セルオートマトン

モデリングとシミュレーション

2019年度

離散モデルの可能性

- 離散(discrete)
  - 値が飛び飛び、例えば整数の値しかとらない
- 連続的な時間ではなく、ある時間間隔で観測する
- 空間もある間隔で観測する、あるいはある区間の平均を観測する
- 状態も離散化

離散化の利点・欠点

- 微分方程式に書けない変化
  - 発展の規則として記述できる
  - 本当に正しいかの検証が必要
- シミュレーション
  - 規則として記述できる
  - 整数演算は高速
  - 計算誤差が出ない

セルオートマトン Cellular Automata

- 空間をセル(cell、ます)に区切る
- 時間を離散化する
- 時間発展規則を定める
  - 周囲の状態で次の状態を定める
- automata
  - automaton の複数形
    - a machine that moves without human control.

1次元セルオートマトンの例

$$s_i(t+1) = \left(s_{i-1}(t) + s_i(t) + s_{i+1}(t)\right) \bmod 2$$

注目しているセルの次の時刻の値は
そのセルと隣接するセルの値の和を2で除した余り

0	0	1	1	1	0	1	1	0	0	t

$s_i(t)$

$s_i(t+1)$

0	1	0	1	0	0	0	0	1	0	$t+1$

周期境界条件

周期境界条件
periodic boundary conditions

- 1次元の場合
  - 両端が繋がり、サーキット状になっている
- 例：10個のセル

9	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	0

周期境界条件 periodic boundary conditions

- 例：N個のセル
  - $s_i: 0 \leq i < N$
  - $s_{-1} = s_{N-1}$, $(0-1+N)\%N = N-1$
  - $s_N = s_0$, $(N-1+1)\%N = 0$
  - セルiの右 $(i+1)\%N$
  - セルiの左 $(i-1+N)\%N$

1次元セルオートマトン 一般

- 隣接する$2r+1$個のセルの状態で、次の時刻の状態を決定

$$s_i(t+1) = F\left(s_{i-r}(t), s_{i-r+1}(t), \cdots, s_i(t), \cdots, s_{i+r}(t)\right)$$

- すべてのセルに同一の規則Fを適用
- すべてのセルを同時に更新
  - コンピュータは、すべての状態を同時に更新できない。どうする？

同期的更新(parallel update)一般論

- 複数のセルなどの状態を、同期的（つまり同時に）更新するには
- 次の時刻の状態を入れる専用のデータ構造（ダミーと呼ぶ）を作り、そこに書き込む
  - 元のシステムの状態は更新しない
- ダミーの状態を、元のシステムに書き込む

同期的更新の例

$$s_i(t+1) = \left(s_{i-1}(t) + s_{i+1}(t)\right) \bmod 2$$

t	0	0	1	1	1	0	1	1	0	0

次の時刻の状態をダミーシステムに書き込む

ダミー	0	1	1	0	1	0	1	1	1	0

ダミーシステムの状態を元のシステムに書き込む

$t+1$	0	1	1	0	1	0	1	1	1	0

周期境界条件

非同期的更新の結果は
例えば左端のセルから更新

$$s_i(t+1) = \left(s_{i-1}(t) + s_{i+1}(t)\right) \bmod 2$$

t	0	0	1	1	1	0	1	1	0	0
$t+1$	0	1	0	1	1	0	1	1	1	0

ダミーシステムを使わず、
直接システムの状態を更
新すると

正しくない

周期境界条件

最も簡単な1次元セルオートマトン

- 内部状態$\{0,1\}$
- $r = 1$

$$s_i(t+1) = F\left(s_{i-1}(t), s_i(t), s_{i+1}(t)\right)$$

- 右辺の引数のパターンは3bit=8通り
  - 三つの連続するセルの状態
- 8通りの入力に0か1を割り当てる
  - 規則の総数：$2^8 = 256$通り

