

Simulácia celulárnych automatov pomocou prostredia MCell

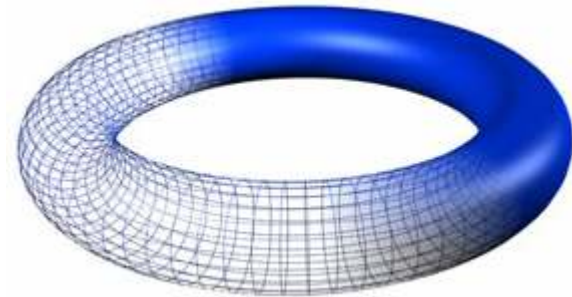
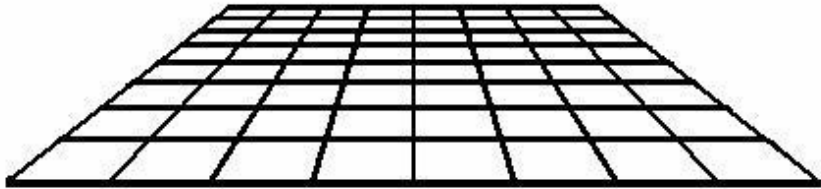
Peter Sýkora, Peter Vojtek

Obsah

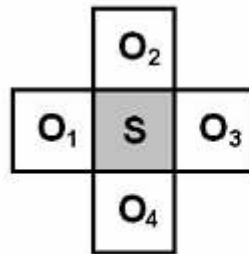
- Úvod do celulárných automatov
- Podporované typy celulárných automatov v MCell
- Výhody a nevýhody programu MCell

Celulárny automat

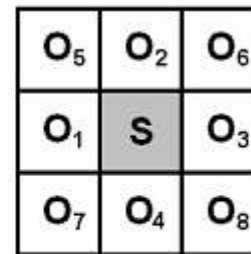
- dynamický systém, diskretný v čase a priestore
- pravidelná štruktúra buniek v priestore (1D, 2D, ...)



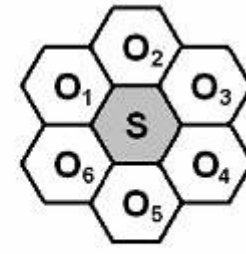
- bunka nadobúda jeden z K stavov
- stav bunky v čase $t+1$ je určený stavom okolia bunky v čase t



Neumann



Moore



Hexagonal

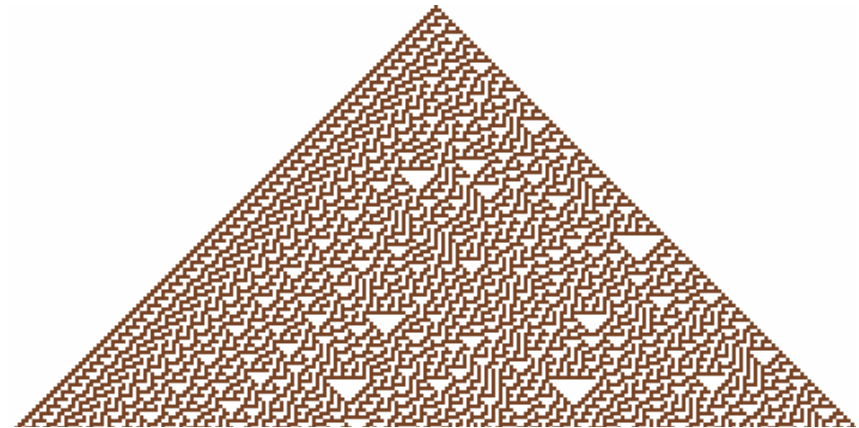
Celulárny automat – praktické aplikácie

- Simulácia dopravy v meste
- Kryptovanie dát
- Simulácia katastrof (lesné požiare)
- Biológia (predikovanie epidémií)

