

# 知の探究

## 3.セルオートマトン

堀田 敬介

2023年10月23日(月)

# セルオートマトン

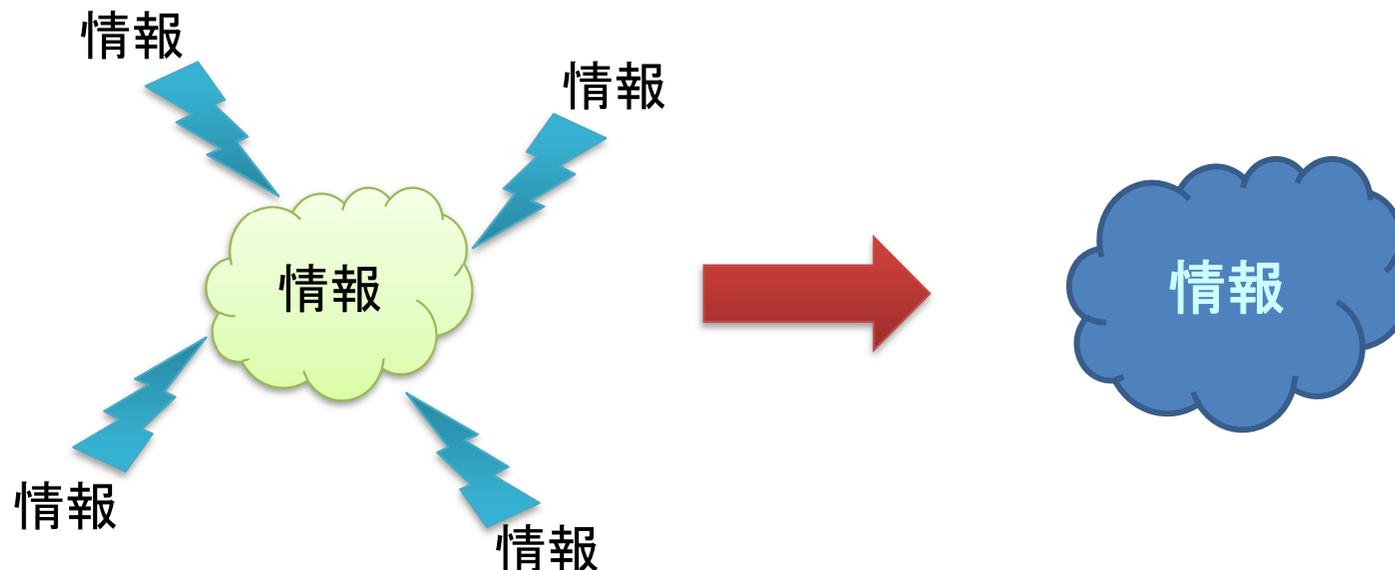
- Cellular Automata

- オートマトンとは？

「内部に何らかの情報を保持しながら、外部からの情報を入力され、その結果として情報を出力するシステム」 [1]

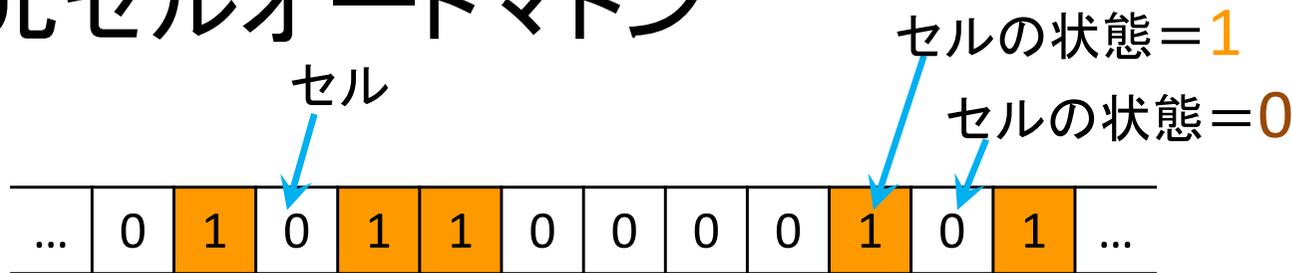
Cellular 細胞の  
Automata 自動人形(からくり人形)  
※Automaton(単数形)

※有限オートマトン(Finite Automata)  
John von Neumann(1940s)



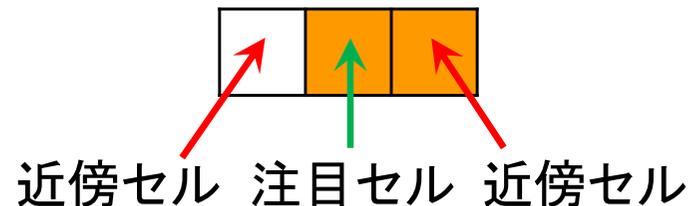
# セルオートマトン

## • 1次元セルオートマトン



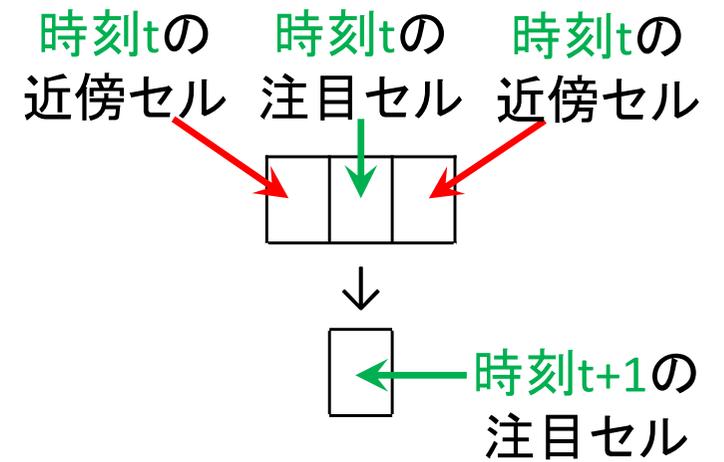
### – 現在注目している1つのセルとその近傍

- あるセル(注目セル)に対し, その両隣のセルを**近傍**とよぶ



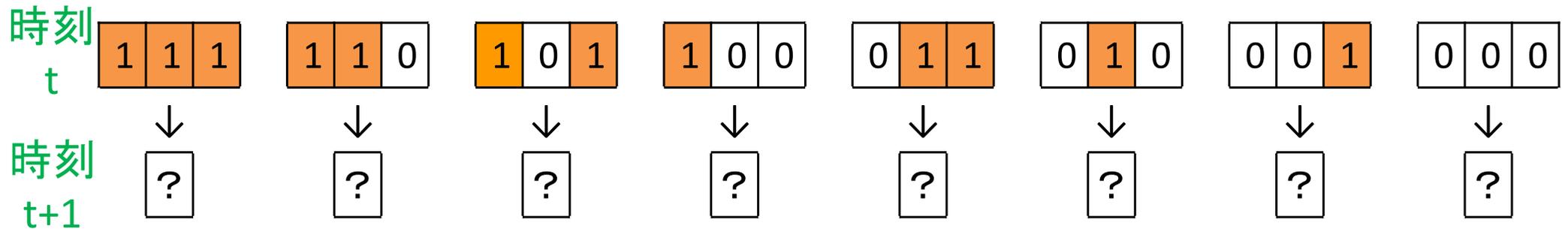
- 両端のセルの近傍の設定方法は2種類
  1. **周期**境界条件 ... 「左端セルの左側近傍は, 一番右端のセル」「右端セルの右側近傍は, 一番左端のセル」と考える(※ドーナツのイメージ)
  2. **開放**境界条件 ... 「左端セルの左側近傍は, なし」「右端セルの右側近傍は, なし」と考える

