

METODE INTELIGENTE DE REZOLVARE A PROBLEMELOR REALE



Laura Dioşan
Tema 7

Modelare

□ 3 borne științifice:

- Exeprimente
- Teorie matematică
- Simulare și modelare

□ La ce sunt utile modelele?

- Instrumente pentru analiza datelor
- Metode pentru descoperirea de noi cunoștințe
- Înțelegerea naturii ca un sistem de procesare a informațiilor
- Explicarea modului de funcționare a lucrurilor (mecanisme)

...modele...

- ❑ Game of life
- ❑ Modelarea răspândirii epidemiilor (virusilor)
- ❑ Răspândirea incendiilor (în păduri)
- ❑ Împrăștierea uleiului peste rocile poroase
- ❑ Răspândirea bolilor în culturi agricole
- ❑ Modelarea dungilor unei zebre
- ❑ Modelarea dinamicii fluidelor
- ❑ Modelarea sistemelor fero-magnetice (Ising, Potts)
- ❑ Interacțiunile prădător-pradă
- ❑ Reacții între enzime

Game of life

❑ Inventat de John Conway în 1970

- Experimentul original: un sistem cu o regulă simplă (de calcul) poate crea un "calculator universal"
- "calculator universal" Turing – o mașină, bazată pe un sistem de calcul cu reguli simple, capabilă să "depășească" orice tip de procesare de informație
- <https://bitstorm.org/gameoflife/>

❑ Ce este?

- Cel mai simplu univers (sistem) capabil să efectueze calcule
- Arhitectură
 - ❑ Grid rectangular de celule vii (living/on) și celule moarte (dead/off)
- Funcționare
 - ❑ iterativă
 - ❑ În fiecare iterație, celulele sunt guvernate de reguli (de tranziție) simple
 - ❑ Apar forme complexe plecând de la structuri simple

Game of life

□ Problema

- Cum se comportă celulele (trăiesc, mor sau se înmulțesc) de-a lungul vieții
 - Apar anumite comportamente?
- E un joc de tipul *zero-jucători*

□ Modelarea evoluției unei populații de celule cu ajutorul unui grid (rectangular 2D) cu

- celule vii (living/on) și
- celule moarte (dead/off)
- În fiecare iterație, celulele sunt guvernate de reguli (de tranziție) simple → apar forme complexe rezultate din structuri simple

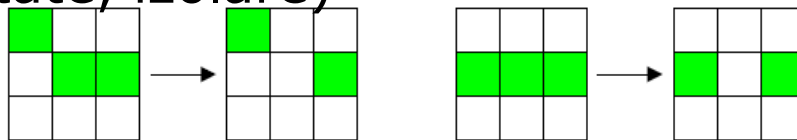
□ Scopul:

- Înțelegerea modului în care
 - condițiile inițiale
 - dimensiunea suprafeței
 - intervalul de timp
- pot afecta răspândirea & evoluția populației de celule
- Descoperirea unui univers (sistem) simplu capabil să efectueze calcule

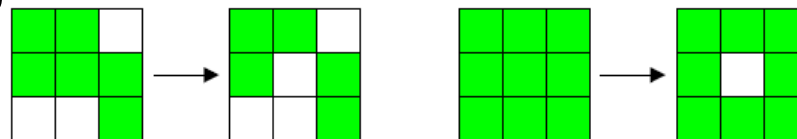
Game of life

□ Modelare -> reguli

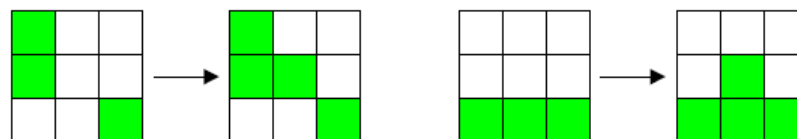
- Moare dacă numărul de celule vii vecine este ≤ 2 (singurătate, izolare)



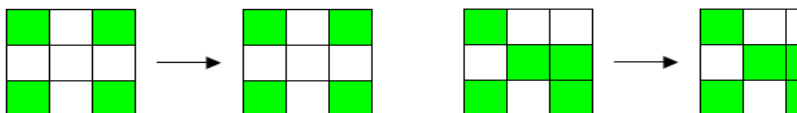
- Moare dacă numărul de celule vii vecine este ≥ 5 (supraaglomerare)



