

# *Intelligenza Artificiale II*

## *Automati cellulari*

Marco Piastra

*“The presentation of Conway's construction by Martin Gardner in the October 1970 issue of Scientific American made it so famous that, in 1974, Time magazine even complained about how much computer time could be wasted because ‘growing hordes of fanatics’ spent their office days playing with the new ‘toy’”*

[Rennard, 2002]

# Game of Life (J.H. Conway, 1970)

- **Universo**

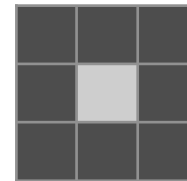
  - Reticolo ortogonale infinito di celle quadrate
  - Le celle sono tutte uguali (universo omogeneo)

- **Stati**

  - Ciascuna cella può essere viva o morta (1 o 0)

- **Intorno (di una cella)**

  - Ciascuna cella ha un intorno formato dalle otto celle contigue



- **Dinamica**

  - L'universo si evolve nel tempo, a partire da una configurazione iniziale

  - Evoluzione parallela:

  - Ciascuna cella decide il proprio stato futuro in base al proprio stato ed allo stato delle celle nel suo intorno

  - Tempo discreto:

  - Ad ogni istante, ciascuna cella dell'universo decide quale stato avrà nell'istante successivo

# Game of Life (J.H. Conway, 1970)

- **Universo**

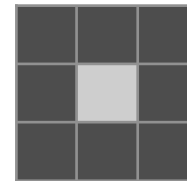
  - Reticolo ortogonale infinito di celle quadrate
  - Le celle sono tutte uguali (universo omogeneo)

- **Stati**

  - Ciascuna cella può essere viva o morta (1 o 0)

- **Intorno (di una cella)**

  - Ciascuna cella ha un intorno formato dalle otto celle contigue



- **Dinamica**

  - Regole di transizione:

- 1) Una cella viva muore se l'intorno contiene meno di due celle vive (*solitudine*)
- 2) Una cella viva muore se l'intorno contiene più di tre celle vive (*sovrappopolazione*)
- 3) Una cella viva rimane viva se l'intorno contiene due o tre celle vive (*mutuo sostegno*)
- 4) Una cella morta con tre celle vive nel suo intorno diventa viva (*genesì*)

# Game of Life: configurazioni e dinamica

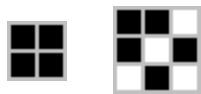
*L'aspetto emergente è il comportamento della popolazione di celle vive*

- **Configurazioni iniziali**

Il comportamento è completamente determinato dalla configurazione iniziale

- **Comportamenti possibili** (vedi dimostrazione con Golly)

Nessuna variazione



Comportamento periodico



Altri comportamenti ....



# *Game of Life: configurazioni e dinamica*

*L'aspetto emergente è il comportamento della popolazione di celle*

- **Configurazioni iniziali**

Il comportamento è completamente determinato dalla configurazione iniziale

- **Altre categorie (informali)**

*Still lifes*

Pattern statici o quasi statici

*Oscillators*

Pattern che si riproducono dopo un certo periodo di tempo

*Gliders*

Pattern che si 'spostano' nell'universo

*Guns*

Pattern che 'emettono' *gliders*

# Interludio: *Billiard Ball Model* (BBM) (Fredkin & Toffoli, 1992)

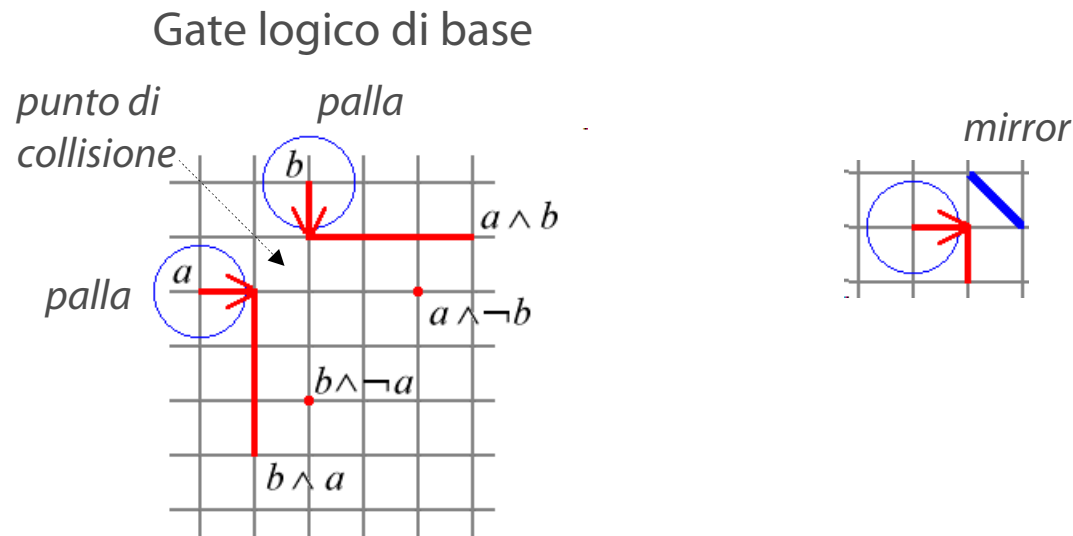
- *Una macchina di Turing con le palle da biliardo*

Le palle, tutte identiche, si muovono su una griglia rettangolare, alla stessa velocità