# セルオートマトン



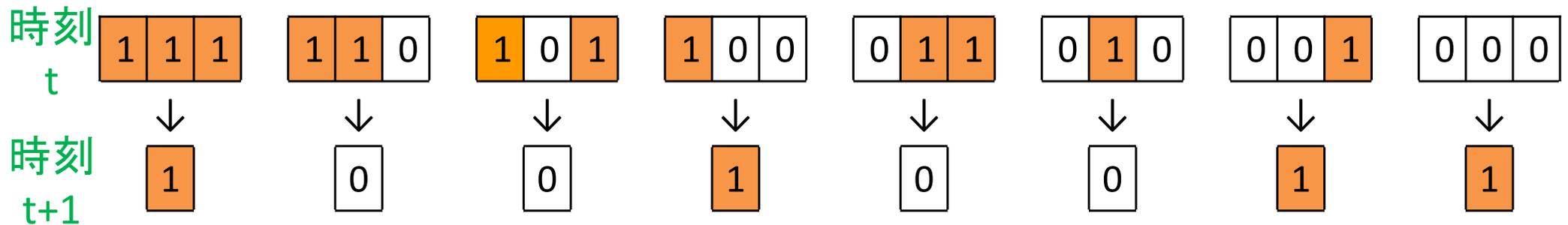
## • 1次元セルオートマトン

- 各セルは時間と共に状態が変化する
- 次の状態への変化は、自分と近傍の現在状態による  
(時刻t+1の状態は、時刻tの自分と近傍の状態による)
- 状態は「0」「1」の2値で表現するので、現在の状態の可能性は全部で8パターンとなる(なぜか?)



# セルオートマトン

- 時刻 $t \rightarrow t+1$ の状態変化の例



時刻 $t+1$ の8パターンの状態を2進数  $10010011$  と考え、  
10進数に変換すると  $(10010011)_2 = (147)_{10}$  である  
故に、この時刻 $t \rightarrow t+1$ の状態変化を「**ルール147**」とよぼう

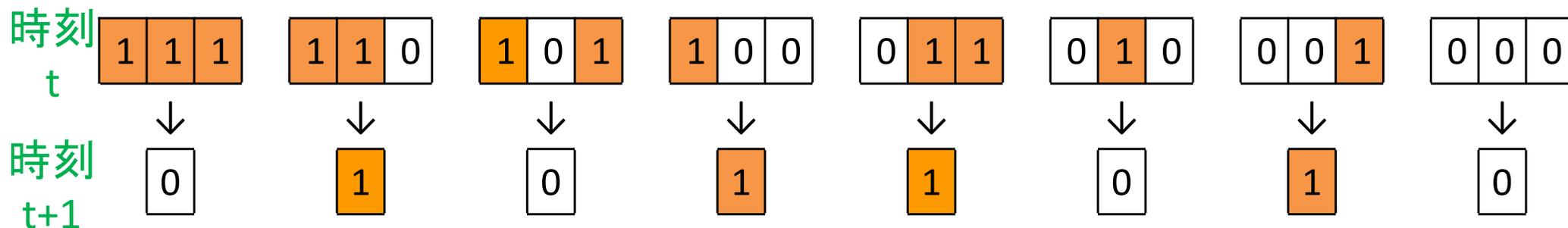
時刻 $t+1$ の考えられるパターンは全部で256個ある

なぜなら  $(00000000)_2 \sim (11111111)_2$  より  $2^8=256$ 個

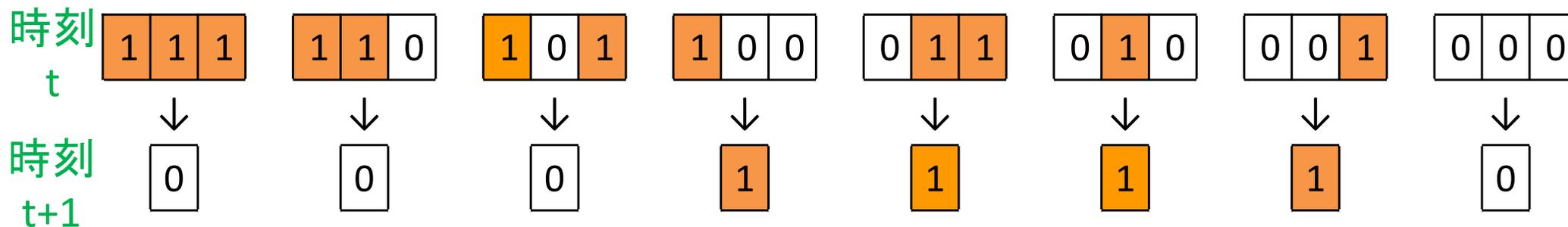
$(00000000)_2 =$ 「**ルール0**」 $\sim (11111111)_2 =$ 「**ルール255**」