- Supraviețuiește dacă numărul de celule vii vecine = 3 (procreare)



- Obs. Dacă o celulă are 4 vecini vii, starea ei rămâne nemodificată

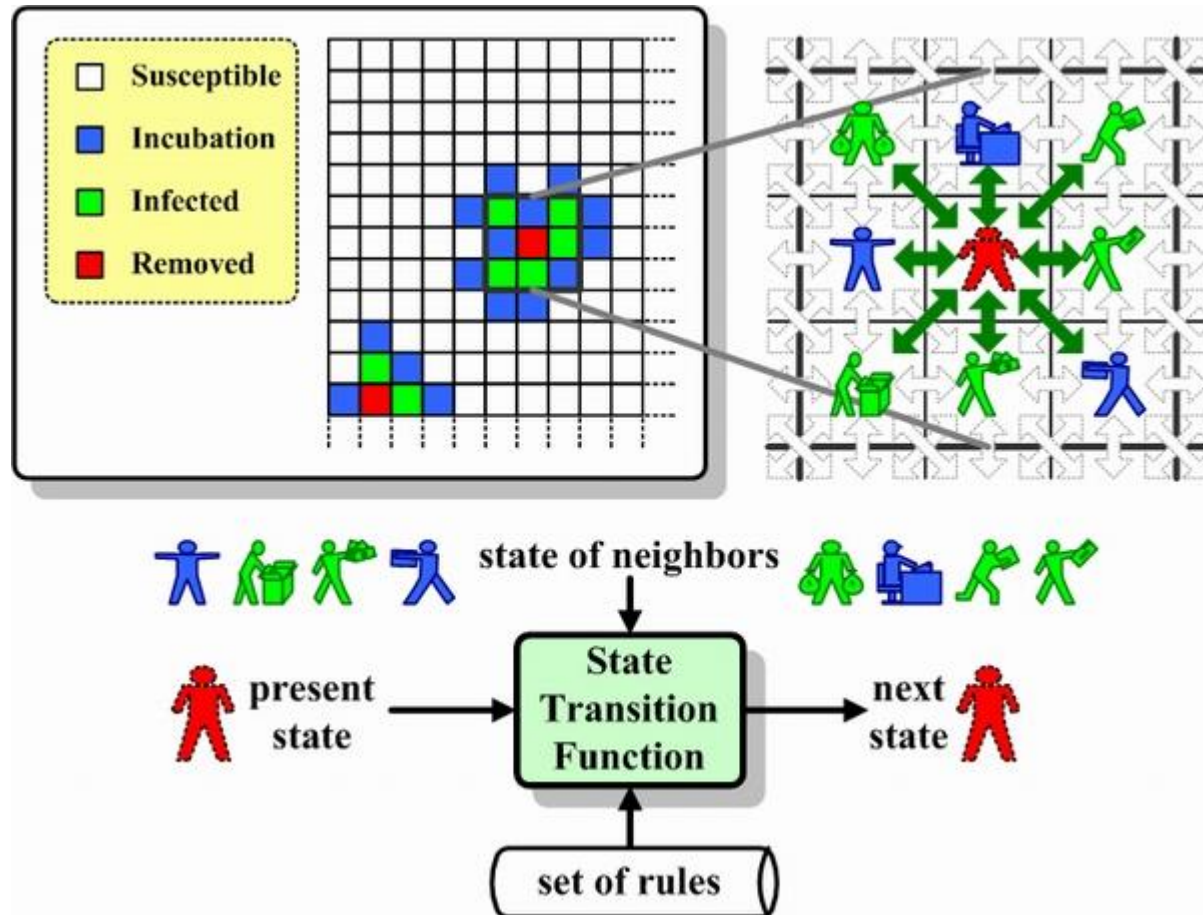


Game of life

□ Remarci

- Reguli de modificare foarte simple care determină comportamente și configurații nepredictibile
- Conway: este imposibil de decis dacă o formă finită (o configurație de celule în care numărul celulelor vii este finit) va "muri" complet sau nu
 - Nu eistă un program de calculator care să primească la intrare o formă finită și care să decidă exact dacă forma va muri sau nu
- S-a dorit un computer capabil de a efectua orice fel de calcule
 - Identificarea modului de propagare a structurilor – componentele unui calculator digital