Le palle sono disposte in modo da urtarsi solo per angoli retti

Le collisioni sono elastiche: non si perde energia

Si possono utilizzare dei 'mirror' per guidare il percorso delle palle



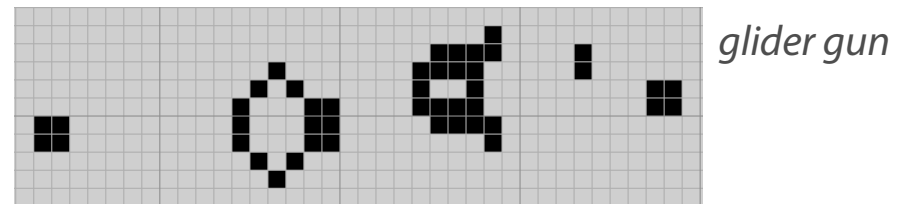
# Gate logici nel *Game of Life*

- Componenti di base

Segnali: *glider*



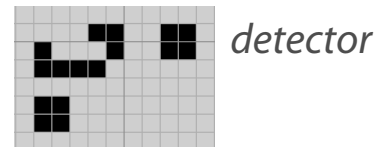
Generatori di segnale: *glider guns*



Interruttori: *stopper*



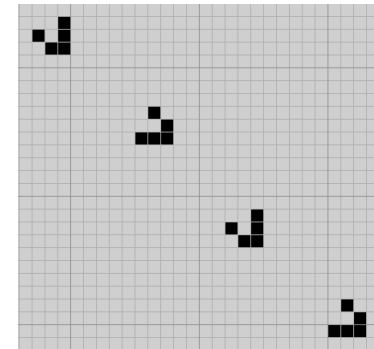
Output: *detector*



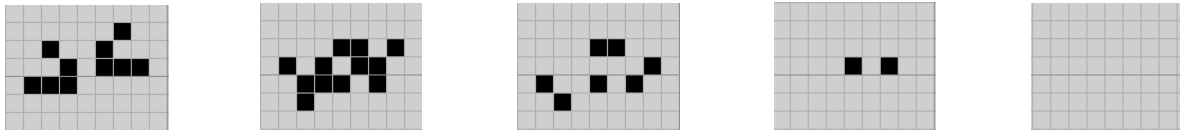
# Gate logici nel *Game of Life*

- Meccanismi di base

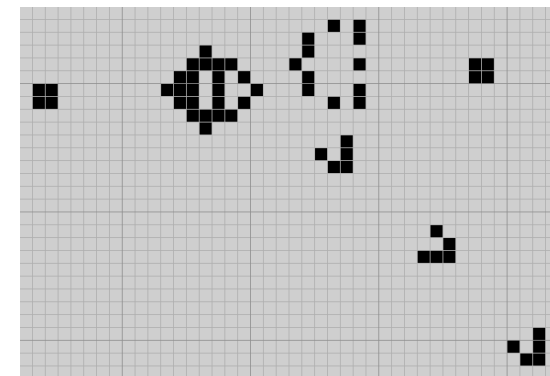
I *glider* si muovono nello spazio seguendo una traiettoria lineare



Nelle collisioni si possono 'annullare' a vicenda



I *glider gun* emettono *glider* in continuazione

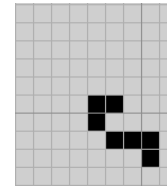
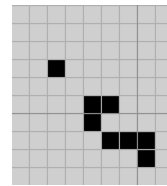
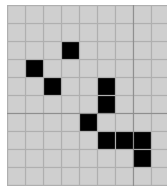
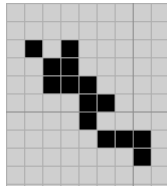
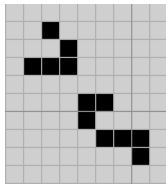




# Gate logici nel *Game of Life*

- Meccanismi di base

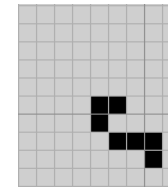
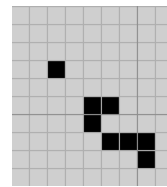
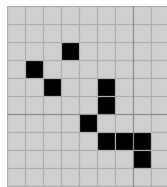
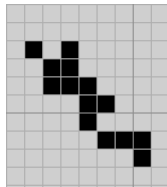
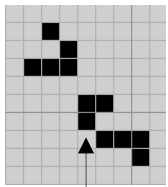
*Gli stopper 'assorbono' i glider*



