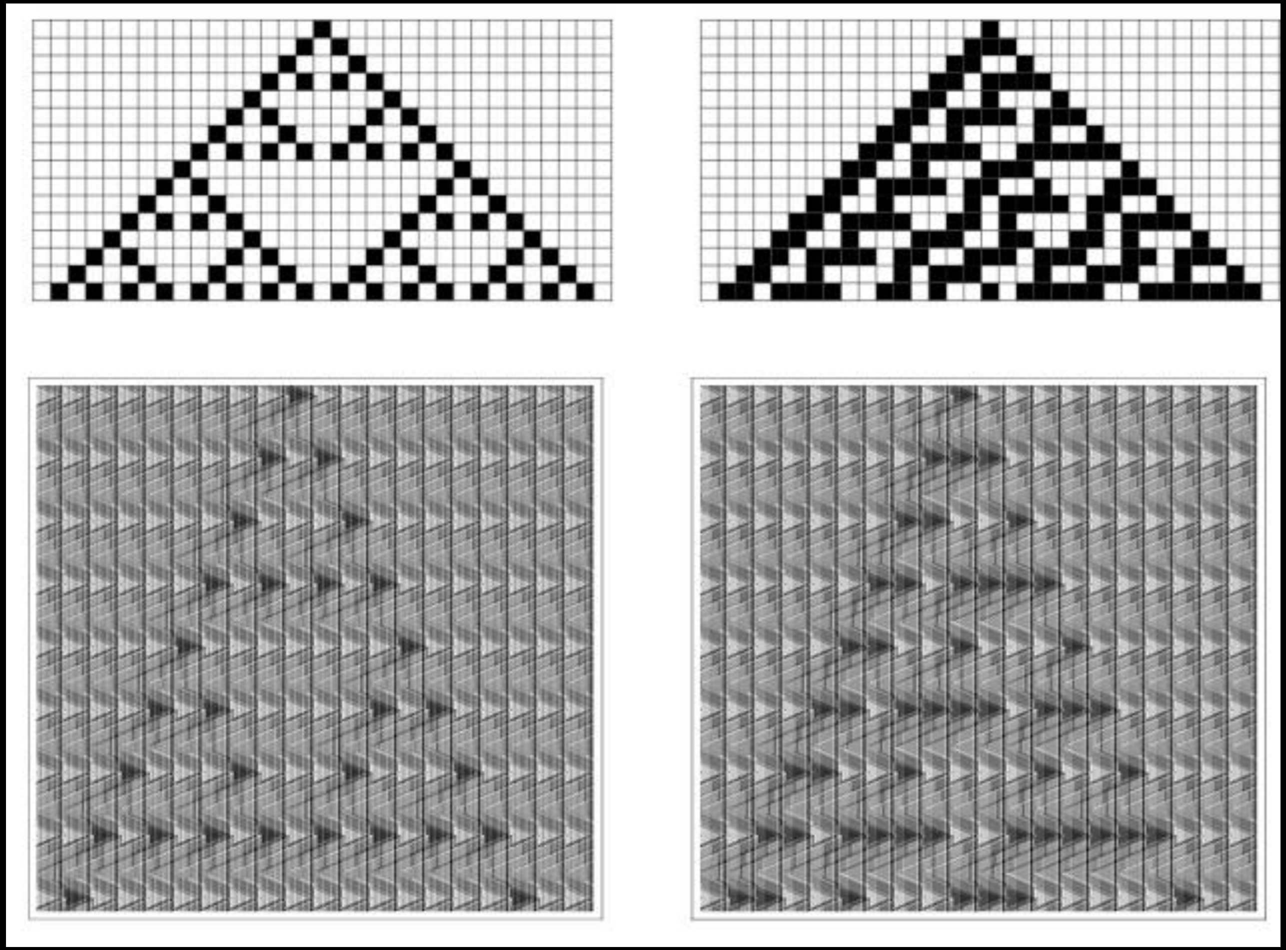
Uniwersalna maszyna Turinga

to taka maszyna Turinga, która jest w stanie emulować działanie dowolnej innej maszyny Turinga; można na niej obliczyć „wszystko” – za pomocą odpowiednio długiego programu....

"It is possible to invent a single machine which can be used to compute any computable sequence. If this machine **U** is supplied with a tape on the beginning of which is written the action table of some computing machine **M**, then **U** will compute the same sequence as **M**”