規則の例

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

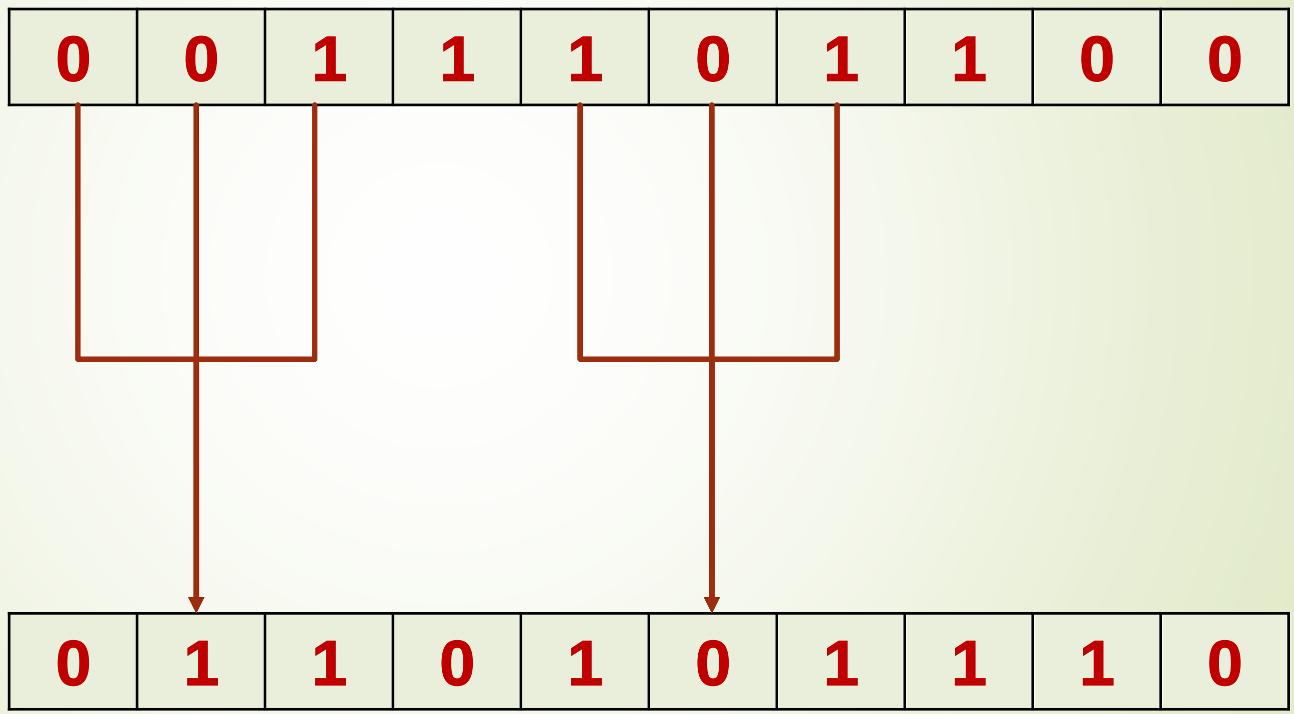
$$(10111000)_2 = 184$$

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

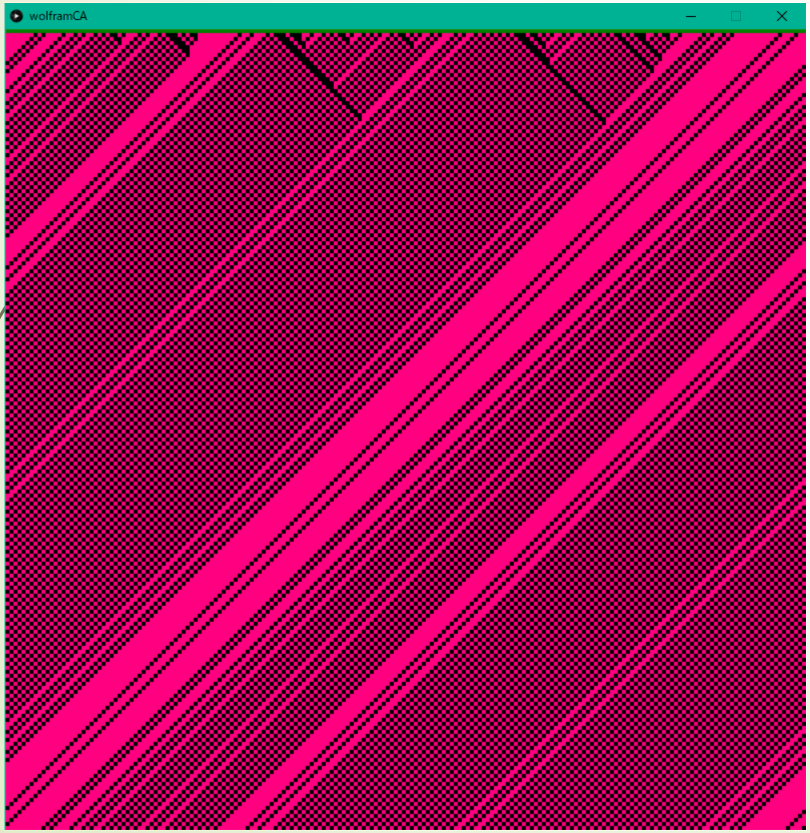
$$(01011010)_2 = 90$$

Wolframの基本CA：左右対称なもの