Celulárny automat v prírode



Homôľka – *Conus textile*



Rule 30

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

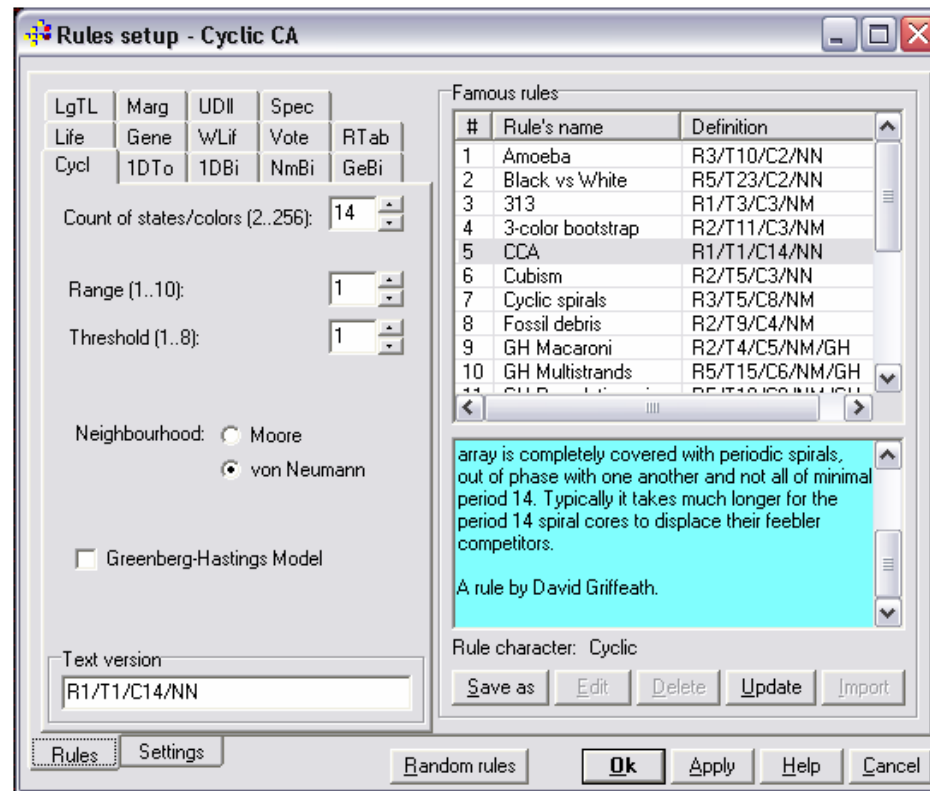
MCell

- platforma Windows, distribuované pod GNU GPL
- až 256 stavov bunky (farieb)
- veľkosť plochy až 50 000 x 5 000
- databáza 1700 vzorov v 15 skupinách
- intuitívne ovládanie, klávesové skratky

- <http://www.mirekw.com>

MCell

- jednoduché nastavovanie pravidiel
- pravidlá sú uložené ako textové reťazce – jednoduché zdieľanie

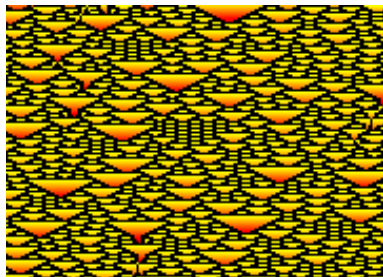


MCell

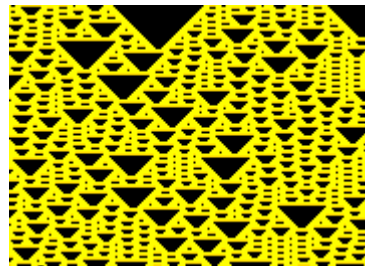
- možnosť „kreslenia“ stavov
 - aj počas behu simulácie
- analytické možnosti
 - určenie periódy a oscilácií
 - štatistický výskyt stavov buniek
 - monitorovanie prechodov stavov v určenej lokalite
 - korelácia stavov určených buniek
 - vnesenie šumu

1D Totalistic

- triedu navrhol autor programu MCell
- prežitie/narodenie bunky závisí od počtu buniek v určitom okolí
- V prípade použitia histórie (počet stavov väčší ako 2) bunka umiera najskôr po prežití všetkých stavov(starne). Staršie bunky ako novonarodené zaberajú miesto, ale pre narodenie nových buniek sa nerátajú
- maximálny polomer okolia je 10