Turing, 1936

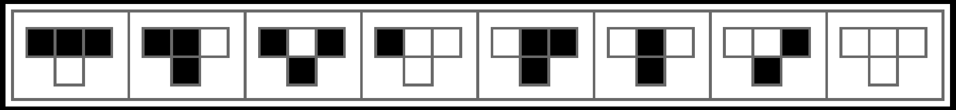
Uniwersalna maszyna Turinga

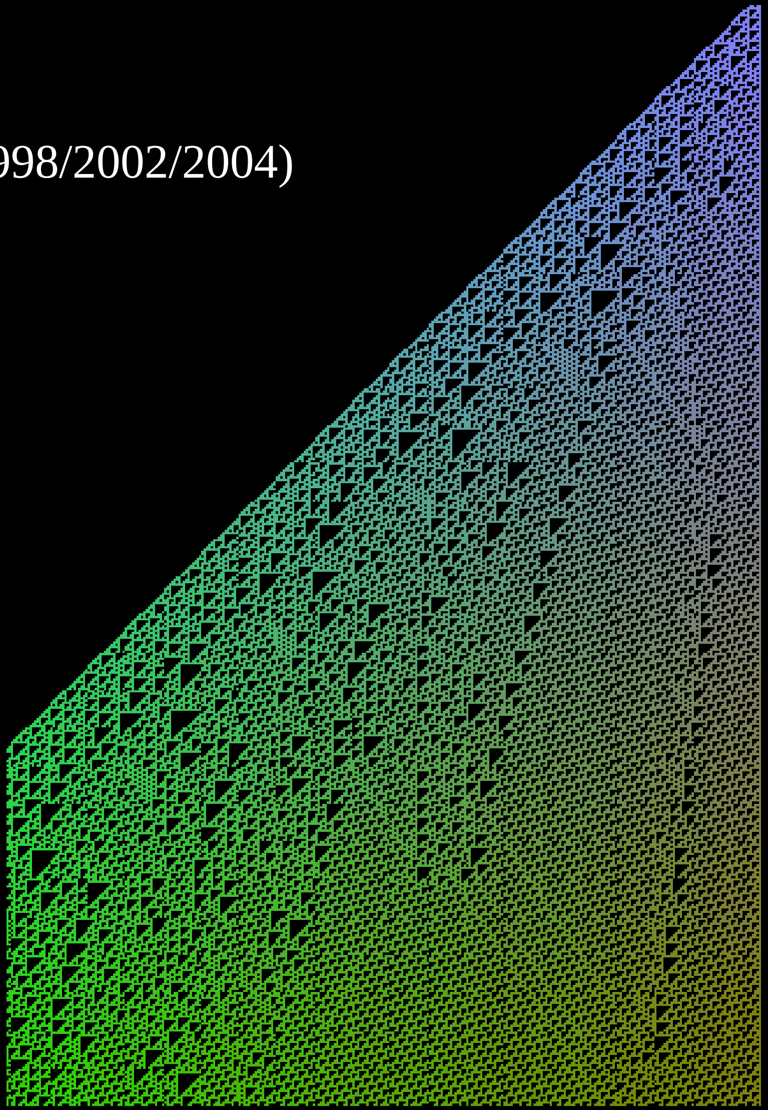


Automat 110 jest uniwersalną maszyną Turinga

Matthew Cook i Stephen Wolfram (1985/1994/1998/2002/2004)

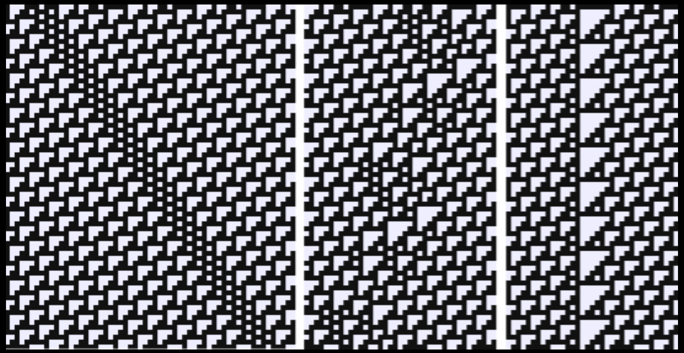
http://web.mit.edu/cfox/www/knitting/f.html