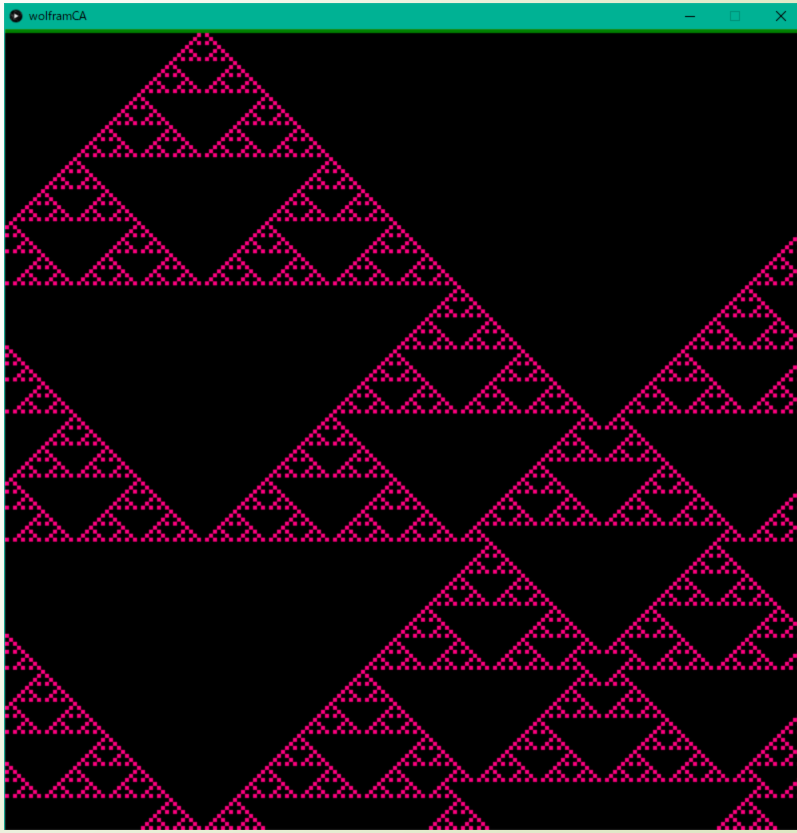
例：Rule-90 (周期境界条件)



Rule 184



Rule 90



Ruleから作ろう

- Rule 150

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

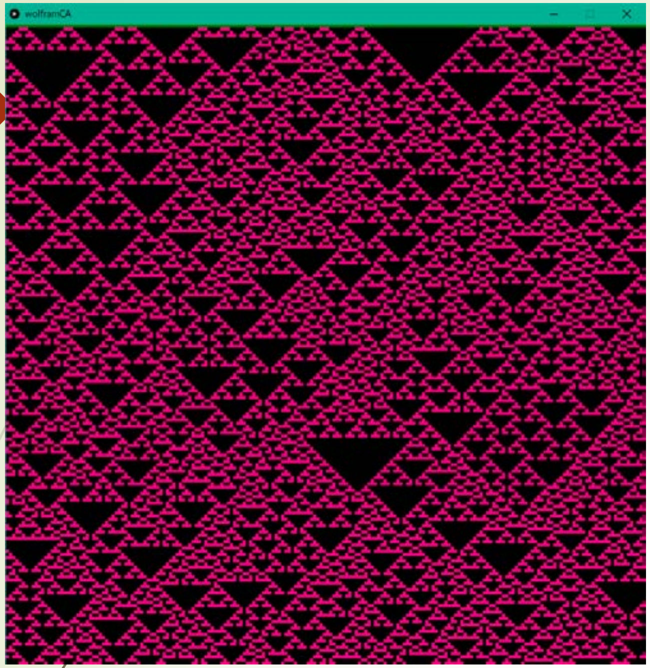
$$150 = 128 + 16 + 4 + 2 = 2^7 + 2^4 + 2^1 = (10010010)_2$$