Răspândirea bolilor (epidemiilor)



Răspândirea bolilor (epidemiilor)

□ Problema

- Cum se răspândește o epidemie într-o anumită comunitate de indivizi

□ Modelarea comunității cu ajutorul unui grid (rectangular 2D) cu celule care conțin indivizii

- Sănătoși
- Infectați

□ Scopul:

- Înțelegerea modului în care
 - condițiile inițiale
 - dimensiunea populației
 - intervalul de timp
- pot afecta răspândirea epidemiei

Răspândirea bolilor (epidemiilor)

□ Posibile stări ale celulelor

- Sănătoasă
- Infectată, capabilă să răspândească virusul (A1)
- Infectată, în ultimul stadiu (urmează să fie ucis de sistemul imunitar) (A2)
- Moartă

□ Configurația inițială

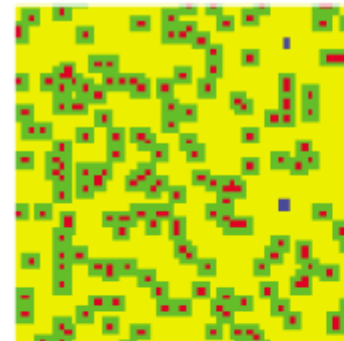
- Grid cu celule sănătoase și o mică parte de celule infectate (A1) (ex. contaminarea inițială cu HIV)

□ Reguli simple de modificare a stării unei celule

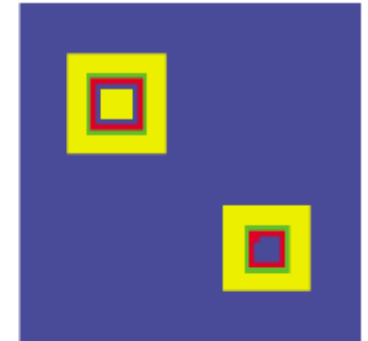
- Modificarea unei celule sănătoase
 - Dacă are cel puțin un vecin infectat A1, devine și ea infectată A1 – răspândirea HIV prin contact
 - Dacă nu are vecini infectați A1, dar are cel puțin un vecin infectat A2, devine infectată A1 – celulele infectate pot contamina alte celule înainte de a muri dacă concentrația lor este peste un anumit prag
 - Altfel, rămâne sănătoasă
- O celulă infectată A1 devine infectată A2 după t pași (t – timpul necesar ca sistemul de imunitate să răspundă)
- O celulă infectată A2 moare
- Celulele moarte pot fi înlocuite de celule sănătoase (cu o anumită probabilitate)

Răspândirea bolilor (epidemiilor)

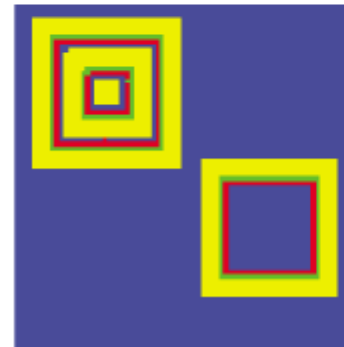
- ❑ Configurația gridului la diferite momente de timp
- ❑ Coduri
 - Celule sănătoase
 - Celule infectate A1
 - Celule infectate A2
 - Celule moarte



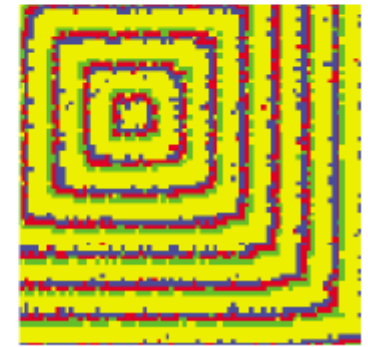
(a)



(b)