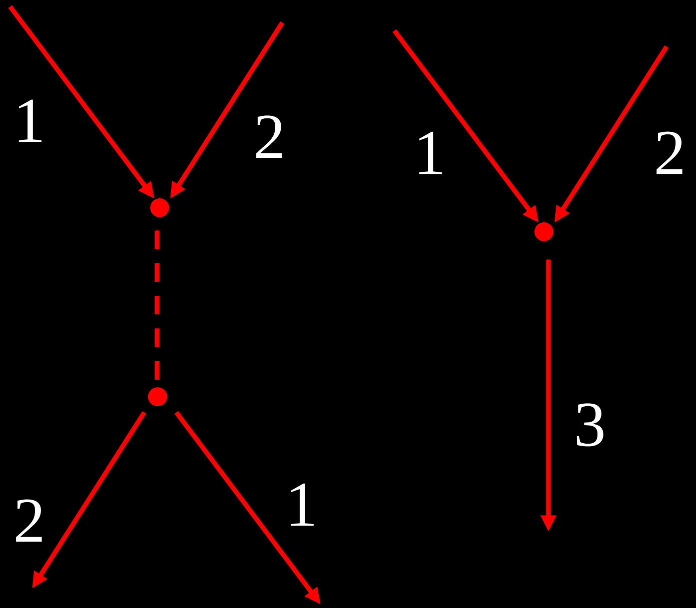


reguła 110 wykiełkowana z 11011

Ślizgacze

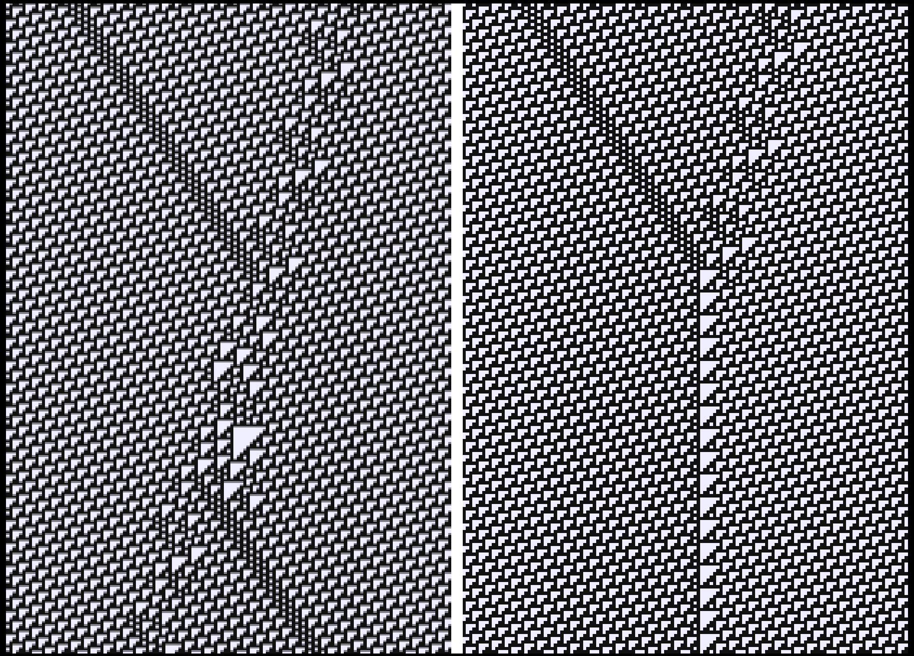


typ 1 typ 2 typ 3



obliczenia emulujące maszynę Turinga opierają się na oddziaływaniach między tymi strukturami

oddziaływanie:



Znaczenie uniwersalności

(według Wolframa)

- układy uniwersalne są w stanie emulować dowolne inne układy, łącznie z ich (dowolnie złożonymi) zachowaniami

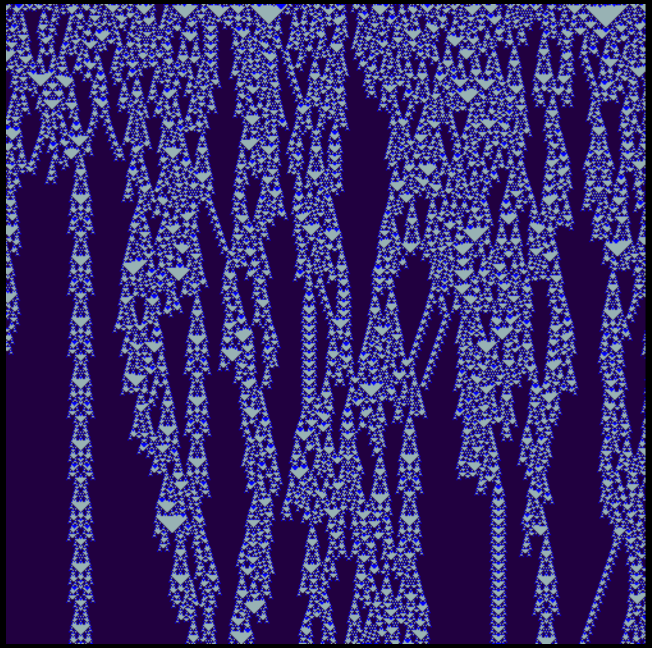
- dalsze komplikowanie ich własności nie prowadzi zatem do wzrostu złożoności

- można mówić o pewnym „progu złożoności”, w którym układ staje się uniwersalny

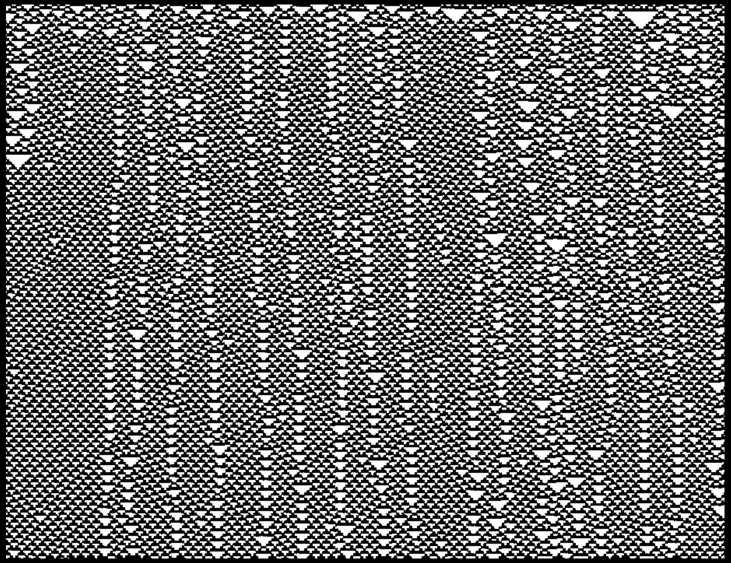
- okazuje się, że ten próg może być osiągany przez układy o bardzo prostych regułach (automat 110, gra w życie etc.)