Champagne

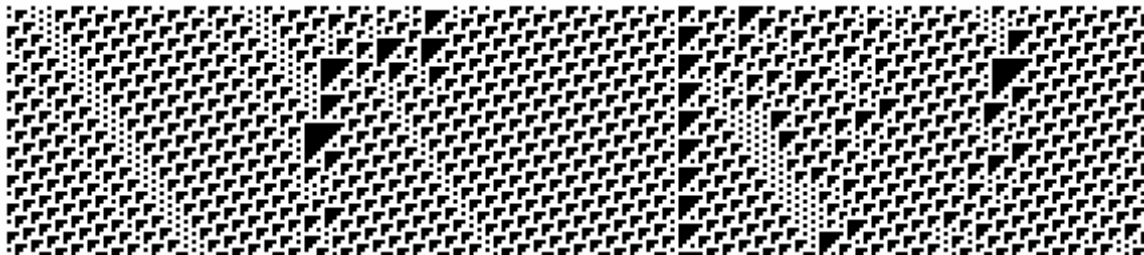


Shaded Triangles

1D Binary

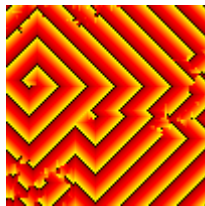
- asi najpreskúmanejšia trieda automatov
- stav bunky je určený prechodovou tabuľkou pre každú možnú kombináciu okolia
- polomer okolia je 1 až 4 bunky
- Touto triedou sa dajú opísať aj automaty triedy 1D Totalistic do polomeru 4, ale zápis je komplikovanejší
- Rule 110 – veľmi jednoduchý systém, v ktorom vznikajú pomerne zložité štruktúry

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



Cyclic

- prechod zo stavu S_i je možný len do jediného ďalšieho stavu S_{i+1}
- bunka v stave S_i zmení stav na S_{i+1} , ak je v okolí bunky určitý minimálny počet buniek v stave S_{i+1}



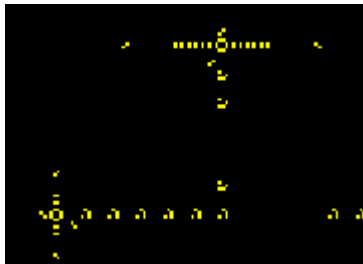
CCA



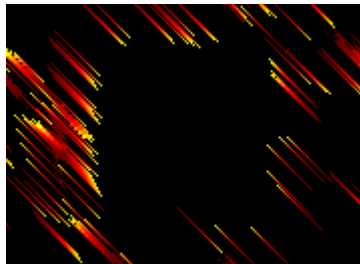
LavaLamp

General Binary

- stav bunky závisí nielen od počtu buniek okolia v určitom stave, ale tiež od ich presnej polohy
- umožňuje definovať anizotropné (smerovo závislé) pravidlá



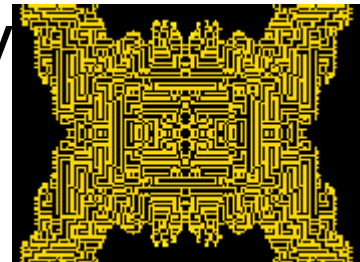
LogicalRule



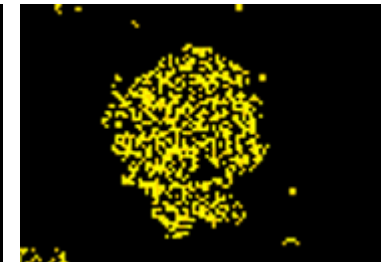
Meteorama

Life

- uvažuje sa Moorovo okolie, polomer 1 (8 susedov)
- pravidlá v tvare: Survival/Birth
 - ak bunka susedí so *Survival* bunkami, tak prežije do ďalšieho kola, inak umiera
 - *Birth*: počet buniek, s ktorými keď susedí mŕtva bunka, tak v ďalšom kole oživa
- najznámejšia Conway's Life (23/3):
 - živá bunka s menej ako dvoma susedmi zomiera kvôli osamoteniu
 - živá bunka s viac ako tromi susedmi zomiera kvôli preplneniu
 - mŕtva bunka s tromi susedmi „oživa“
 - živá bunka s dvoma, alebo tromi susedmi zostáva živá
- množstvo alternatív (explozívne, chaotické, stabilné)
 - napr. 12345/3 bludisko (explozívne)