# Gate logici nel *Game of Life*

- Meccanismi di base

*Gli stopper 'assorbono' i glider*

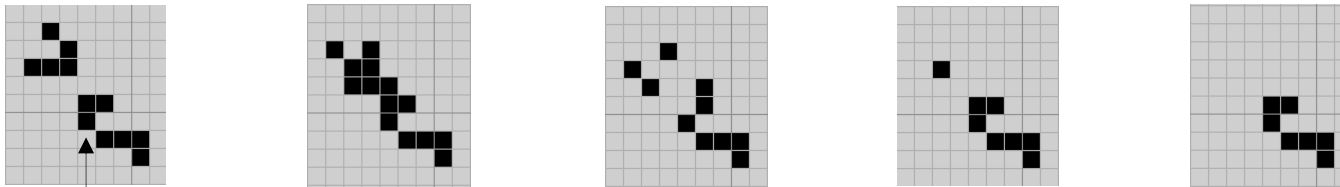


*Ma se questa cella è attiva anche lo stopper si annulla (switch)*

# Gate logici nel *Game of Life*

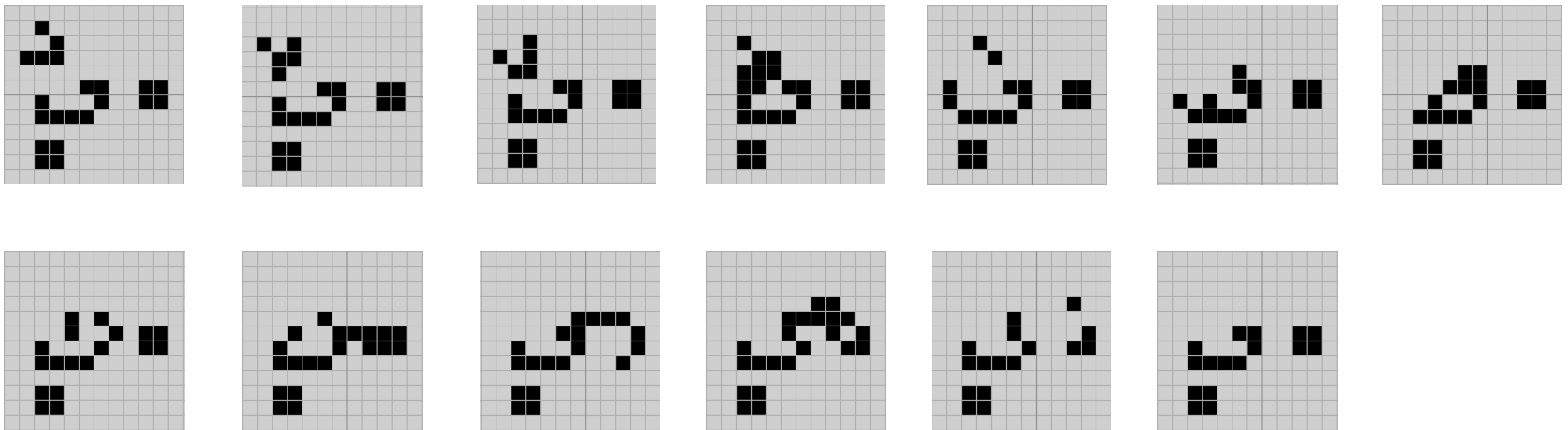
- Meccanismi di base

Gli *stopper* 'assorbono' i *glider*



Ma se questa cella è attiva anche lo stopper si annulla (*switch*)

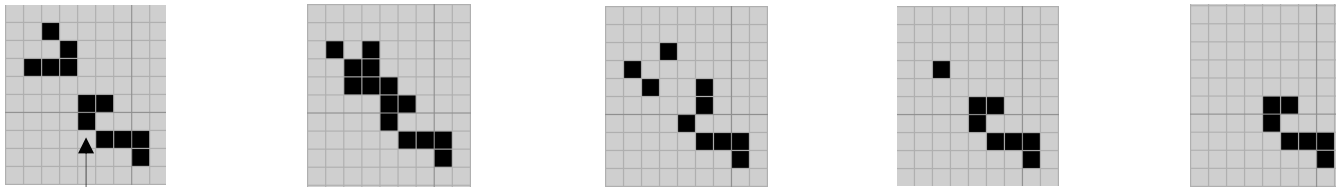
I *detector* 'assorbono' i *glider* e cambiano temporaneamente stato



# Gate logici nel *Game of Life*

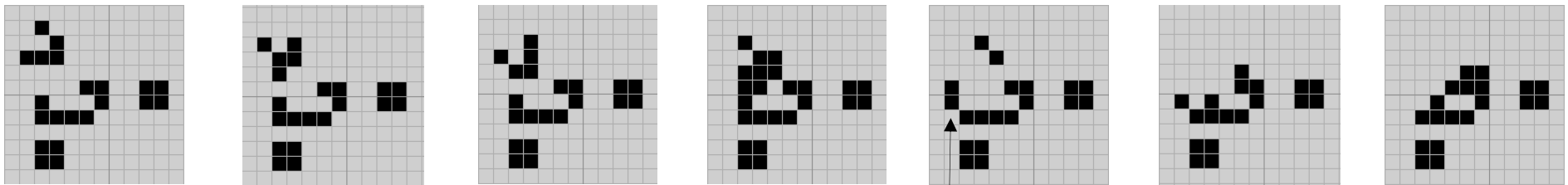
- Meccanismi di base

Gli *stopper* 'assorbono' i *glider*

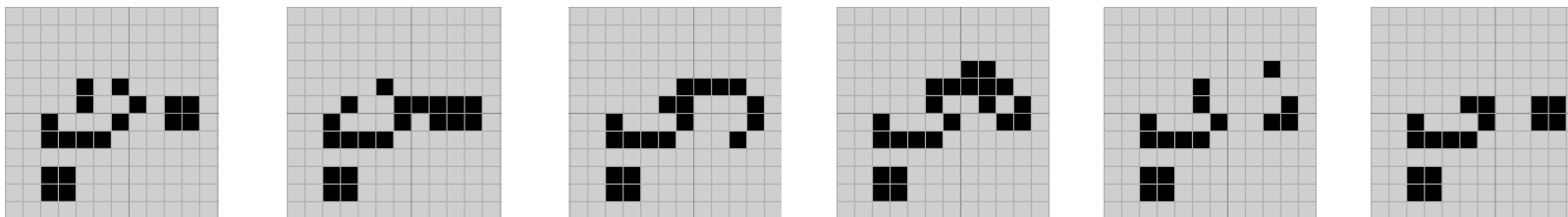


Ma se questa cella è attiva anche lo stopper si annulla (switch)

I *detector* 'assorbono' i *glider* e cambiano temporaneamente stato



Questa cella si attiva temporaneamente (output)

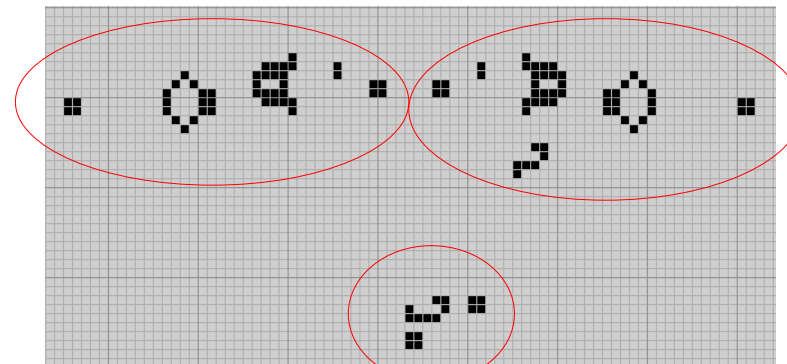


# Gate logici nel *Game of Life*

- Gate NOT

Due *glider gun*, uno *stopper*, un *detector*

Generatore di segnale  
(costante)



Input A

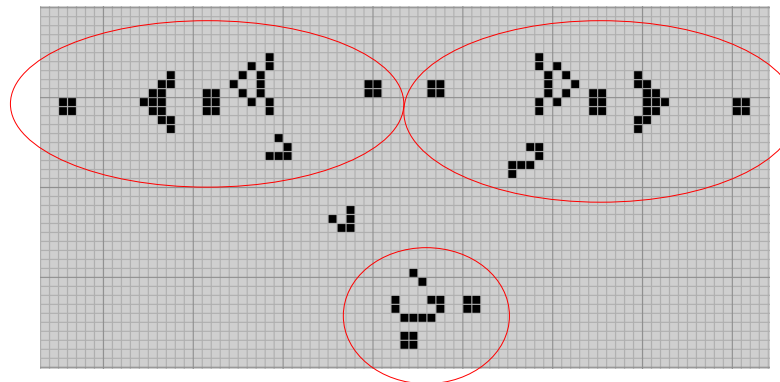
Output NOT A

# Gate logici nel *Game of Life*

- Gate NOT

Due *glider gun*, uno *stopper*, un *detector*

Generatore di segnale  
(costante)