# セルオートマトン

- ルールの例



➡  $(01011010)_2 = (90)_{10}$  より, これは「**ルール90**」

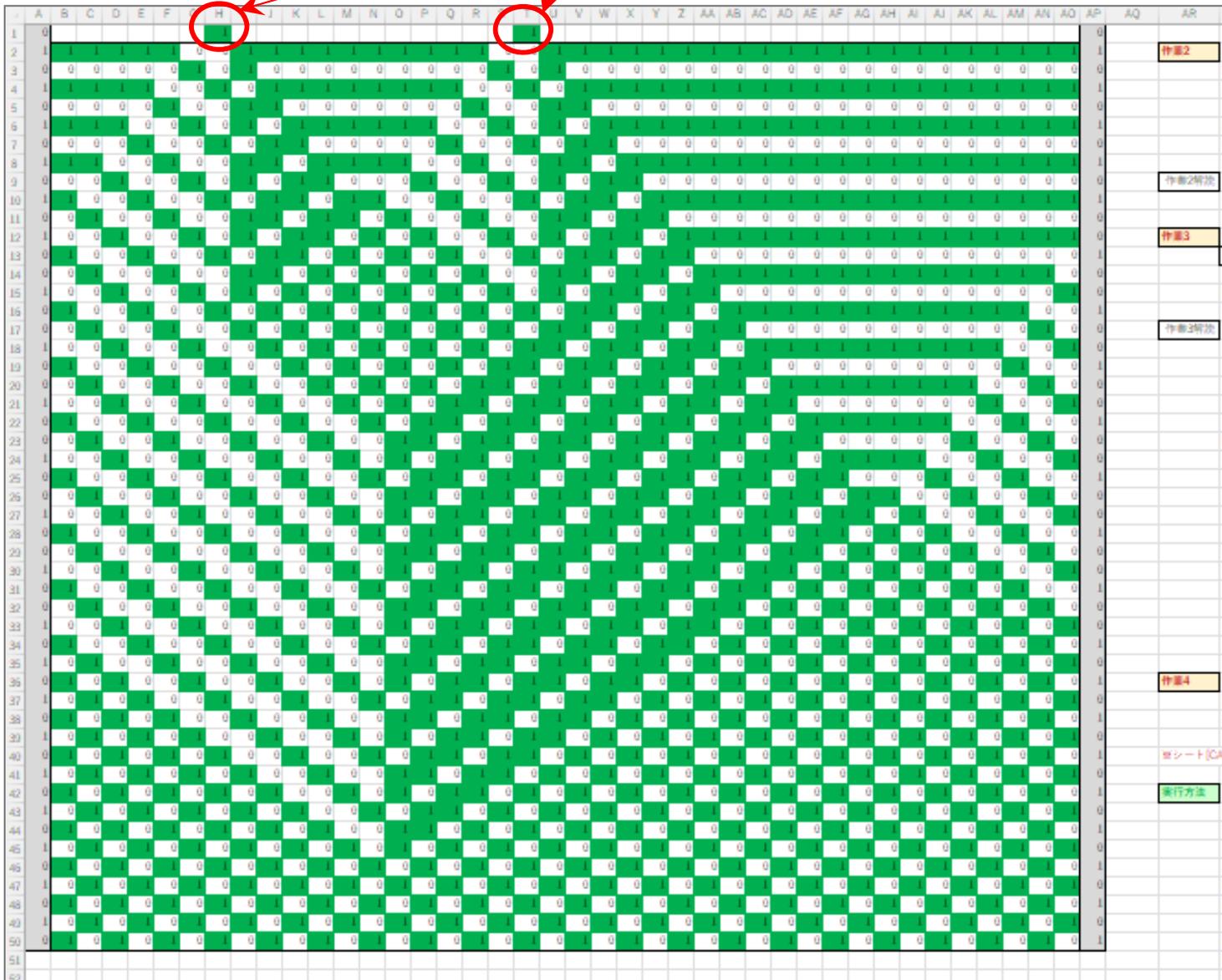


➡  $(00011110)_2 = (30)_{10}$  より, これは「**ルール30**」

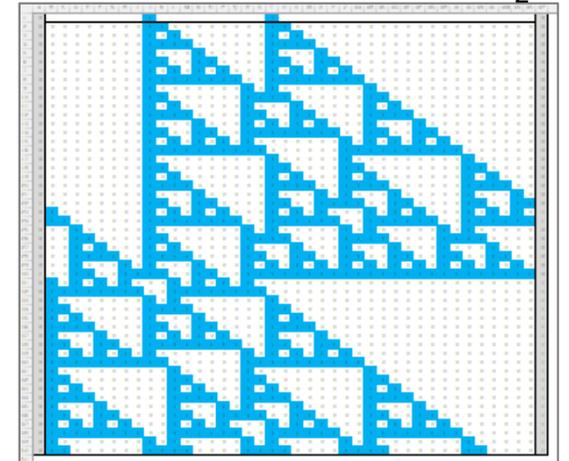
# セルオートマトン

## ● 実行例)

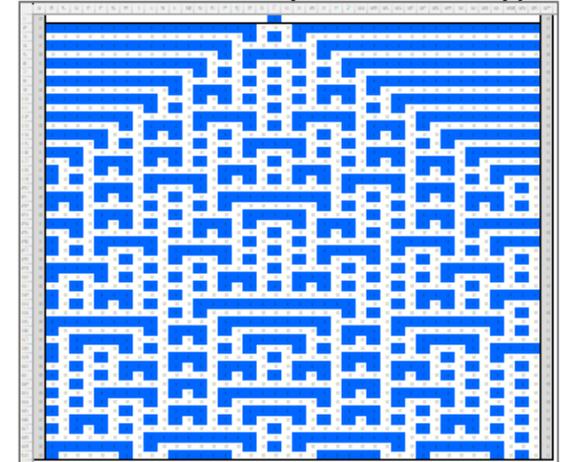
- ✓ 初期状態(時刻 $t=0$ の状態)は, セル [H1]と[T1]の2箇所が1でそれ以外は0
- ✓ 適用規則はルール57(=  $(0011\ 1001)_2$ )



- ✓ 初期状態  $[J1]=[T1]=1$
- ✓ rule60(=  $(0011\ 1100)_2$ )



- ✓ 初期状態  $[T1]=1$
- ✓ rule73(=  $(0100\ 1001)_2$ )



# セルオートマトン

- 2次元セルオートマトン

1	1	0	1	0	0	1	1	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	1
0	1	1	0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0	0	1	0	1
0	1	0	1	1	0	1	1	0	1	1	0
1	0	1	0	0	1	0	0	0	0	1	0

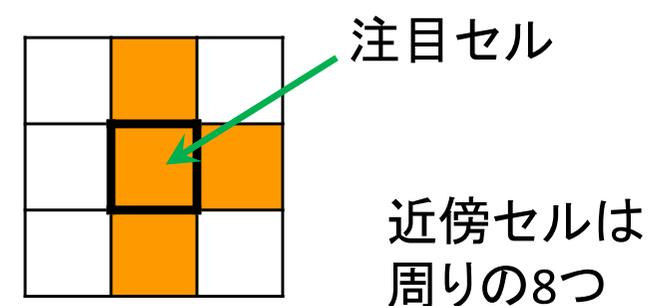
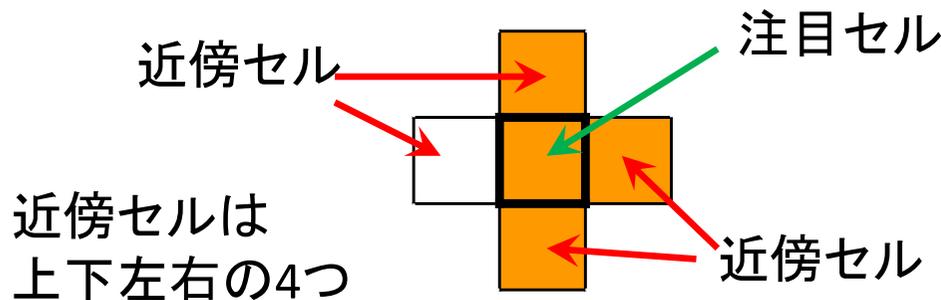
セルの状態=1  
セルの状態=0

※上下左右の端のセルの近傍は周期境界条件とする(上下左右に無限にスクロールするゲーム画面等をイメージ)

- 現在注目している1つのセルとその近傍

## ノイマン近傍

## ムーア近傍

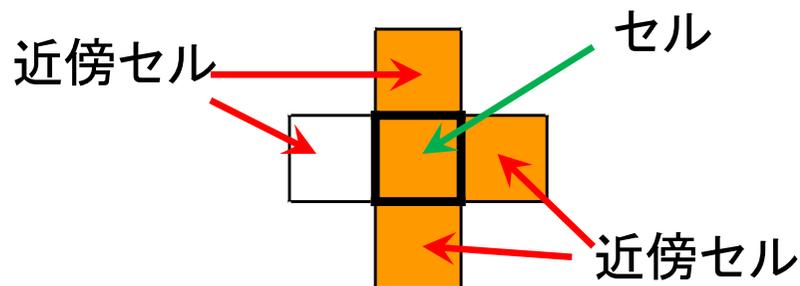


# セルオートマトン

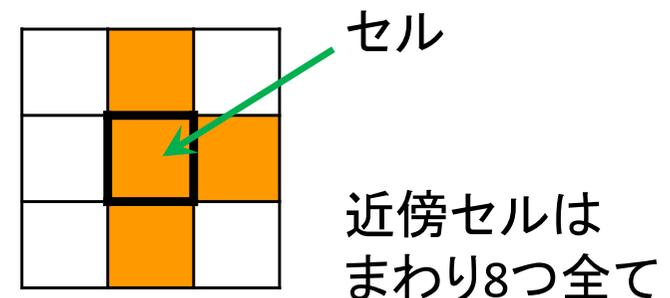
- 2次元セルオートマトン

- 各セルは時間と共に状態が変化する
- 次の状態への変化は, 自分と近傍の現在状態による  
(時刻 $t+1$ の状態は, 時刻 $t$ の自分と近傍の状態による)
- 状態は「0」「1」の2値で表現するので, 現状の可能性は  
ノイマン近傍の場合, 全部で32パターン(なぜか?)  
ムーア近傍の場合, 全部で512パターン(なぜか?)