- Rule 54

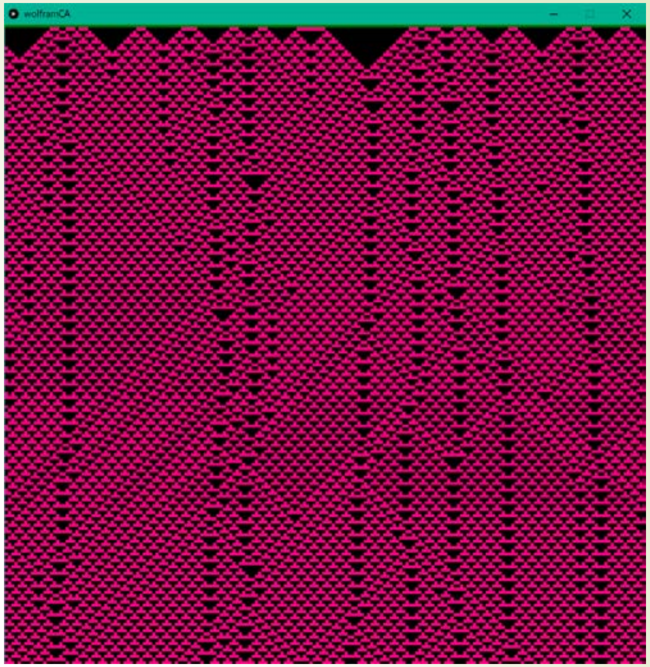
入力	111	110	101	100	011	010	001	000
出力	0	0	1	1	0	1	1	0