Input A: lo stopper è chiuso  
il valore è FALSE

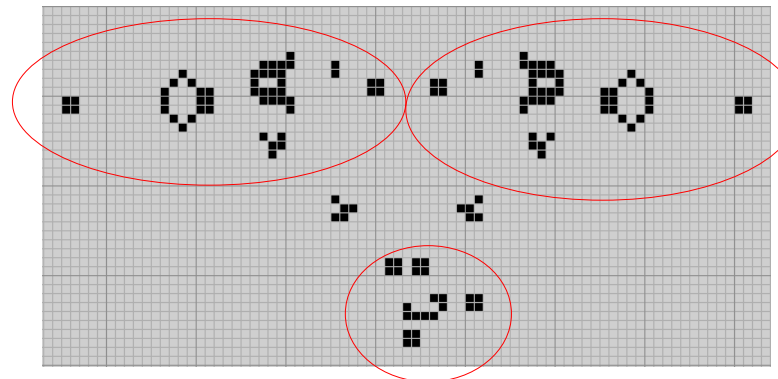
Output NOT A: i segnali del generatore arrivano  
il valore è TRUE

# Gate logici nel *Game of Life*

- Gate NOT

Due *glider gun*, uno *stopper*, un *detector*

Generatore di segnale  
(costante)



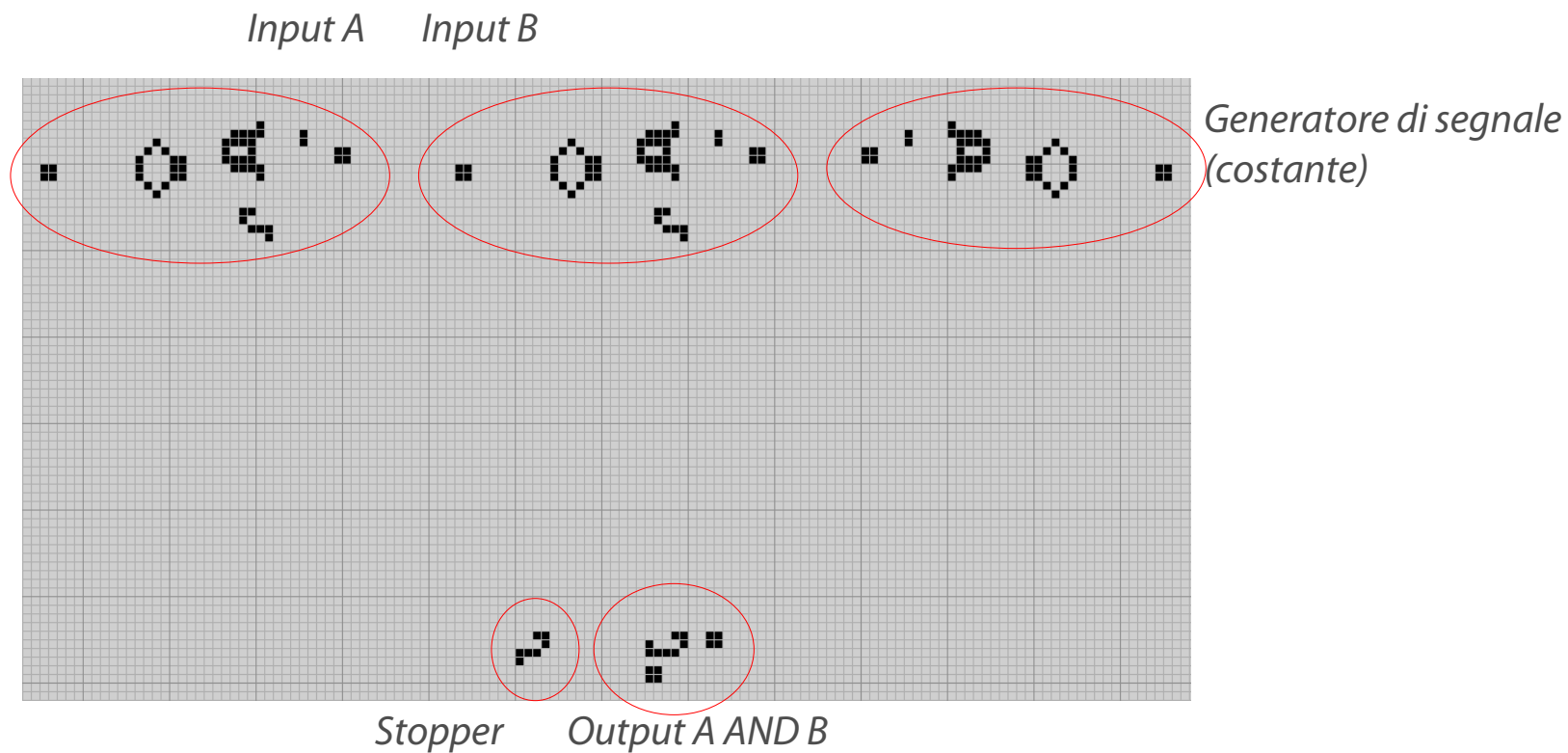
Input A: lo stopper è aperto  
il valore è TRUE

Output NOT A: i segnali dell'input  
annullano quelli del generatore  
il valore è FALSE

# Gate logici nel *Game of Life*

- Gate AND

Tre *glider gun*, tre *stopper*, un *detector*





# Gate logici nel *Game of Life*

## ■ Gate AND

Tre *glider gun*, tre *stopper*, un *detector*

Input A: lo stopper è chiuso  
il valore è FALSE

Input B: lo stopper è chiuso  
il valore è FALSE



Generatore di segnale  
(costante)

Stopper: annulla i segnali del generatore  
Output A AND B: non arriva segnale  
il valore è FALSE