ノイマン近傍



ムーア近傍



# セルオートマトン

## • 2次元セルオートマトンとライフゲーム

- セルの状態は生(1)と死(0)の2つとする
- ムーア近傍を使う(ただし512パターンを以下の5つに分類)
- 状態更新ルールは次の5つ
  1. 瞑想... 自セル=死(0) & 近傍セルの0-2,4-8つが生(1) → 0
  2. 誕生... 自セル=死(0) & 近傍セルの3つが生(1) → 1
  3. 生存... 自セル=生(1) & 近傍セルの2 or 3つが生(1) → 1
  4. 過疎... 自セル=生(1) & 近傍セルの0 or 1つが生(1) → 0
  5. 過密... 自セル=生(1) & 近傍セルの4~8つが生(1) → 0

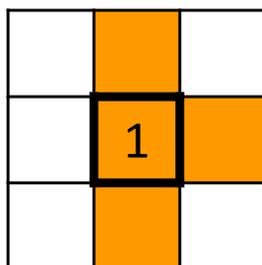
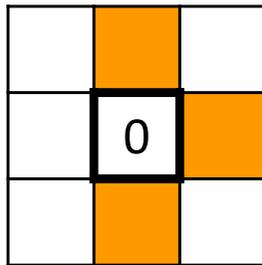
# セルオートマトン

## • ライフゲーム

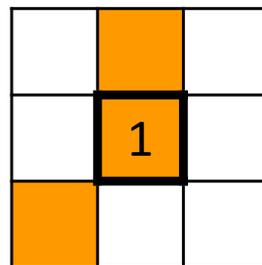
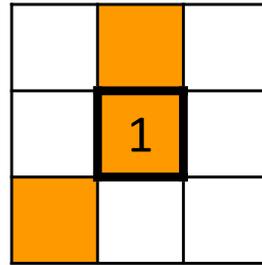
1. 瞑想... 自セル=死(0) & 近傍セルの0-2,4-8つが生(1) → 0
2. 誕生... 自セル=死(0) & 近傍セルの3つが生(1) → 1
3. 生存... 自セル=生(1) & 近傍セルの2 or 3つが生(1) → 1
4. 過疎... 自セル=生(1) & 近傍セルの0 or 1つが生(1) → 0
5. 過密... 自セル=生(1) & 近傍セルの4~8つが生(1) → 0

例)

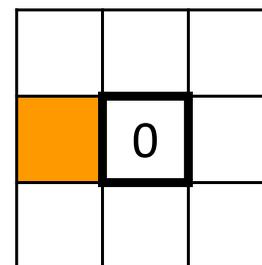
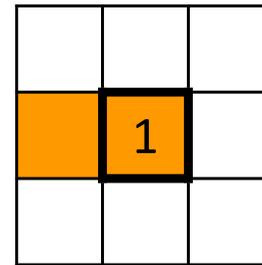
誕生



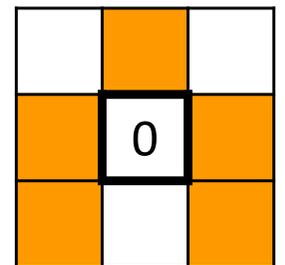
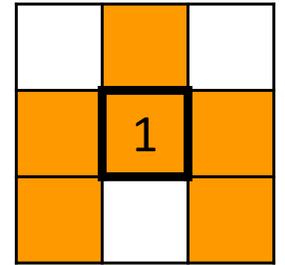
生存



過疎



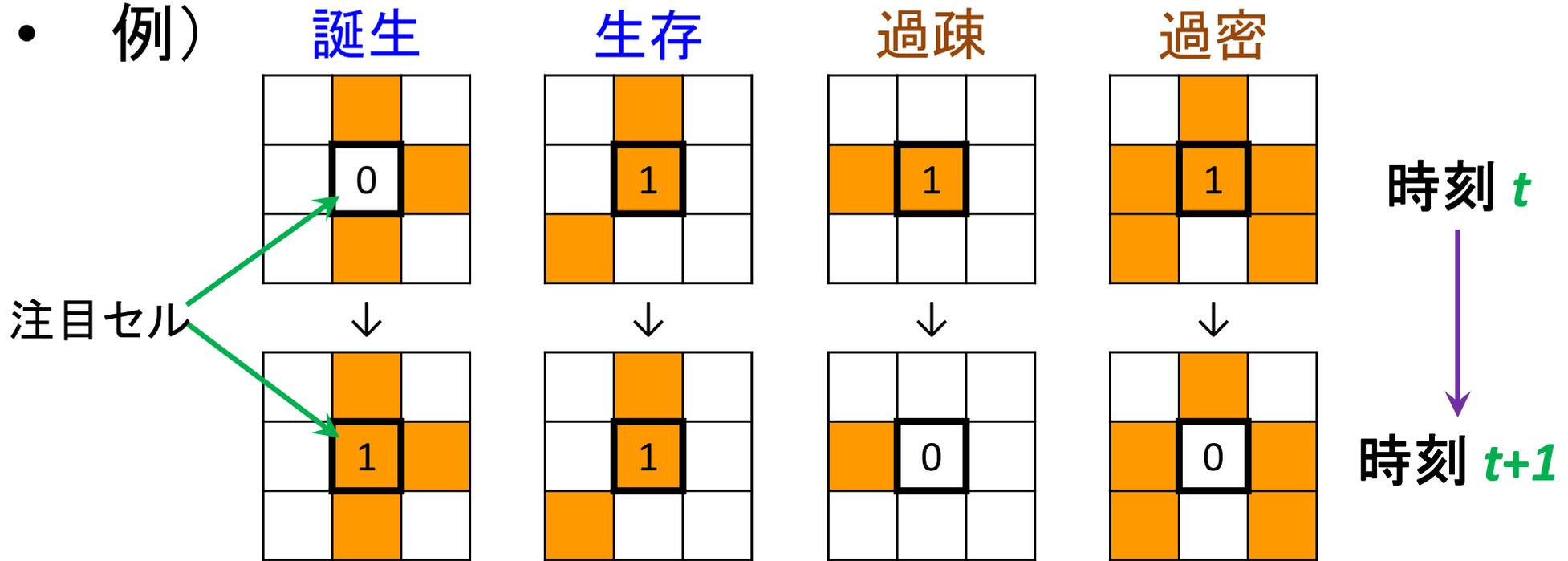
過密



※まわりの8セルがどうなるかは、それらの近傍による。ここではそのままにした

# セルオートマトン

• 例)