Życie w życiu

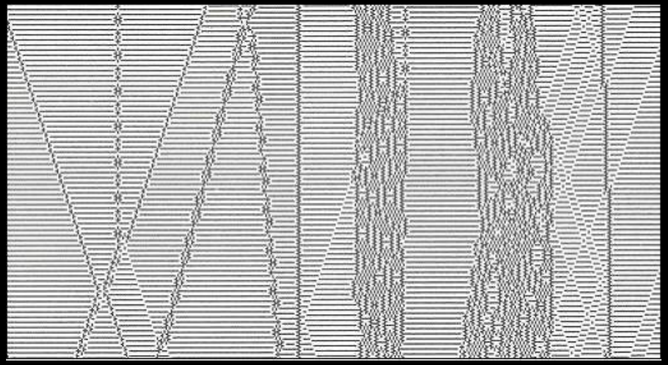
Inne przykłady automatów klasy IV



1815 (trójstanowy, sumacyjny)



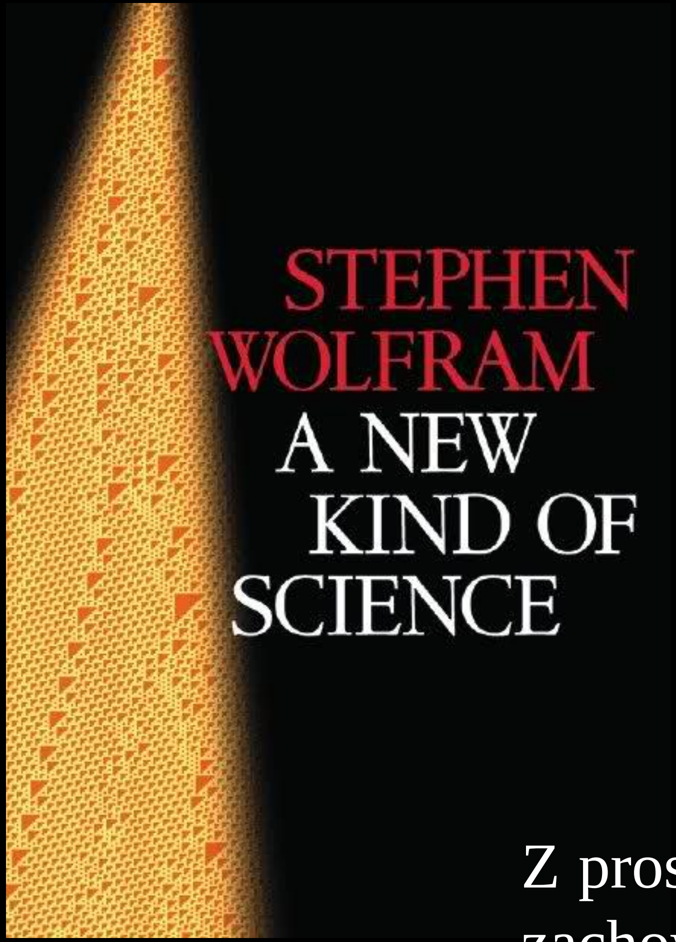
54



37R

niewykluczone, że też są uniwersalnymi maszynami Turinga

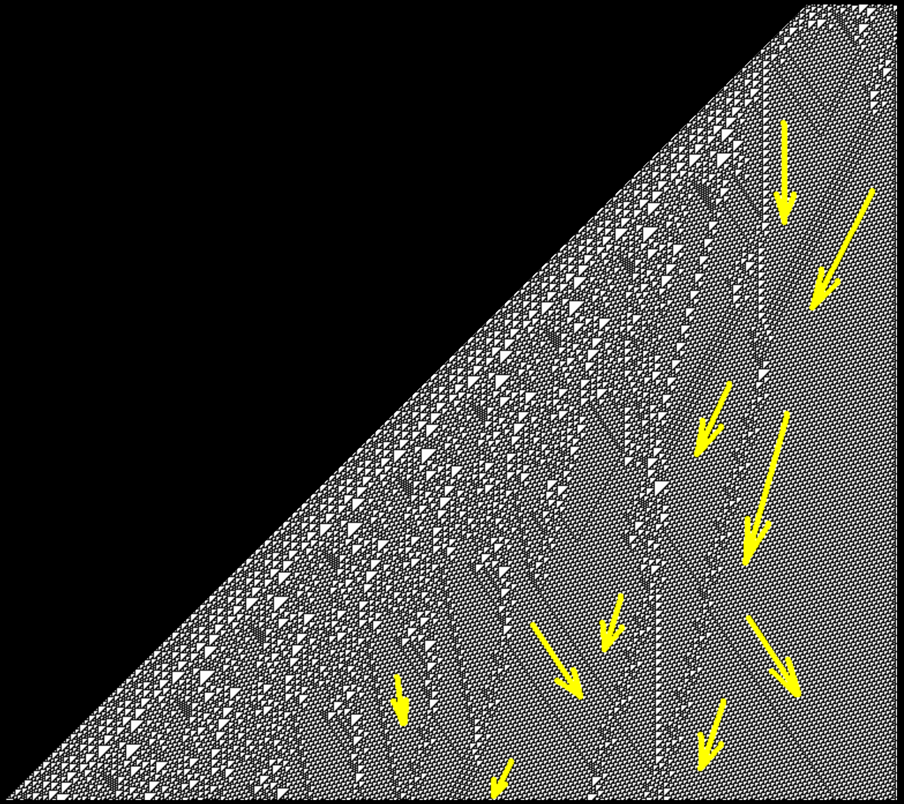
Nowa nauka?