# Gate logici nel *Game of Life*

## ■ Gate AND

Tre *glider gun*, tre *stopper*, un *detector*

Input A: lo stopper è aperto  
il valore è TRUE

Input B: lo stopper è chiuso  
il valore è FALSE



Generatore di segnale  
(costante)

Stopper: Output A AND B: non arriva segnale  
è inattivo il valore è FALSE

# Gate logici nel *Game of Life*

## ■ Gate AND

Tre *glider gun*, tre *stopper*, un *detector*

Input A: lo stopper è chiuso  
il valore è FALSE

Input B: lo stopper è aperto  
il valore è TRUE



Generatore di segnale  
(costante)

Stopper: Output A AND B: non arriva segnale  
è inattivo il valore è FALSE

# Gate logici nel *Game of Life*

## ■ Gate AND

Tre *glider gun*, tre *stopper*, un *detector*

Input A: lo stopper è aperto  
il valore è TRUE

Input B: lo stopper è aperto  
il valore è TRUE



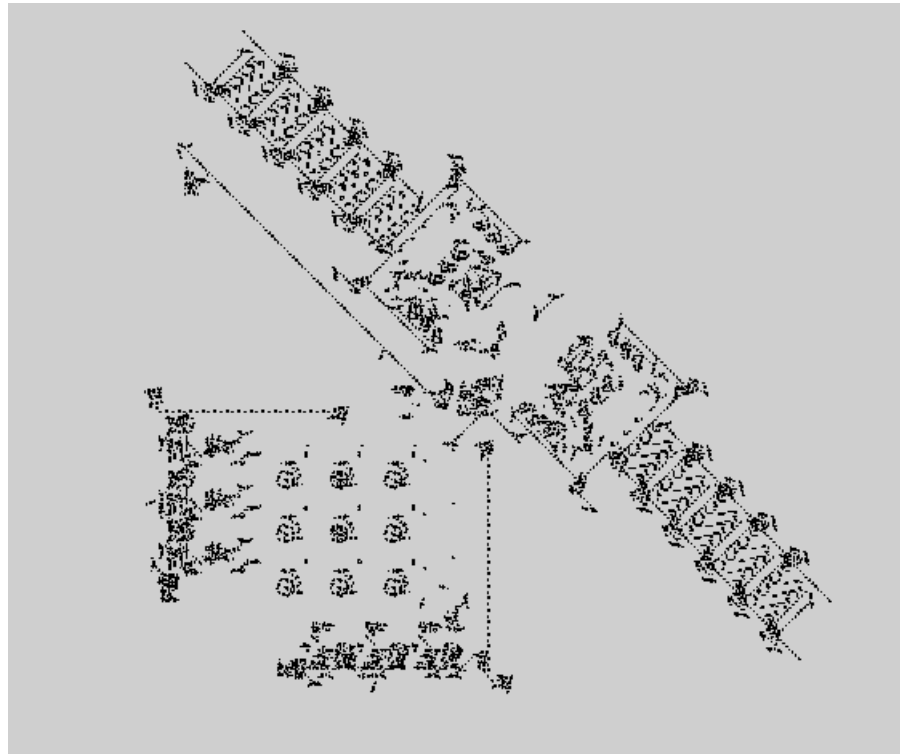
Generatore di segnale  
(costante)

Stopper:  
è inattivo

Output A AND B: arriva segnale (da A)  
il valore è TRUE

# Gate logici nel *Game of Life*

- Macchina di Turing  
Per combinazione di *gate*



# Classi di automi cellulari (secondo Wolfram, 1984)

*Classificazione basata sul comportamento, non sulla struttura o le regole*

- **Classe 1: *uniformi***

Dopo un numero finito di passi, l'automata tende ad un'unica configurazione stabile, a partire da quasi tutte le configurazioni iniziali

- **Classe 2: *periodici***

L'automata produce schemi che si ripetono periodicamente, all'infinito (probabilmente equivalente alla classe degli automi a stati finiti)

- **Classe 3: *caotici***

L'automata produce schemi aperiodici e/o caotici (assomigliano a rumor bianco), che hanno proprietà statistiche quasi identiche alle configurazioni iniziali, almeno dopo un certo numero di passi (configurazioni auto-simili).

- **Classe 4: *complessi***

L'automata produce schemi periodici e comportamenti caotici nello spazio e nel tempo.

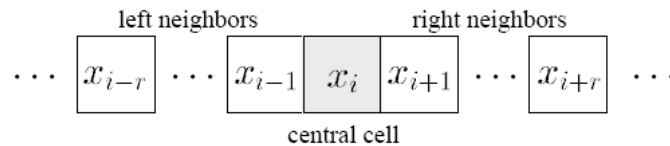
# Automati unidimensionali

- Un anello di celle identiche  $c_i$   
Insieme (finito) di stati  $\Sigma$ ,  $k = |\Sigma|$   
Raggio del vicinato  $r$   
Funzione di transizione  $\varphi : \Sigma^{2r+1} \rightarrow \Sigma$