• ルールをまとめた「時刻  $t \rightarrow t+1$ 」の状態遷移表

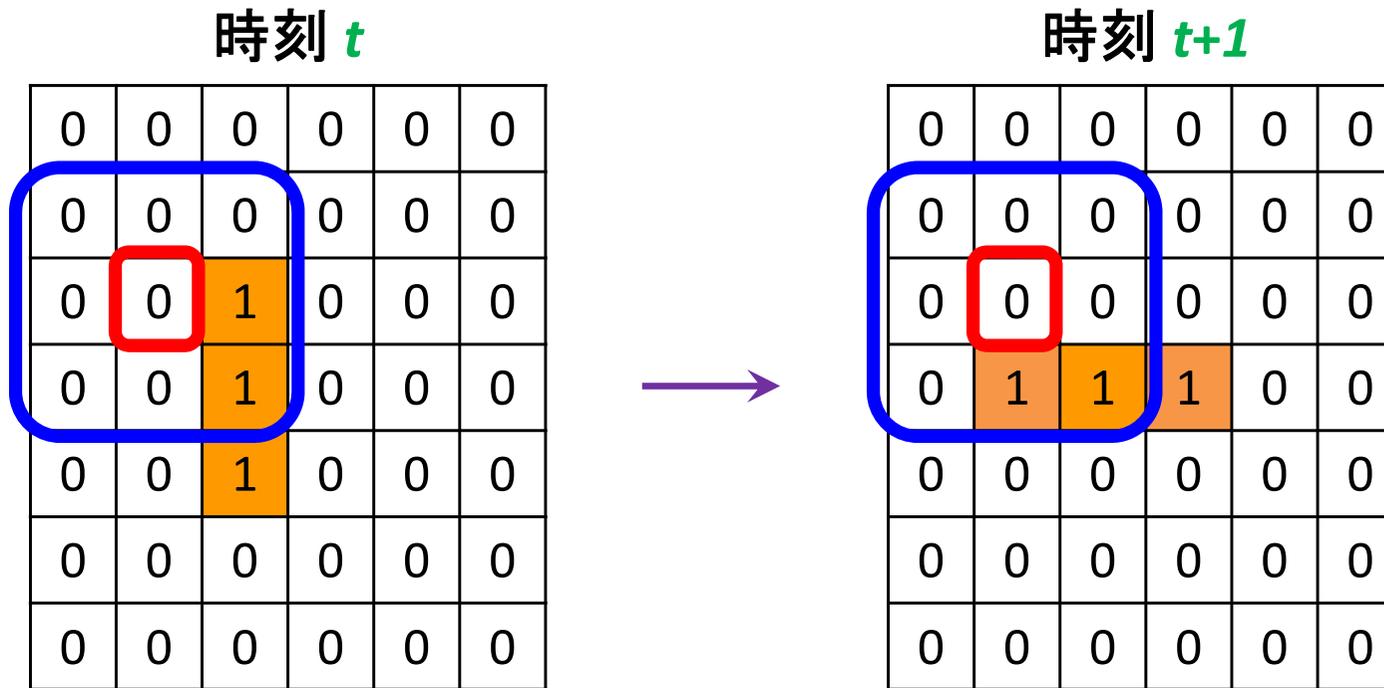
–  $s =$  近傍セルの状態の合計 (= 生存数) とすると

		$s =$ 時刻 $t$ の近傍セルの状態の合計 (= 生存数)								
		0	1	2	3	4	5	6	7	8
時刻 $t$ の注目セルの状態	0	0	0	0	1	0	0	0	0	0
	1	0	0	1	1	0	0	0	0	0

時刻  $t+1$  の注目セルの状態

# セルオートマトン

- 例)



- ルールをまとめた「時刻  $t \rightarrow t+1$ 」の状態遷移表

		s (近傍セルの状態合計=生存数)								
		0	1	2	3	4	5	6	7	8
自セル	0	0	0	0	1	0	0	0	0	0
	1	0	0	1	1	0	0	0	0	0

# Life Game : 実行例

- ✓ 上部に[グライダー銃], 左中央に[ペンタデカスロン], 左下に[パルサー]を配置して実行

Update

ライフゲームの実行用の例

プリンカー 時計 消滅物体 プリンカー×4へ

パルサー

ペンタデカスロン

動く物体の例

グライダー 宇宙船 (軽量級) 宇宙船 (中量級)

宇宙船 (重量級)

作業3 書式 1 | 2 |

作業4 周期 1 | 2 |

作業4解記 前回 1 | 2 | 3 | 4 |

作業5 Exce 0 | 1 | 2 | 3 | 4 | 5 |

作業6 Exce 1 | 2 | 3 | 4 | 5 |

作業7 Visu 1 | 2 |

<各命令文の意味>

シート[Update]のセル範囲[B2:AE31]をコピーせよ

シート[CA]を選択せよ

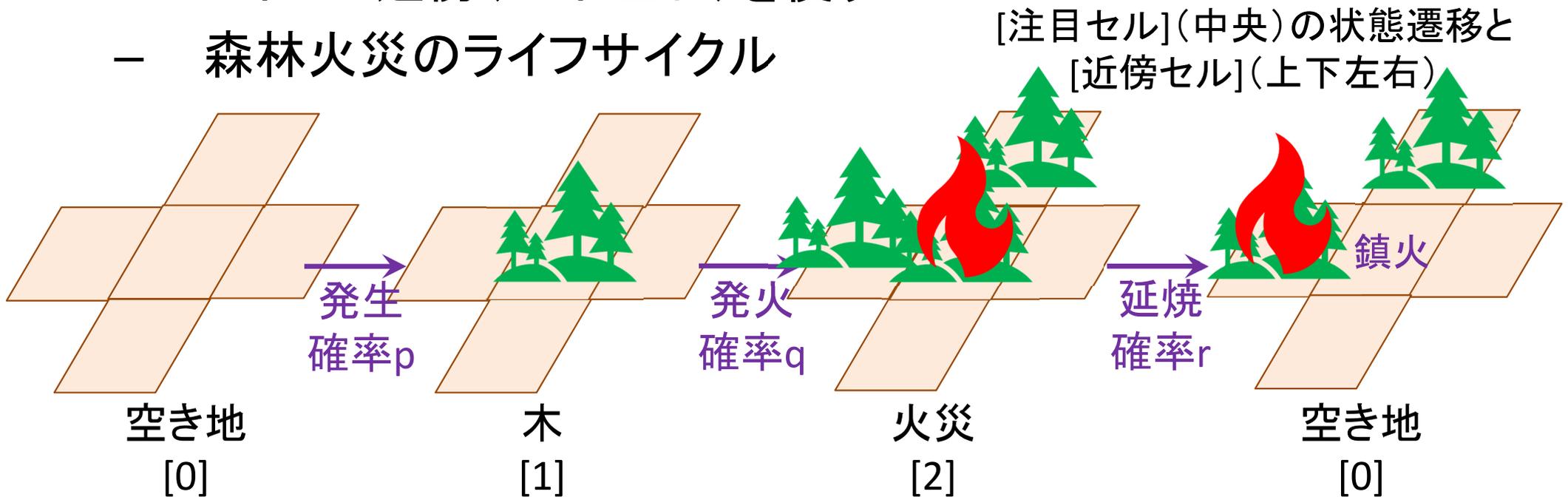
(既に選択しているシート[CA]の)セル範囲[B2]を選択せよ

命令で, [Paste:=xl... ]以下は, 命令を実行する際の詳細なオプション設定 [Selection]=今選んでいるもの, を, [PasteSpecial]貼り付けよ