Stephen Wolfram

Z prostych reguł wynika złożone, nieprzewidywalne zachowanie, a niemożność przewidzenia ewolucji jest własnością fundamentalną.

Miraż teorii wszystkiego

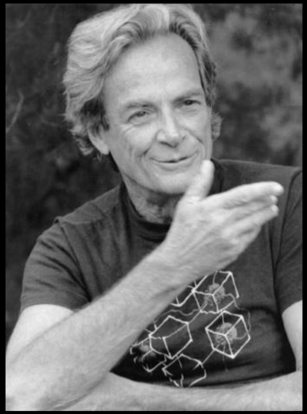


Pełen opis dynamiki układu pozwalający na przewidywanie jego ewolucji jest niemożliwy (bo rozwiązywałby problem stopu!)

Równania czy algorytmy?

Edward Fredkin

„Wszechświat jest komputerem a informacja jest bardziej podstawowa niż materia i energia”



Richard Feynman

„So I have often made the hypothesis that ultimately physics will not require a mathematical statement, that in the end the machinery will be revealed, and the laws will turn out to be simple, like the chequerboard with all its apparent complexities”

Konrad Zuse

„Wszechświat to jedno, wielkie, nieustanne obliczenie”

Wszechświat to „cztery linijki kodu”?

"How long do you envision this rule of the universe to be?"

"I'm guessing it's really very short."

"Like how long?"

"I don't know. In Mathematica, for example, perhaps three, four lines of code."

"Four lines of code?"

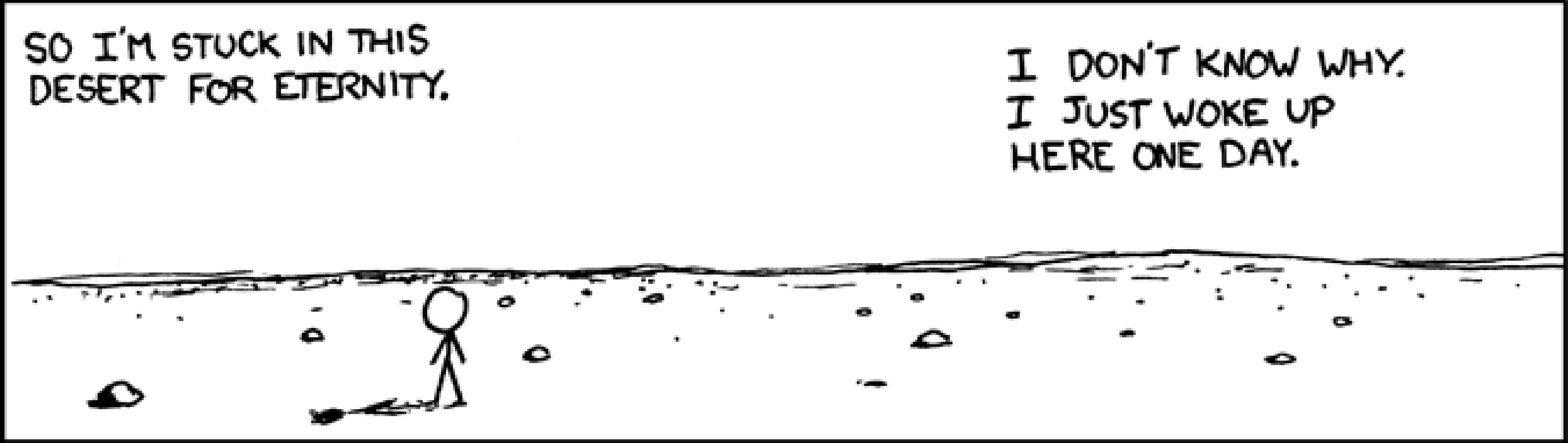
"That's what I'm guessing. (...) We're looking at a handful of lines of code."

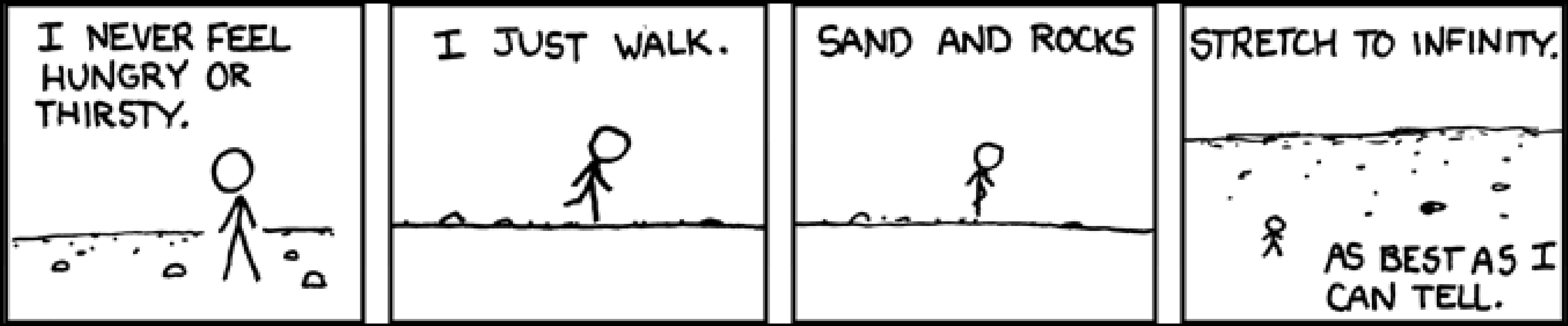
"So it's not like Windows?"