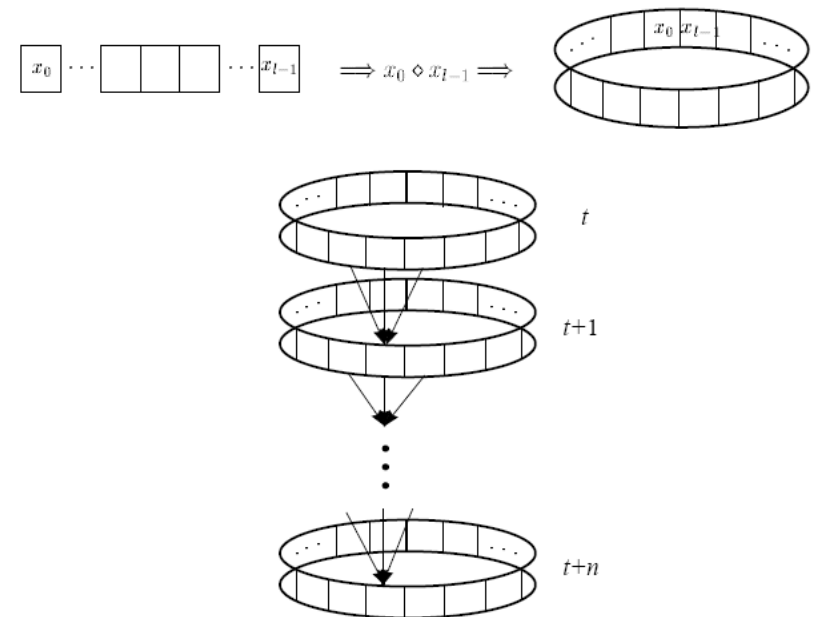
Cellular Automata of  
order  $(k,r)$

$\{\Sigma, r, \varphi, c_i\}$

Neighborhood in one dimension



Dynamics in one-dimension



# Automi unidimensionali di ordine 2,1

- Due stati, una cella di intorno per lato

256 possibile funzioni di transizione  $\varphi$

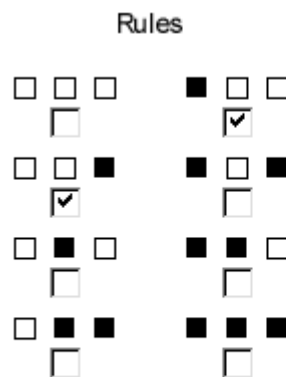
Stato precedente	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0
Nuovo stato	1			0			1			1			0			1			1			1		

Assumendo un ordinamento standard, ciascuna funzione è descritta da otto bit

Nell'esempio è 10110111 (Regola 183)

Lo stato iniziale della riga è l'unico input

La traccia dell'esecuzione è formata dalla lista degli stati successivi della riga



(usare l'applet al link:

<http://www.alesdar.org/oldSite/IS/OneDCA/OneDCA.html>)



# Automati unidimensionali e classi

- Classe 1

  - Regola 32: 00100000

  - Regola 160: 10100000

- Classe 2

  - Regola 33: 00100001

- Classe 3

  - Regola 126: 01111110

  - Regola 30: 00011110

- Classe 4

  - Regola 110: 01101110

# La regola 110

- *Gli automi di classe 4 sono macchine di Turing universali?*

L'ipotesi viene formulata da S. Wolfram già negli anni 80

Egli propone la regola 110 come candidata

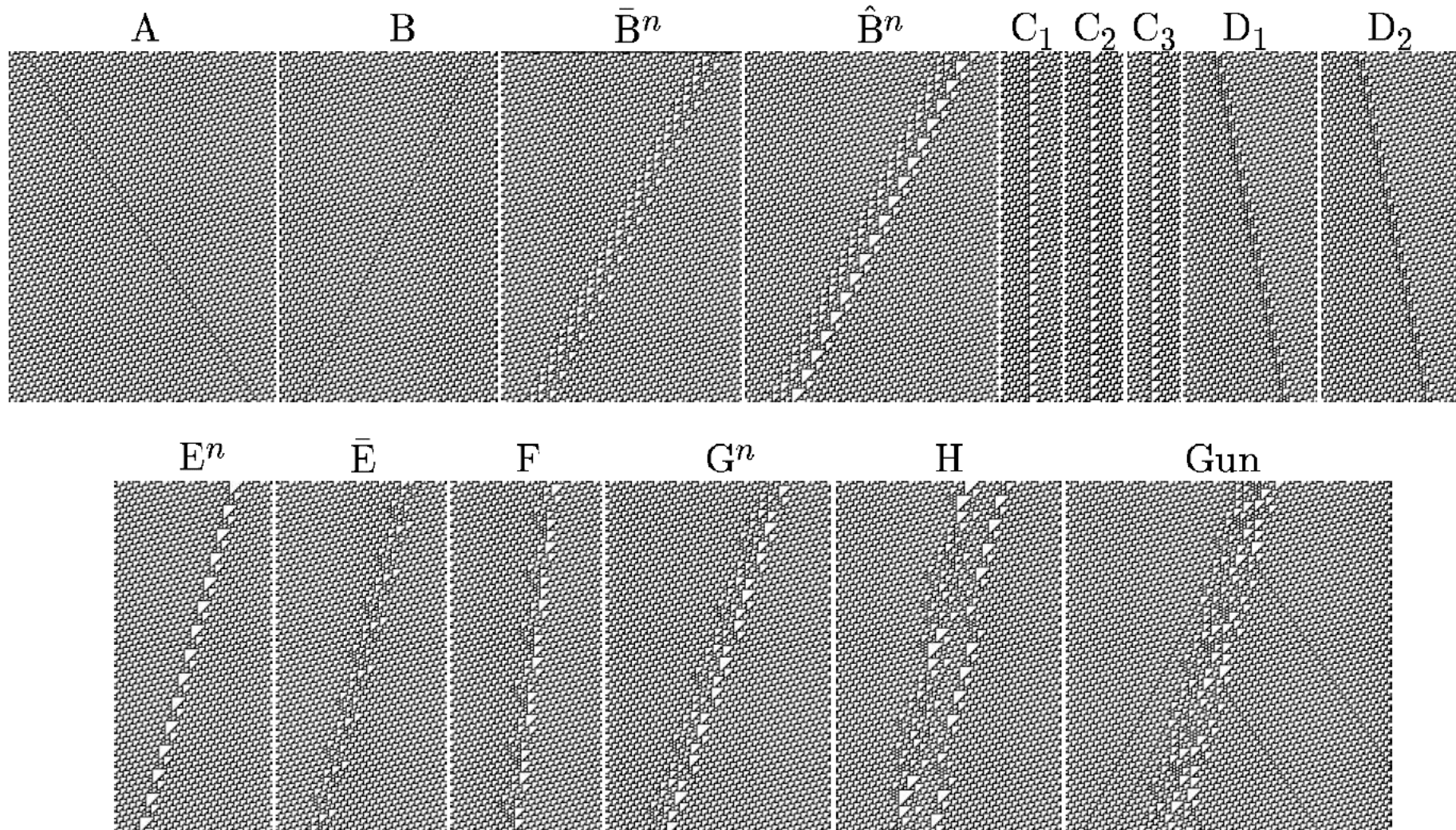
Nel 1999 Matthew Cook dimostra che la regola 110 è in effetti universale