# セルオートマトン

## • 2次元セルオートマトンと森林火災シミュ

- セルの状態は0(空き地),1(木),2(火災)の3つ
- ノイマン近傍(上下左右)を使う
- 森林火災のライフサイクル

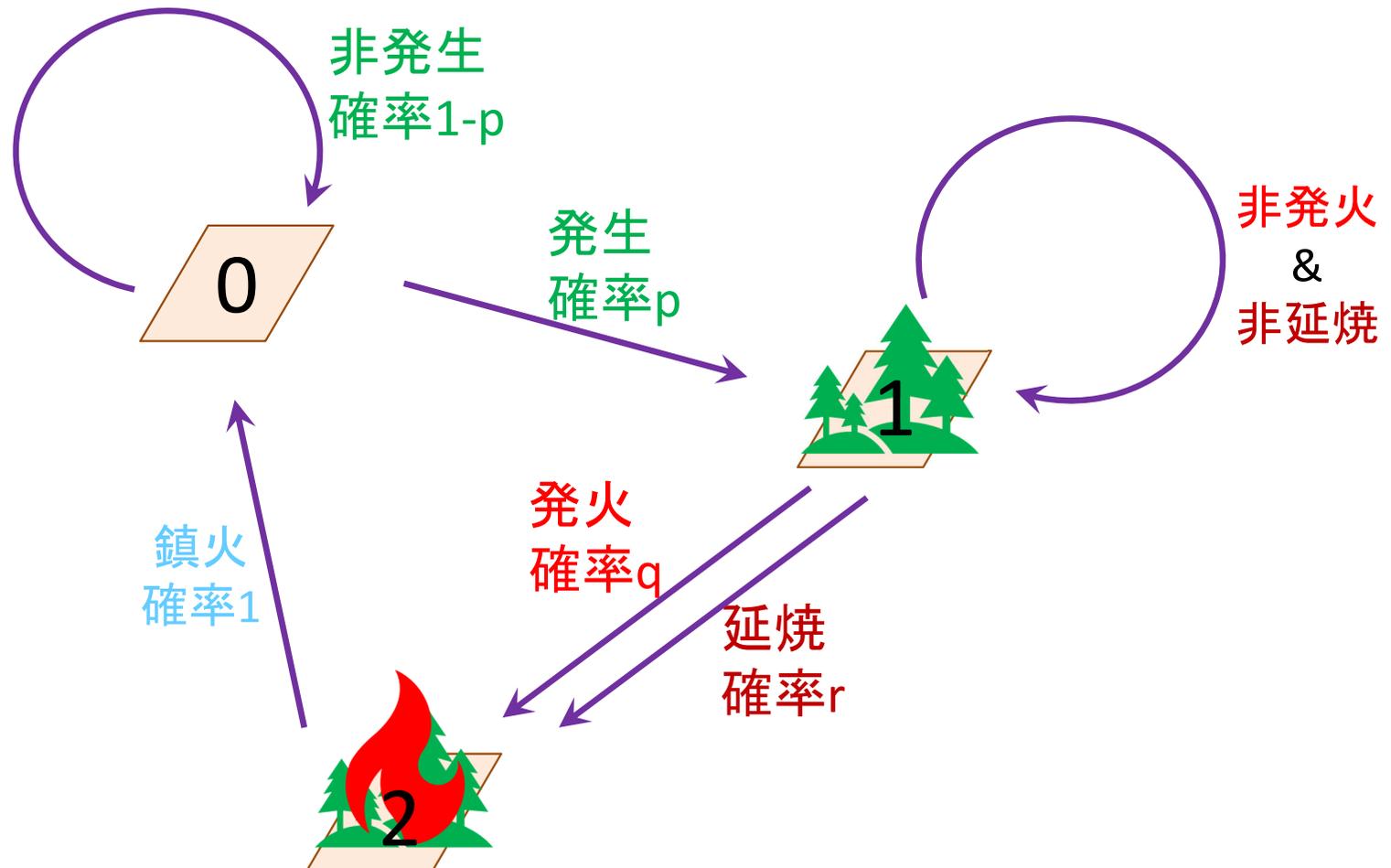


- 空き地[0]の状態では、一定確率 $p$ で木が発生
- 木[1]の状態かつ近傍全部非火災では、一定確率 $q$ で発火
- 火災[2]状態の近傍の木[1]は、一定確率 $r$ で延焼
- 火災[2]状態の木は、時刻1期後に燃え尽き(鎮火)空き地[0]に