Maze (explosive)

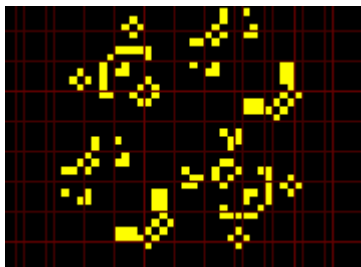


Amoeba (chaotic)

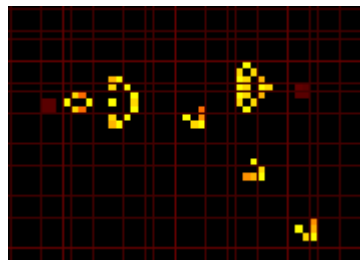
Life (pokr.)

- Príšery (Conway's life)

- blinker 
- slider (klzák)  smer pohybu
- klzákové delá
- oscilátory



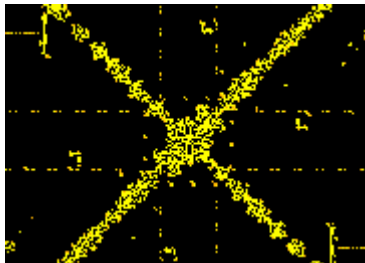
Osc p 66



GUN30

Generations

- pravidlá ako hra Life, ale bunky, ktoré by podľa pravidiel Life zomreli, starnú (nemôžu vytvoriť novú bunku)
- okrem parametrov *Survival* a *Birth* aj parameter, ktorý udáva počet stavov (*States*)



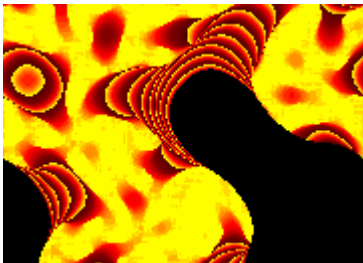
StarWars3



Bloomerang

Larger than Life

- oproti Life ponúka širšie susedstvo (von Neuman/Moore), možnosť starnutia buniek(ako pri Generations).
- parametre *Survival* a *Birth* sa však zadávajú rozsahom od-do

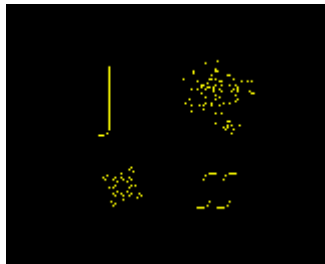


Majority

Margolus

- 2D mriežka je rozdelená na bloky 2x2
- pravidlá sú definované prechodovou tabuľkou (blok => blok)

– napr.:  =>   =>  ...



Rotations IV

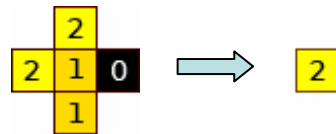


Sand

Neumann Binary

- 2D, von Neumannova susednost'
- bunky v 2 – 4 stavoch.
- stav bunky sa zmení na základe svojho stavu a stavov okolitých buniek
- pravidlá sú prechodová tabuľka ($State^5 \rightarrow State$)

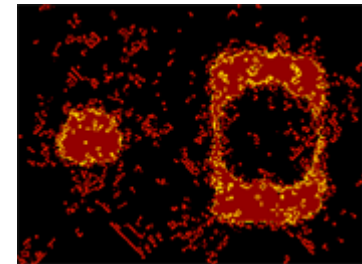
– príklad



GreenBerg

Rules Tables

- 2D, dovoľuje bunkám skok do akéhokoľvek stavu na základe aktuálneho stavu a počtu živých susedov
- Prechody
 - (State, NoFiringNeigh) -> State
- Conway's life by sa zapísal pomocou takýchto pravidiel takto (pri použití Moorovej susednosti):