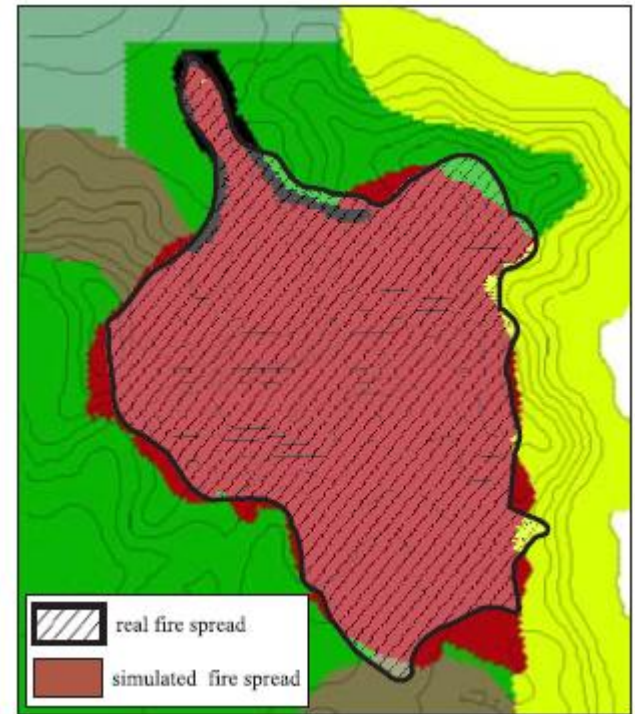
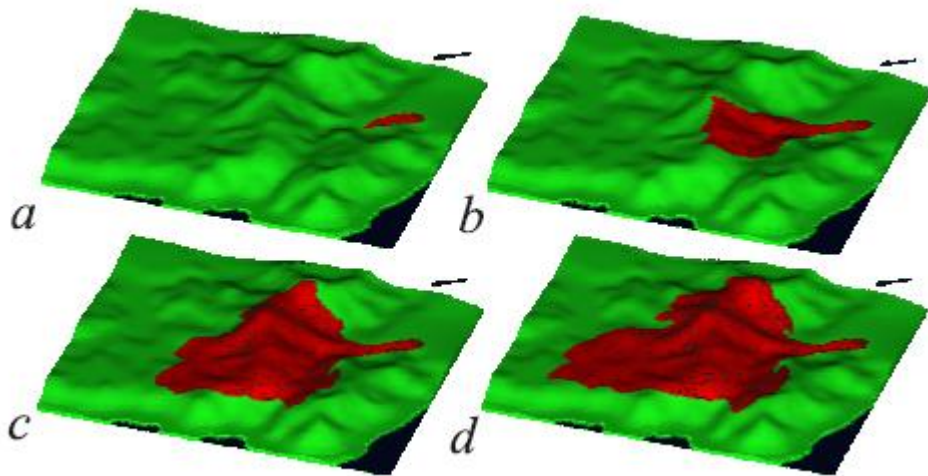


(c)



(d)

Răspândirea incendiilor (în păduri)



Răspândirea incendiilor (în păduri)

□ Problema

- Cum se răspândește focul pe o anumită suprafață (cu arbori)

□ Modelarea suprafeței cu ajutorul unui grid (rectangular 2D) cu celule

- celule care conțin arbori
- celule care conțin arbori care ard (flăcările pot aprinde alți arbori, dar pot exista și arbori rezistenți la foc)

□ Scopul:

- Înțelegerea modului în care
 - condițiile inițiale
 - dimensiunea suprafeței
 - intervalul de timp
- pot afecta răspândirea focului

Răspândirea incendiilor (în păduri)

□ Stări ale unei celule

- Goală
- Cu arbore
- Cu arbore arzând

□ Reguli

- O celulă cu un arbore arzând se transformă în celulă goală
- Un arbore va arde dacă are cel puțin un vecin arzând
- Un arbore se aprinde (cu o probabilitate p) chiar dacă nu are vecini arzând
- O celulă goală se va popula cu un nou arbore (cu o anumită probabilitate)

- <http://www.eddaardvark.co.uk/svg/forest/forest.html>

Automate celulare (Cellular Automata – CA)