# セルオートマトン

- 森林火災シミュレーション

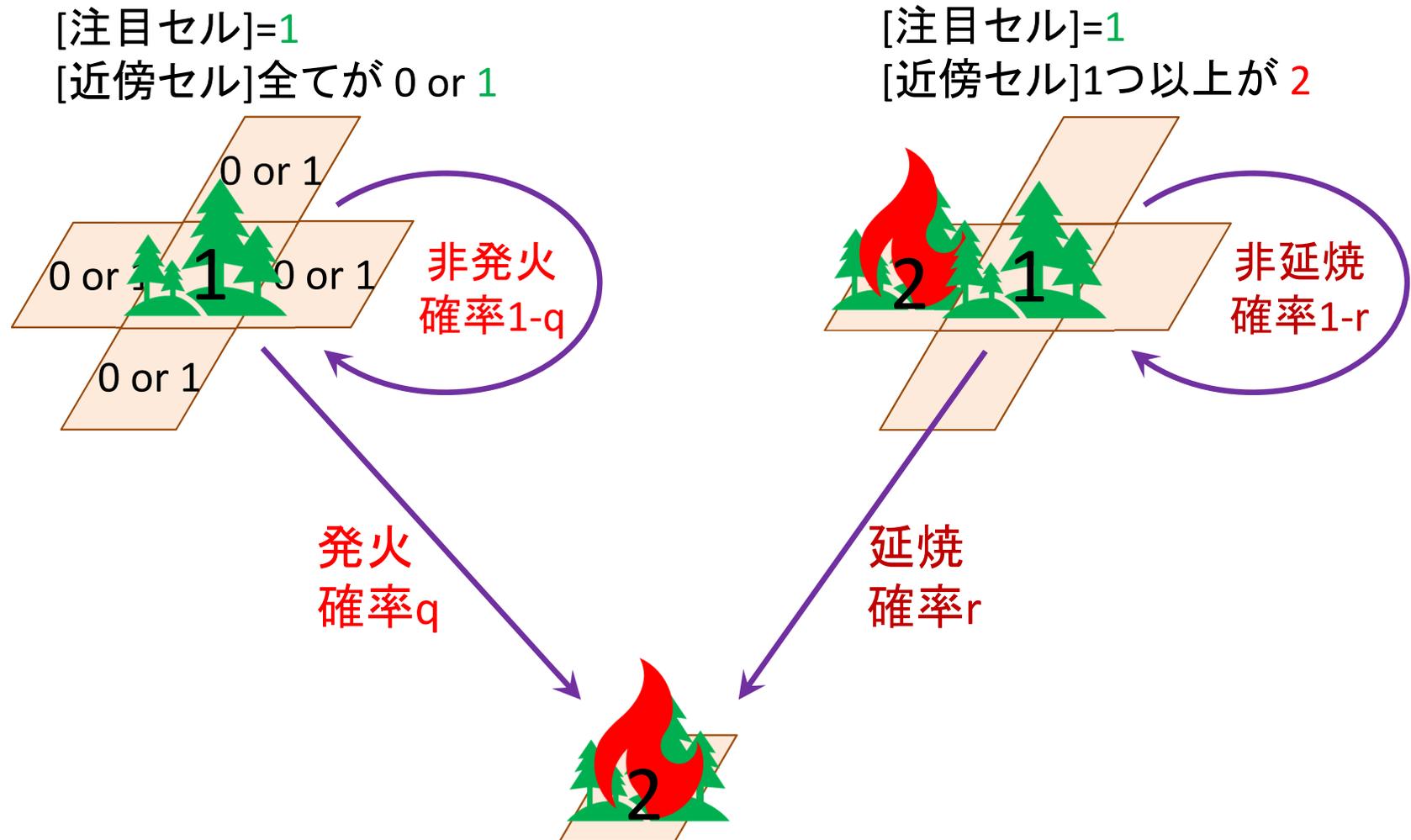
- 状態遷移図, life cycle



# セルオートマトン

- 森林火災シミュレーション

- 状態遷移図(木[1] → 火災[2])の部分の詳細

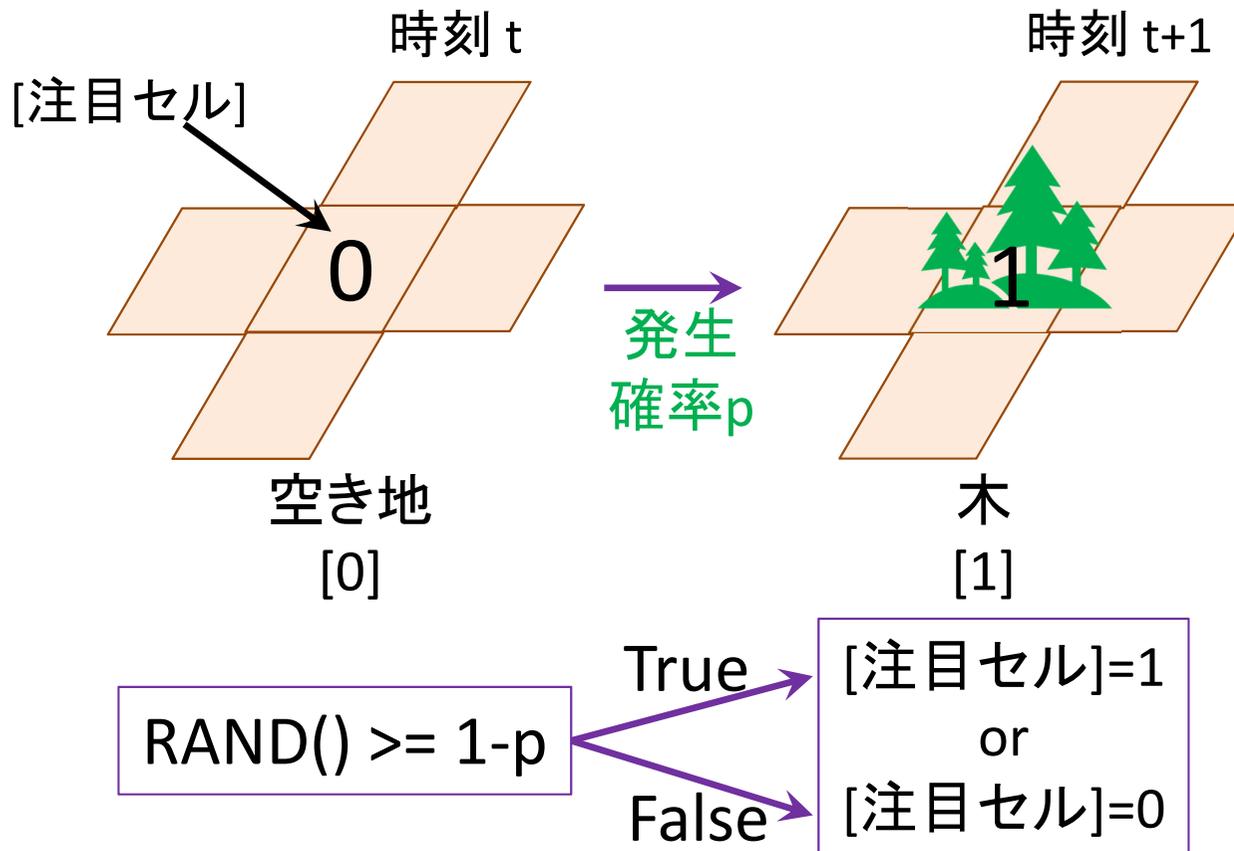


# セルオートマトン

RAND() は,  $[0,1)$ -  
一様擬似乱数を  
生成する関数

## 森林火災シミュレーション: 発生

- [注目セル]=0(空き地)の状態では, 一定確率 $p$ で木が発生



範囲の書き方  
 $[0,1)$  ... 0以上1未満  
 $[0,1]$  ... 0以上1以下  
 $(0,1]$  ... 0より大1以下  
 $(0,1)$  ... 0より大1未満

$RAND() \geq 1-p$

0                      1-p                      1

●                      False                      True                      ○

(1-p)以上の範囲

例えば  $p=0.05$  なら,  
生成した乱数値がこ  
の範囲に入る確率が  
0.05 (5%) となる

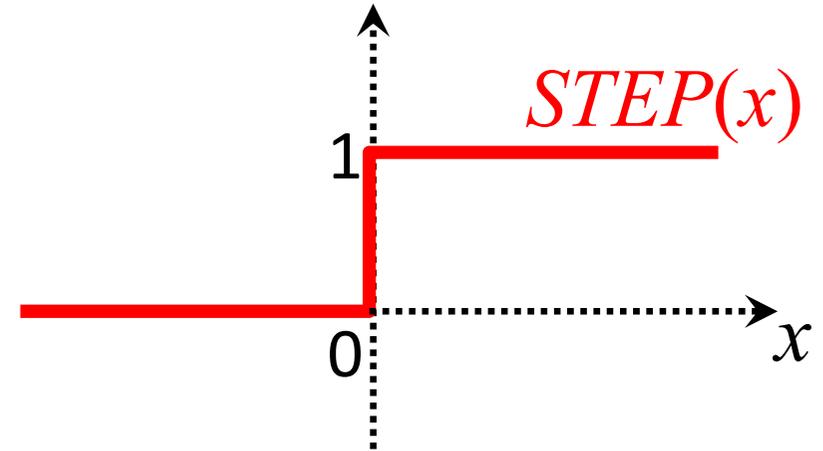
この仕組みを実現する式  
時刻 $t+1$ の[注目セル] =  $IF( RAND() \geq 1-p, 1, 0 )$