$$54 = 32 + 16 + 4 + 2 = 2^5 + 2^4 + 2^2 + 2^1 = (00110110)_2$$

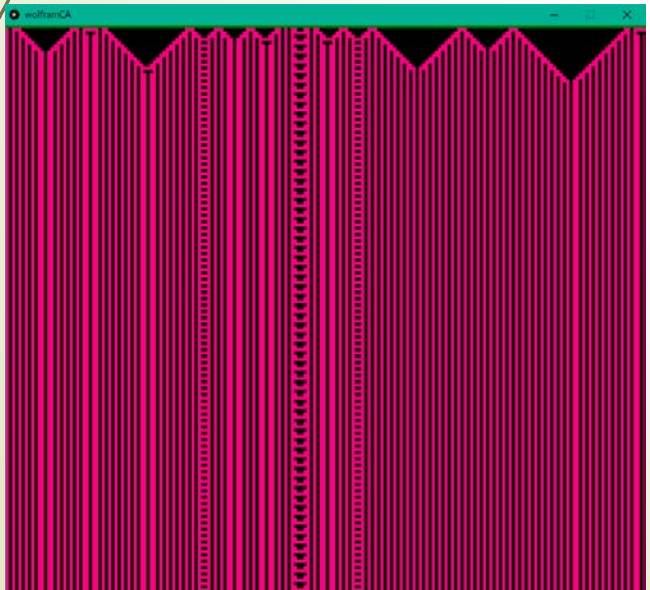
Rule 22



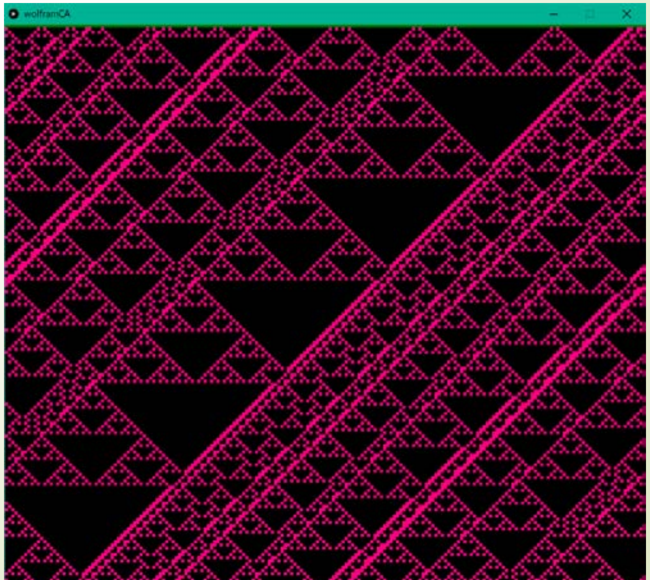
Rule 54



Rule 94



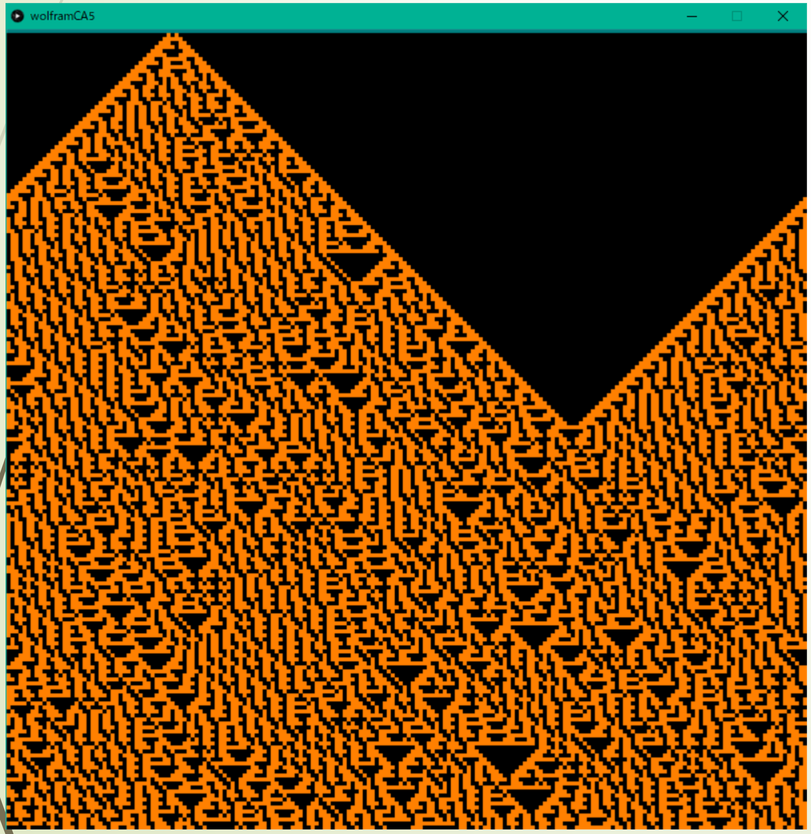
Rule 154



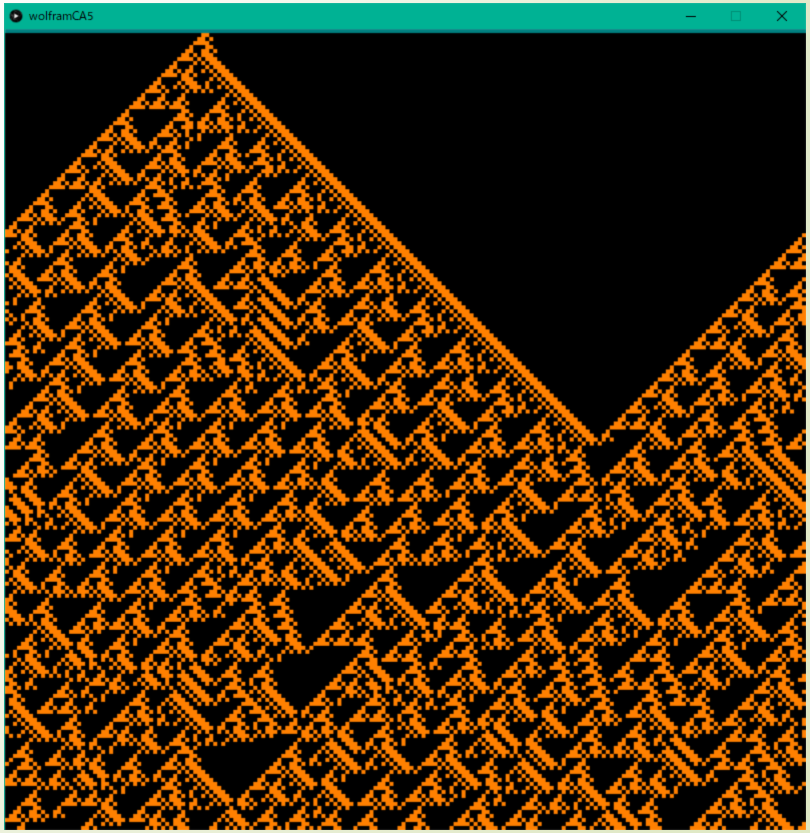
拡張$r = 2$

$$s_i(t+1) = F\left(s_{i-2}(t), s_{i-1}(t), s_i(t), s_{i+1}(t), s_{i+2}(t)\right)$$