- Știința tradițională
 - Legile lui Newton → Stări ale materiei
 - Una din probleme: este imposibil să descrii complet orice stare
- Principiul Heisenberg
 - Este imposibilă cunoașterea exactă a *vitezei* și a *locației* oricărei particule
 - Bazele teoriei quantice
- Concept
 - Simulatoare care încearcă să imite legile naturii
 - Reguli simple pot genera comportamente complexe
 - J. von Neumann & S. Ulam (1940 - 1950)
 - Au dorit crearea și simularea vieții artificiale cu un computer
 - **"As simple as possible, but no simpler."**
 - Construirea *self-replicating patterns* → self-reproducing robots
 - K. Zuse (1960)
 - A propus ca universul să fie un automat celular
 - CA nu au "decolat" până când computerele nu au facilitat simularea lor facilă
 - S. Wolfram (1980)
 - studiază în detaliu CA → proprietăți, capacități de calcul, etc
 - <http://www.wolframscience.com/nksonline/toc.html>

Automate cellulare

□ Concept

- A CA is an **array** of **identically programmed automata**, or cells, which **interact** with one another in a **neighbourhood** and have definite **state**

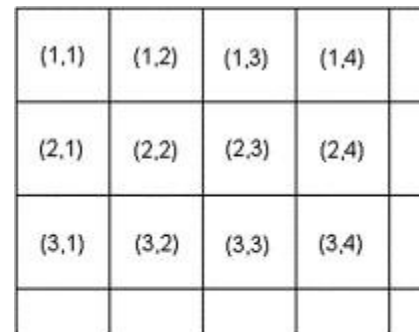
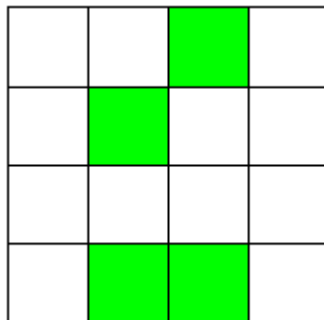
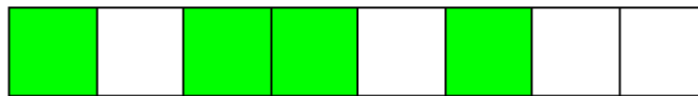
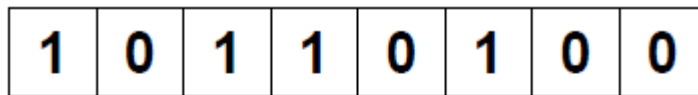
Automate celulare

□ Concept

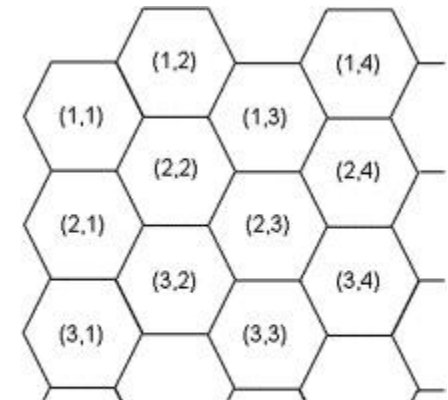
- A CA is an **array** of identically programmed automata, or cells, which interact with one another in a neighbourhood and have definite state

■ Topologia CA

- Şir (liniar)
- Grid (latice)
- Fagure
- Formă neregulată (graf)



(a)



(b)

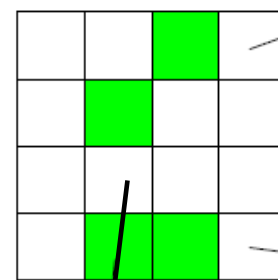
Automate celulare

□ Concept

- A CA is an array of **identically programmed automata**, or cells, which interact with one another in a neighbourhood and have definite state

■ Celula din CA

- Se află într-o anumită stare
- Se modifică conform unor reguli
 - Regula = cum se modifică starea unei celule la momentul $t+1$ fiind date stările vecinilor la momentul t



- if #alive ≤ 2 , then die
- if #alive = 3, then live
- if #alive ≥ 5 , then die

- if #alive ≤ 2 , then die
- if #alive = 3, then live
- if #alive ≥ 5 , then die

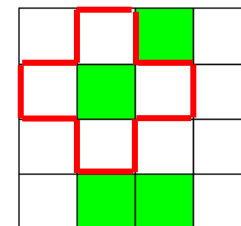
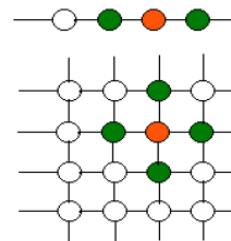
- if #alive ≤ 2 , then die
- if #alive = 3, then live
- if #alive ≥ 5 , then die