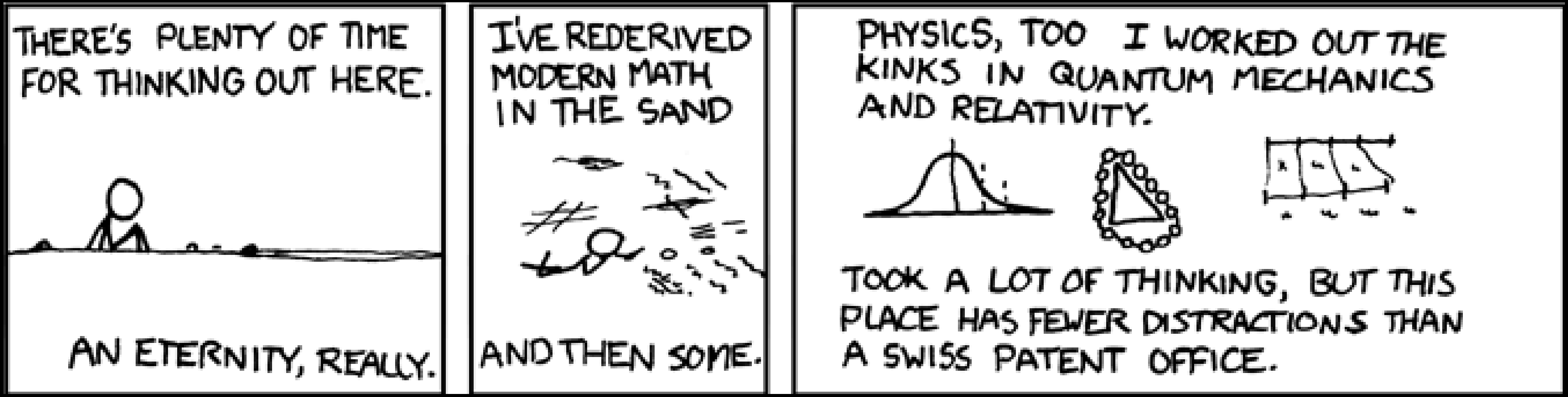
"No. It's not like Windows. It's going to be something small, I think."