EcoLiBra

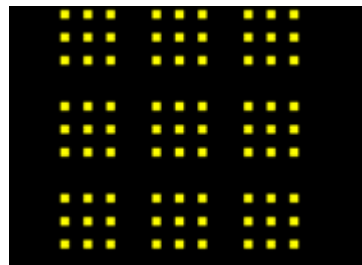
	0	1	2	3	4	5	6	7	8	Vysvetlenie
0	0	0	0	1	0	0	0	0	0	Mŕtve bunky s 3-mi živými susedmi ožijú (dostanú sa zo stavu 0 do stavu 1)
1	0	0	1	1	0	0	0	0	0	Živé bunky s 2-mi a 3-mi susedmi prežijú (ostanú v stave 0)

Vote for Life

- Trieda 2D celulárnych automatov s najjednoduchšími pravidlami
 - Špecifikuje sa iba koľko susedov (v Moore okolí) spolu s bunkou samotnou musí existovať aby bola bunka v nasledujúcom cykle živá
 - Napríklad 46789 je ekvivalentné s Life S35678/B4678



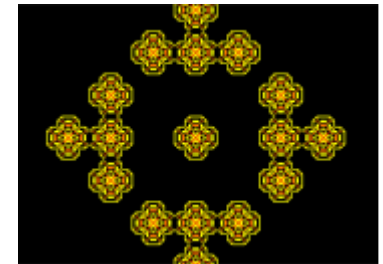
Vote



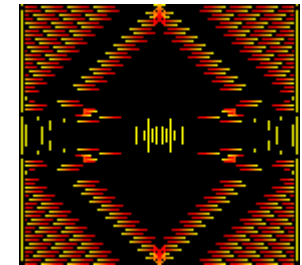
Fredkin

Weighted Life

- navyac od Life sa dajú jednotlivým susedom a taktiež bunke samotnej definovať váhy
 - od -256 do 256
- vypočíta sa súčet váh aktívnych susedov
- ďalej sú zadané hodnoty *Survival* resp. *Birth*, ktoré definujú pri akých hodnotách sumy bunka prežije, resp. mŕtva ožije
- navyac oproti life podporuje aj stavy známe z Generations
- klasický Life – váhy všetkých susedov = 1, váha samotnej bunky 0, počet stavov 2



Parity_2



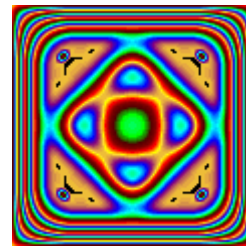
Navaho1

Weighted Generations

- okrem pridania váh jednotlivým susedom sú pridané aj váhy možným stavov
 - ak je sused aktívny a jeho váha je m a je v stave, ktorého váha je n , započítava sa $m \cdot n$

User DLLs

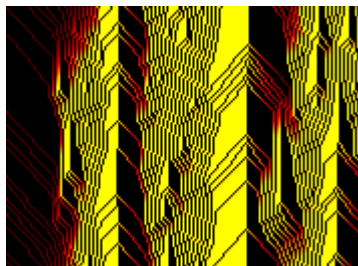
- MCell dovoľuje používateľom naprogramovať vlastné pravidlá, ktoré nemôžu byť špecifikované pomocou existujúcich tried pravidiel
 - jednoducho sa vytvorí DLL, ktorá bude obsahovať implementovanú funkciu CARule(na vstup dostane ako argumenty stavy susedov a vráti nový stav bunky). Táto funkcia je vykonávaná pre každú bunku.
 - od verzie 4.20 môžu byť tieto pravidlá parametrizovateľné



Rug

Special Rules

- obsahuje neštandardné pravidlá, ktoré sa nedajú pridelit' do inej kategórie a nemôžu byť naprogramované ako externé DLL moduly, pretože narúšajú syntax, alebo princípy MCell
 - Napríklad „Traffic CA“ je pravdepodobnostné 1D pravidlo, ktoré môže meniť dve bunky, keď sú splnené určité podmienky o susedoch



Traffic CA

Zdroje

- Wikipedia contributors (2006). Cellular automaton. *Wikipedia, The Free Encyclopedia*.
http://en.wikipedia.org/wiki/Cellular_automata
- plaNeT Alife
<http://alife.tuke.sk/>
- MCell
<http://www.mirwoj.opus.chelm.pl/ca/index.html>