Automate celulare

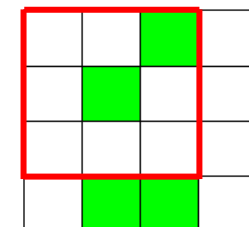
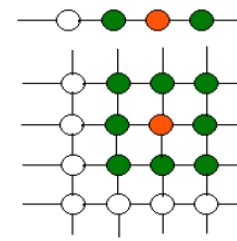
□ Concept

- A CA is an array of identically programmed automata, or cells, which **interact** with one another in a **neighbourhood** and have definite state
- Starea unei celule la momentul $t+1$ este unic determinată de starea sa proprie și de stările celulelor vecine la momentul t
 - interacțiuni locale
- Vecinătatea unei celule
 - topologie (formă)
 - dimensiune

von Neumann



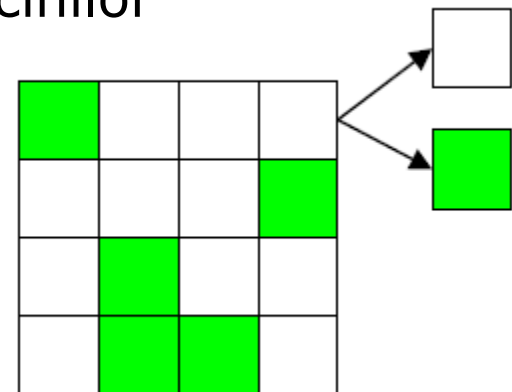
Moore



Automate celulare

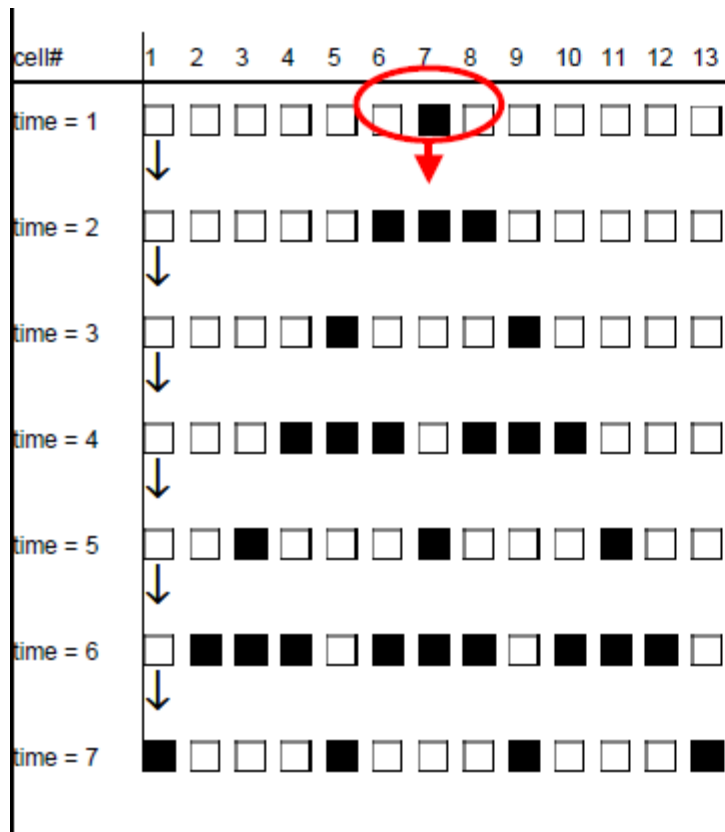
□ Concept

- *A CA is an array of identically programmed automata, or cells, which interact with one another in a neighbourhood and have definite **state***
- Starea unei celule aparține unui domeniu finit
 - 0 sau 1
 - *living* sau *dead*
 - sănătoasă sau infectată
 - Etc.
- Starea unei celule influențează stările vecinilor
 - și viceversa