Anche negli automi realizzati dalla regola 110 ci sono "*guns and gliders*"

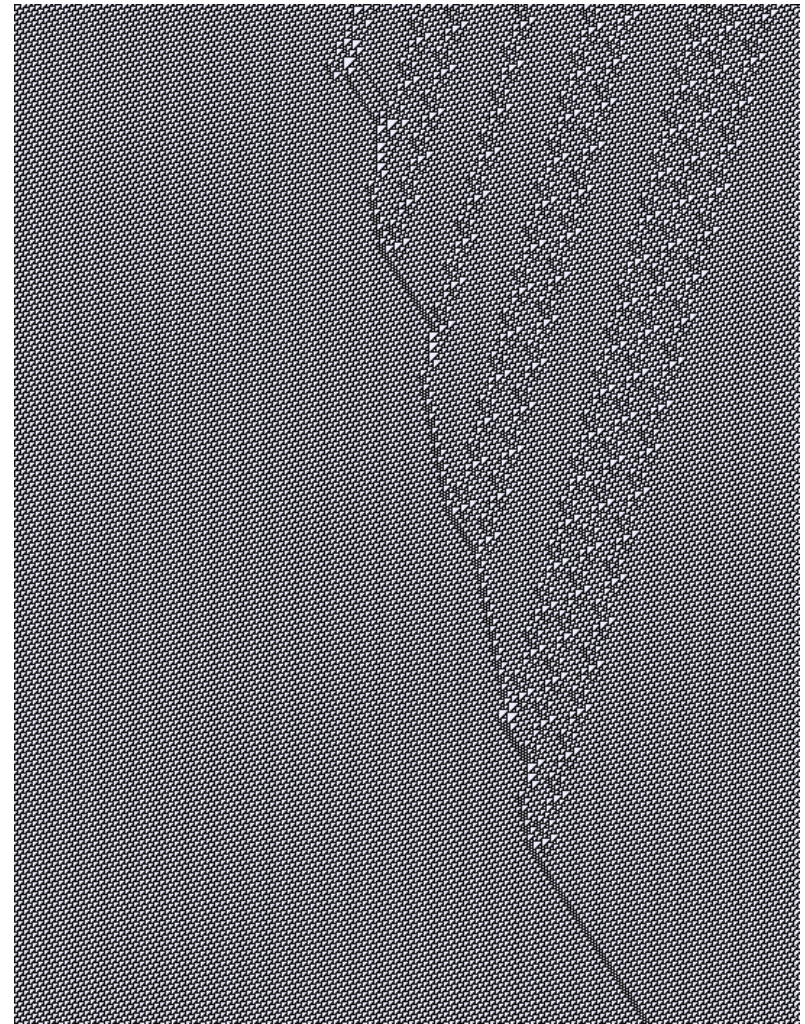
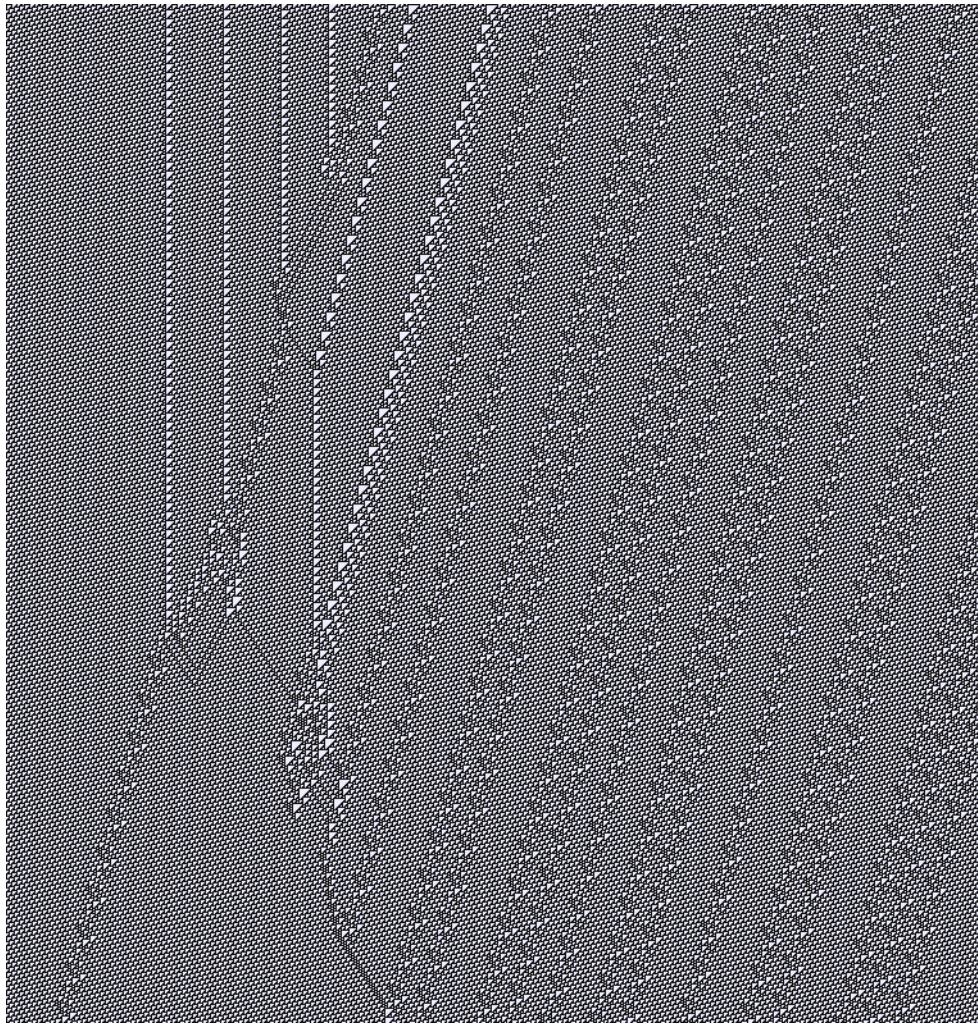
Sfruttando le tipologie di collisioni tra *gliders* si può emulare un sistema di calcolo universale (un *Cyclic Tag System*)