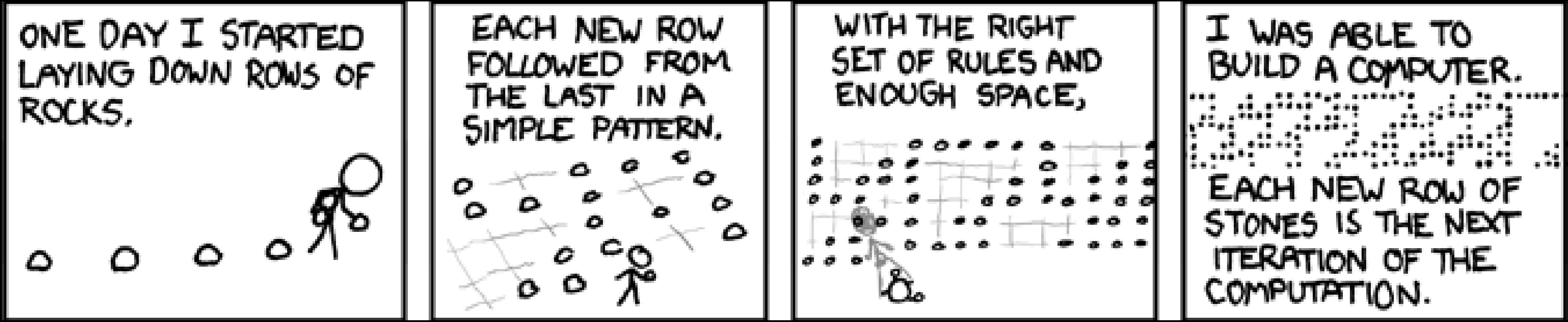
Stephen Wolfram, 2002

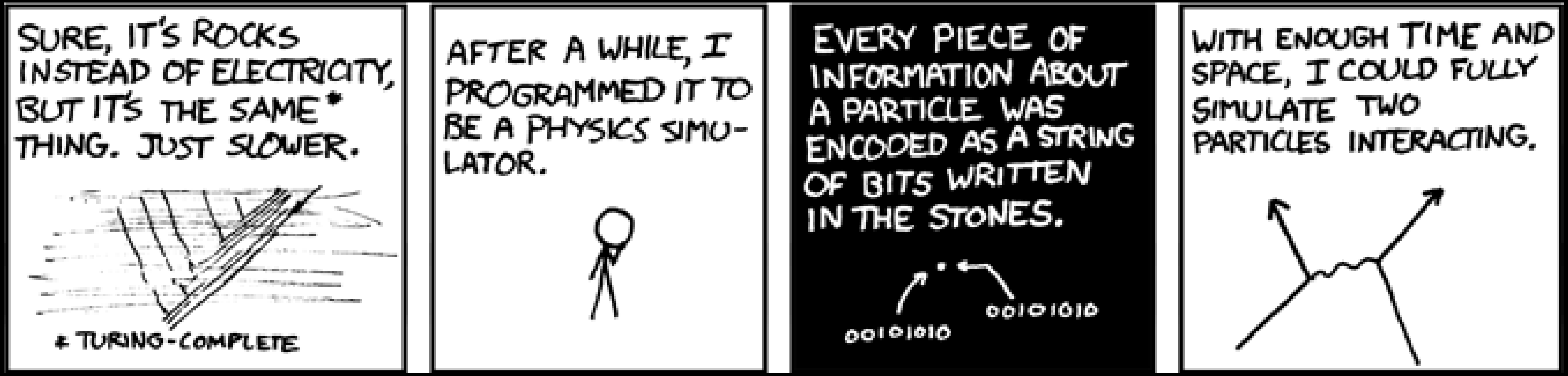
Dawno, dawno temu w odległej galaktyce...


SO I'M STUCK IN THIS DESERT FOR ETERNITY.
I DON'T KNOW WHY. I JUST WOKE UP HERE ONE DAY.

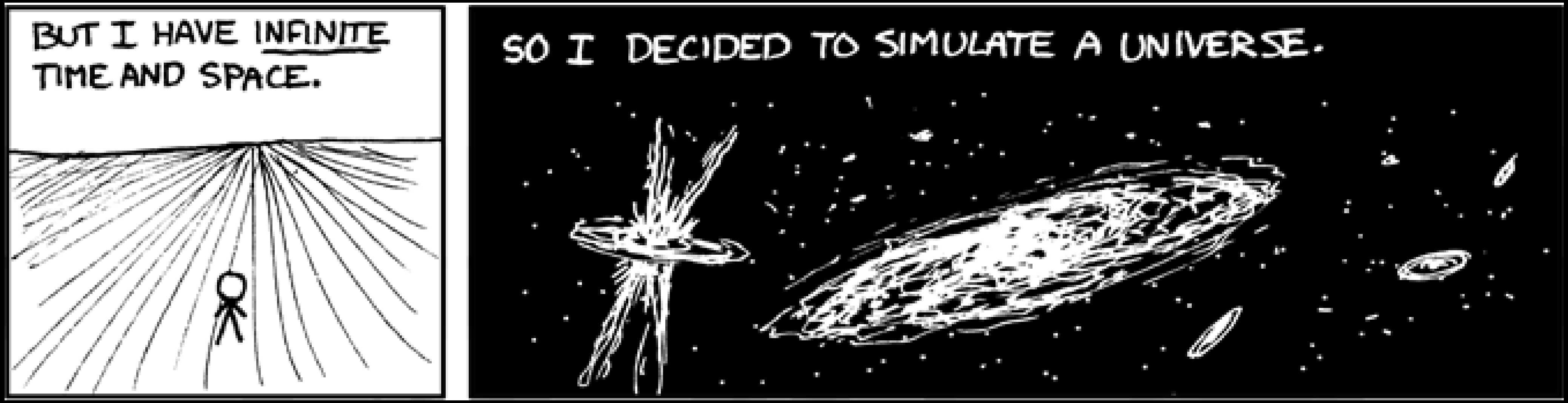

I NEVER FEEL HUNGRY OR THIRSTY.
I JUST WALK.
SAND AND ROCKS
STRETCH TO INFINITY.
AS BEST AS I CAN TELL.