393410540



390097500



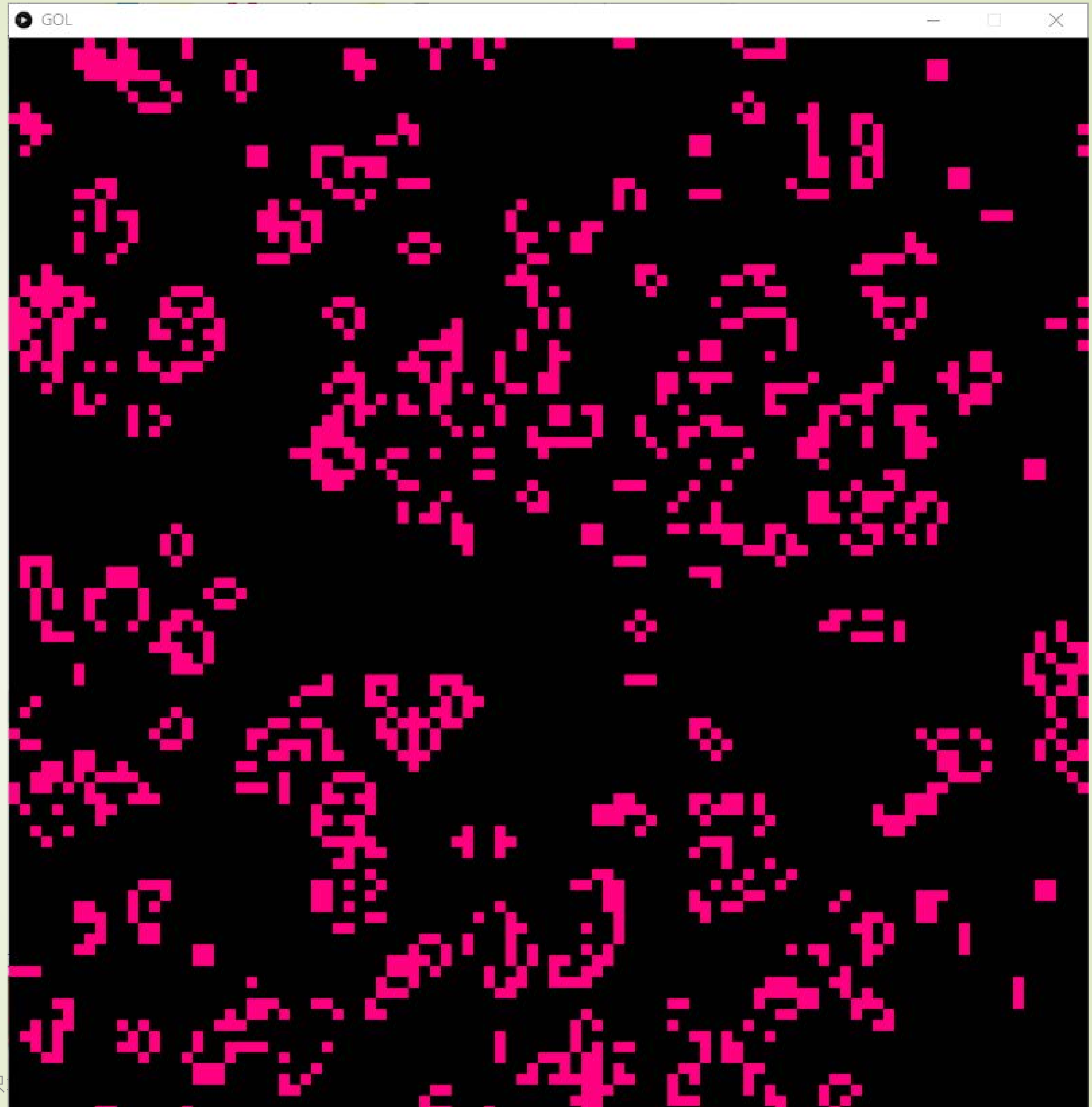
Conway’s Game of Life

- 2次元CAが興味深い挙動をする例
- セルに個体が居る／居ない
- 個体の分布の時間変化を見る
- Moore近傍を考える

NW	N	NE
W	C	E
SW	S	SE

Conway’s Game of Life
時間発展規則

- 個体が居るセル
  - 周囲の個体数が2未満あるいは３以上の場合、個体は消滅
  - それ以外は、個体は生き残る
- 個体の居ないセル
  - 周囲の個体数が3の時、新たに個体が発生


GOL