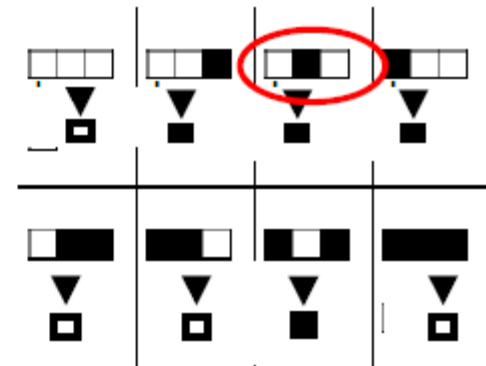


Automate cellulare

□ Exemple



The rules

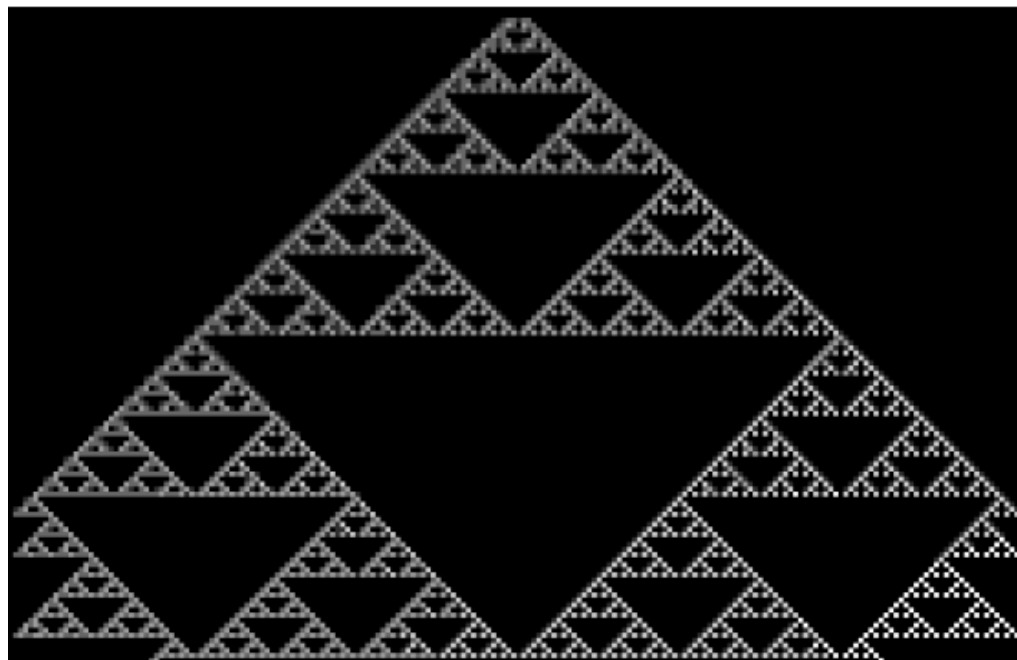


Automate celulare

□ Exemple

■ Triunghiu lui Pascal

				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
	1	7	21	35	35	21	7	1
1	8	28	56	70	56	28	8	1



Automate celulare

□ Clasificare

- Topologia și dimensiunea gridului
 - liniare rectangulare, hexagonale, bazate pe grafe
 - 1D, 2D, 3D, etc.
- Topologia vecinătății
 - von Neumann de rază r
 - Moore de rază r
- Numărul de stări
 - Binare
 - 3 stări
 - Stări multiple
- Reguli

Automate celulare

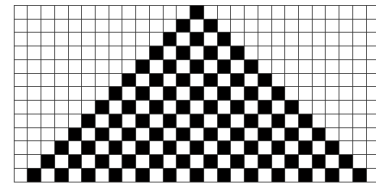
□ Reprezentarea regulilor

- Pp un CA binar (starea unei celule poate fi 0 sau 1)
- Reguli posibile

□ Regula 250 (*repetition*)



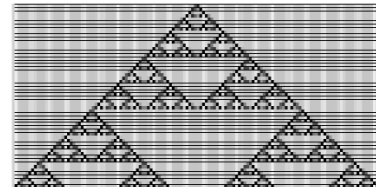
$$11111010_2 = 250_{10}$$



□ Regula 90 (*nesting*)



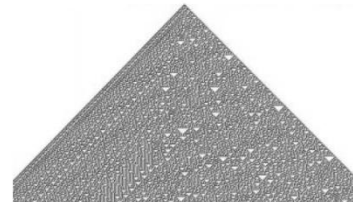
$$01011010_2 = 90_{10}$$



□ Regula 30 (*randomness*)