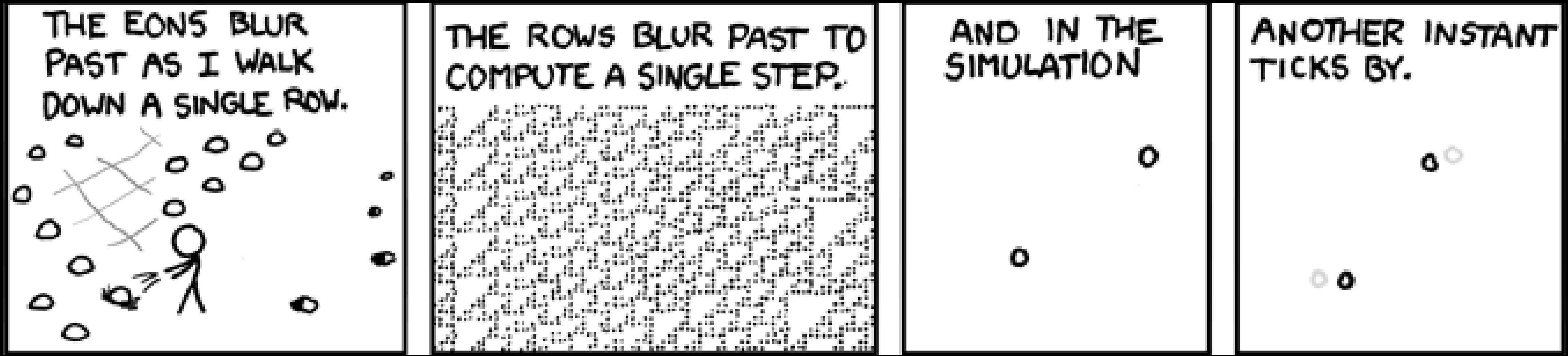

THERE'S PLENTY OF TIME FOR THINKING OUT HERE.
AN ETERNITY, REALLY.
I'VE REDERIVED MODERN MATH IN THE SAND
AND THEN SOME.
PHYSICS, TOO
I WORKED OUT THE KINKS IN QUANTUM MECHANICS AND RELATIVITY.
TOOK A LOT OF THINKING, BUT THIS PLACE HAS FEWER DISTRACTIONS THAN A SWISS PATENT OFFICE.

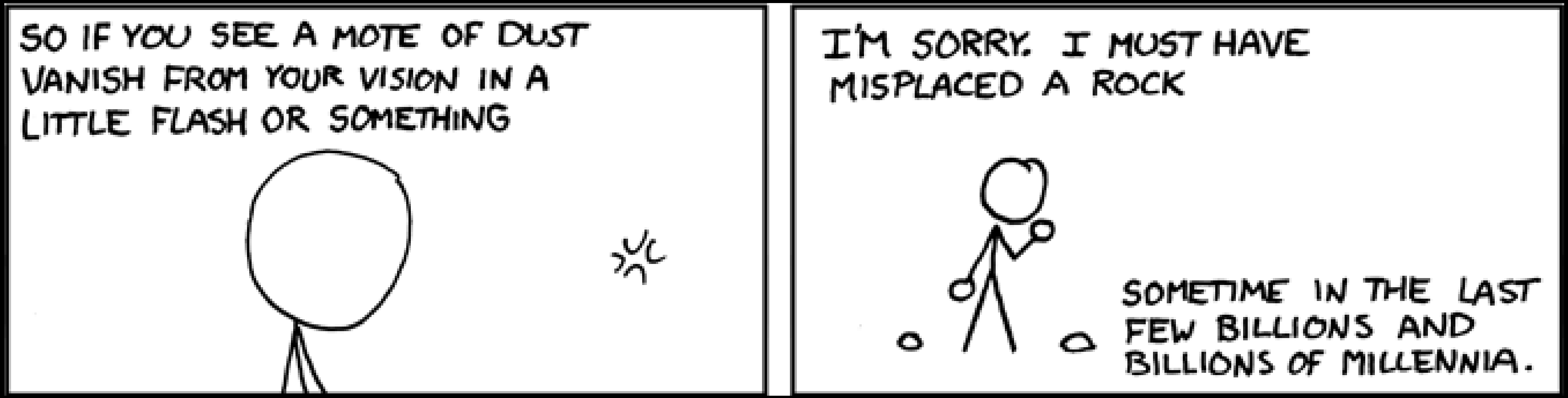

ONE DAY I STARTED LAYING DOWN ROWS OF ROCKS.
EACH NEW ROW FOLLOWED FROM THE LAST IN A SIMPLE PATTERN.
WITH THE RIGHT SET OF RULES AND ENOUGH SPACE,
I WAS ABLE TO BUILD A COMPUTER.
EACH NEW ROW OF STONES IS THE NEXT ITERATION OF THE COMPUTATION.



SURE, IT'S ROCKS INSTEAD OF ELECTRICITY, BUT IT'S THE SAME* THING. JUST SLOWER.
* TURING-COMPLETE
AFTER A WHILE, I PROGRAMMED IT TO BE A PHYSICS SIMULATOR.
EVERY PIECE OF INFORMATION ABOUT A PARTICLE WAS ENCODED AS A STRING OF BITS WRITTEN IN THE STONES.
00101010
00101010
WITH ENOUGH TIME AND SPACE, I COULD FULLY SIMULATE TWO PARTICLES INTERACTING.


BUT I HAVE INFINITE TIME AND SPACE.
SO I DECIDED TO SIMULATE A UNIVERSE.


THE EONS BLUR PAST AS I WALK DOWN A SINGLE ROW.
THE ROWS BLUR PAST TO COMPUTE A SINGLE STEP.
AND IN THE SIMULATION
ANOTHER INSTANT TICKS BY.


SO IF YOU SEE A MOTE OF DUST VANISH FROM YOUR VISION IN A LITTLE FLASH OR SOMETHING
I'M SORRY. I MUST HAVE MISPLACED A ROCK
SOMETIME IN THE LAST FEW BILLIONS AND BILLIONS OF MILLENNIA.

Computer modeling of physical phenomena

Piotr Szymczak, Jakub Tworzydło, Tomasz Szawełło

(wykład z ćwiczeniami w semestrze letnim, po angielsku, 4h/tydz., 6 ECTS)

- Complex systems: cellular automata, self-organized criticality, genetic algorithms, trail formation
- Cellular automata in physics: lattice-Boltzmann methods, fluid flow simulations
- Nonlinear dynamics: chaotic systems, solitons
- Networks and interactions: neural networks and Hopfield model, network growth: from river networks to social interactions, game theory