*(Un'altra candidata notevole è la regola 54 – non si conosce la risposta)*

# Guns & Gliders nella regola 110 (Cook, 1999)



# Collisioni nella regola 110



# Automati cellulari in generale

- Un spazio di celle identiche  $\{c_i\}$ , dove  $i \subseteq \mathbb{Z}^d$

$d$  è la dimensione dello spazio

Insieme (finito) di *stati*  $\Sigma$ ,  $k = |\Sigma|$

Talvolta uno degli stati è definito come *quiescente*

Una *configurazione*  $c$  dello spazio delle celle è una funzione  $c : \mathbb{Z}^d \rightarrow \Sigma$

Associa celle a stati

Una configurazione costante (tutte le celle nello stesso stato) è anche detta *omogenea*

Funzione *locale* di transizione  $\varphi : \Sigma^r \rightarrow \Sigma$

Dove  $r$  è la dimensione dell'*intorno* della cella

La definizione di  $\varphi$  dipende anche dalla definizione della topologia dell'*intorno* delle celle

Funzione *globale* di transizione  $G : C \rightarrow C$

Dove  $C$  è lo spazio di tutte le possibili configurazioni di  $\{c_i\}$

# Automati cellulari: definizioni

- Configurazione spazialmente periodica

E' invariante rispetto ad una determinata traslazione nello spazio

$$c(c_i) = c(c_i + T)$$

Dove  $T$  è una specifica traslazione nello spazio

- Configurazione temporalmente periodica

E' invariante rispetto ad una determinata traslazione nel tempo

$$G(c) = G^p(c)$$

La funzione globale di transizione 'si ripete' con periodo  $p$

- *Nilpotenza*

Se esiste una configurazione  $c$  ed un numero  $n$  tali per cui

$$G^n(e) = c$$

Per qualsiasi configurazione di partenza  $e$

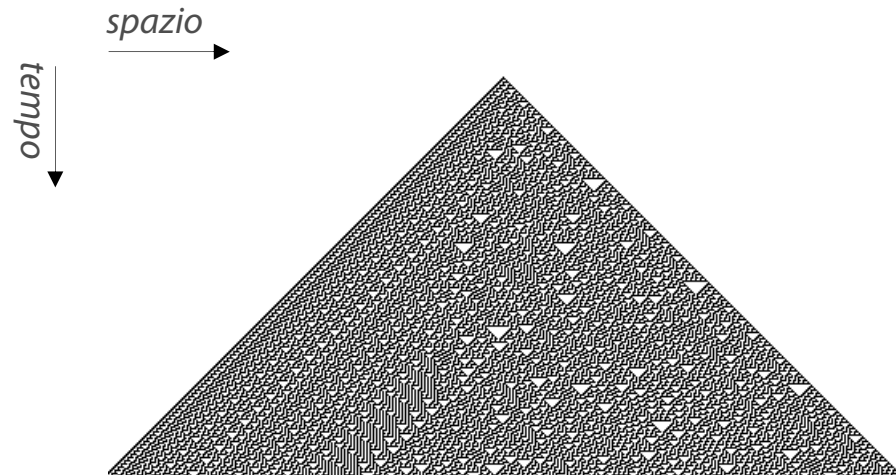
# Automati cellulari: definizioni

- *Space-Time diagrams*

Un diagramma spazio-tempo di un automa  $G$  è formato dalla sequenza

$c, G(c), G^2(c), G^3(c) \dots$

*Diagramma spazio-tempo di un CA unidimensionale*



# Automati cellulari: proprietà

- Reversibilità

Quando la funzione di transizione globale ammette una funzione inversa che sia definibile nei termini di una funzione di transizione locale

$$G_1(c) \circ G_2(c) = I$$

Dove  $I$  è la transizione identica

$G_1$  è detto l'automa *inverso* di  $G_2$

- Iniettività

Quando la funzione  $G$  è iniettiva

Se configurazioni di partenza sono diverse, quelle di arrivo sono pure diverse

- Suriettività

Se la funzione  $G$  è suriettiva

Non esiste una configurazione che non abbia un possibile predecessore

- *Garden of Eden*

Se la funzione  $G$  non è suriettiva, esiste almeno una configurazione che non è 'raggiungibile' (può comparire solo come configurazione iniziale)



# Automati cellulari: risultati

- Garden-of-Eden Theorem (Moore & Myhill, 1962)

La funzione  $G$  è suriettiva sse è iniettiva

Corollario: le seguenti proprietà di  $G$  sono equivalenti

- Iniettività
  - Suriettività
  - Bijettività
  - Reversibilità
- Reversibilità di automi uni-dimensionali
    - La reversibilità di automi uni-dimensionali è decidibile (Amoroso & Patt, 1972)
  - Reversibilità di automi in generale
    - La reversibilità di automi di dimensione  $\geq 2$  non è decidibile (Kari, 1994)

# Automati cellulari: risultati

- Simulabilità (Toffoli, 1977)

*Simulazione:* un automa  $G$  simula un automa  $G'$  se, a meno di cambiamenti di scala, l'insieme dei diagrammi spazio-tempo di  $G$  include l'insieme dei diagrammi di  $G'$

Qualsiasi automa a  $d$  dimensioni  $G$

può essere simulato da un automa reversibile  $G'$  a  $d + 1$  dimensioni

Una macchina di Turing può essere vista come un automa cellulare a una dimensione

Quindi esiste un automa che è in grado di simulare la macchina di Turing in ogni dimensione

- Universalità della regola 110

La regola 110 è universale (Cook & Wolfram, 1999)

- Universalità intrinseca

Un automa  $G$  a  $d$  dimensioni è *intrinsecamente* universale

se è in grado di simulare qualsiasi altro automa di pari dimensione

Sono noti automi unidimensionali, a 6 stati, che sono intrinsecamente universali

La regola 110 è intrinsecamente universale?

Non si sa.

# Automati cellulari: risultati

*“It seems that computational universality  
is a very common property in cellular automata,  
a rule rather than an exception”*

[Wolfram, 2002]