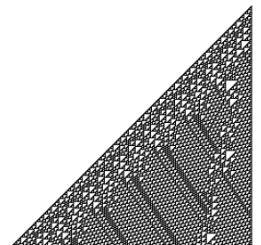
$$00011110_2 = 30_{10}$$



□ Regula 110 (*localised structures*)



$$01101110_2 = 110_{10}$$



Automate celulare - formalism

□ Teoria automatelor

■ Ramură a științei calculatoarelor care

- Încearcă să răspundă la "Ce pot computerele să facă și ce nu pot (ce este dincolo de capacitatea lor)?"
- Ajută la crearea și studierea unor noi modele de calcul (într-un mod clar, fără ambiguități)
- Are implicații foarte practice și reprezintă baza pentru multe aplicații ale lumii reale

Automate celulare - formalism

- Un CA este reprezentat de un graf interconectat Γ (de obicei latice n-dimensională)
- Fiecare celulă a CA poate fi în una din câteva posibile stări.
 - Mulțimea stărilor Q = mulțimea tuturor stărilor posibile ale unei celule
- Perechea (Γ, Q) – spațiul celulelor din CA
- Configurație x a CA
 - este o asociere între graful Γ și mulțimea stărilor Q care atribuie fiecărui nod din Γ o stare din Q
 - $x : \Gamma \rightarrow Q$
 - $x(i) = q$, unde $i \in \Gamma$ și $q \in Q$
 - descrie starea întreagă (starea fiecărei celule) a unui CA pe o scară globală

Automate celulare - formalism

- ❑ Calculul efectuat de CA este un proces local
 - Starea următoare a unei celule depinde doar de starea ei curentă și de stările vecinilor ei (aproapiați)
- ❑ Vecinătatea unei celule N = colecția de celule situate la distanța r sau mai puțin de celula în cauză
- ❑ Dinamica locală (funcția de tranziție)
 - Fiecare celulă a unui CA este o simplă mașină cu stări finite
 - Dinamica locală δ a unei celule este o funcție care primește ca input starea celulei și a vecinilor ei și calculează starea viitoare a celulei

- ❑ De ex, pt un CA 1D: $\delta(x_{i-1}, x_i, x_{i+1}) = x'_i$

- ❑

x_{i-1}, x_i, x_{i+1}	000	001	010	011	100	101	110	111
$\delta(x_{i-1}, x_i, x_{i+1})$	1	0	0	1	0	1	1	0

Automate celulare - formalism

- Un CA este un cvatuplu $M = (\Gamma, Q, N, \delta)$
 - Γ – graf de interconexiuni
 - Q – mulțimea stărilor
 - Game of life sau Procesări de imagini alb-negru $Q = \{0, 1\}$
 - Fizică: stare = vector și $Q = \{\text{poziția vectorilor, momentele}\}$
 - N – vecinătatea
 - δ – dinamica locală (funcția de tranziție)
 - Game of life: cine moare, cine supraviețuiește
 - Fizică: matrici sau ecuații diferențiale

Automate celulare - formalism

□ Dinamica locală δ

- Descrie calculul efectuat de fiecare celulă

□ Dinamica globală T

- Descrie calculul efectuat de întreg automatul
- Descrie cum se modifică starea întregului automat de la un moment de timp la altul (de la o iterație la alta)
- Este o funcție de asociere între mulțimea configurațiilor C și ea însăși
 - $T: C \rightarrow C$

□ Aplicații

■ CA în jocuri

- Game of life, firing squad, dilema prizonierilor

■ CA în calcul paralel

- Multiplicări, cirul nr prime, sortări
- Procesări de imagini

■ CA pentru modelarea naturii și societății

- Alternativă pentru ecuații diferențiale → modele de spin
- 1D CA + regula 30 → crearea numerelor pseudo-aleatoare (Mathematica)

□ Problema majorității

- Identificarea regulilor care pentru un automat 1D binar (stări 0 sau 1) de dimensiune $i + j$ (i – nr celulelor în starea 0, j – numărul celulelor în starea 1) efectuează un calcul de vot majoritar (proporția elementelor = 1 este peste un prag dat)