# 補足：ステップ関数

階段(steps)を1段上げるイメージ  
 $x \geq 0$  のときは1(段)にいる  
 $x < 0$  のときは0(段)にいる

- ステップ関数(階段関数)

$$STEP(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases}$$



↑階段を横から見てるイメージ

– IF文をSTEP関数を使って等価な式に書き換える

$$\text{IF}( \text{RAND}() \geq 1-p, 1, 0 )$$

$$= \text{IF}( \text{RAND}() - 1 + p \geq 0, 1, 0 )$$

$$= \text{STEP}( \text{RAND}() - 1 + p )$$

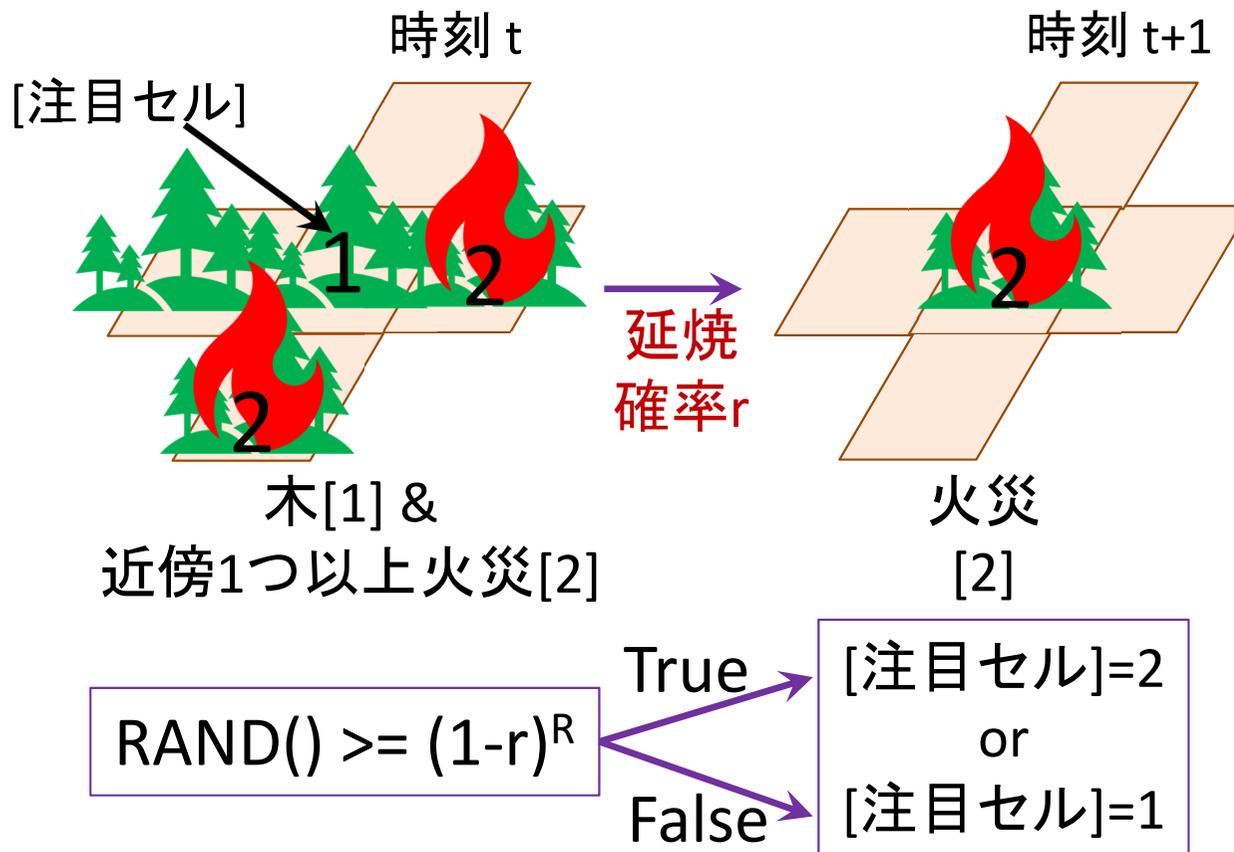
※簡単に  $\text{IF}( \text{RAND}() < p, 1, 0 )$  としないのはSTEP関数へ変換したいため(条件式を非負にしておきたいから)



# セルオートマトン

- 森林火災シミュレーション: **延焼**

- [注目セル]=1(木)&近傍1つ以上火災では, 一定確率 $r$ で**延焼**



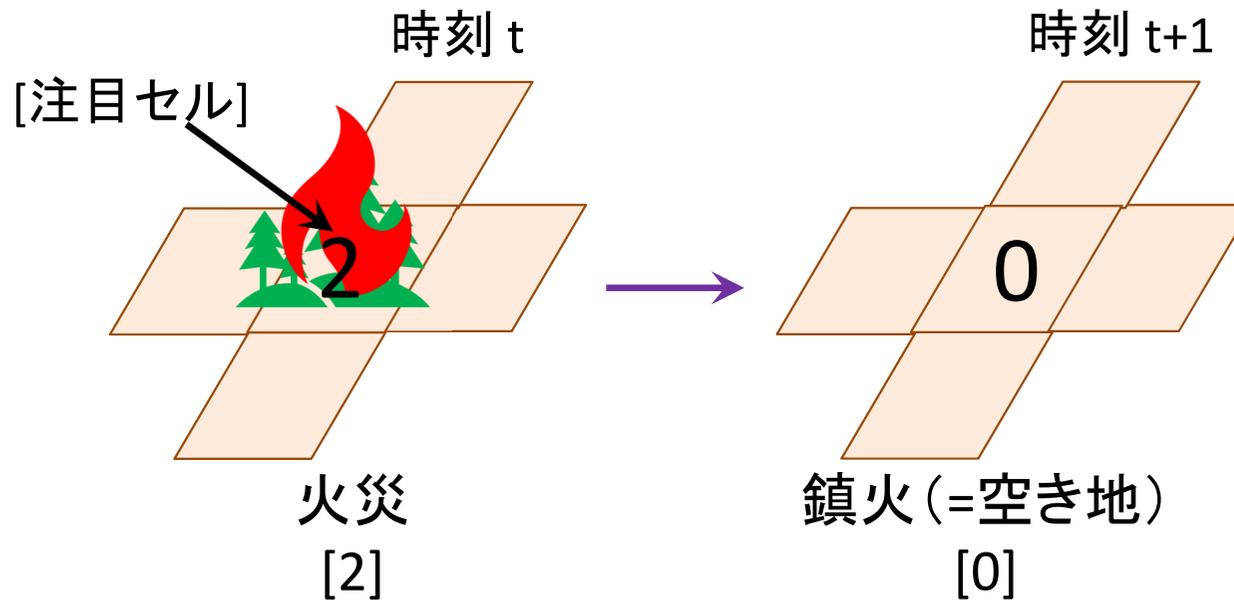
この仕組みを実現する式

$$\text{時刻}t+1\text{の}[\text{注目セル}] = \text{IF}(\text{RAND}() \geq (1-r)^R, 2, 1) = 1 + \text{STEP}(\text{RAND}() - (1-r)^R)$$

# セルオートマトン

- 森林火災シミュレーション: 鎮火

- [注目セル]=2(火災) 状態は, 1期後に(必ず)鎮火



[注目セル]=2

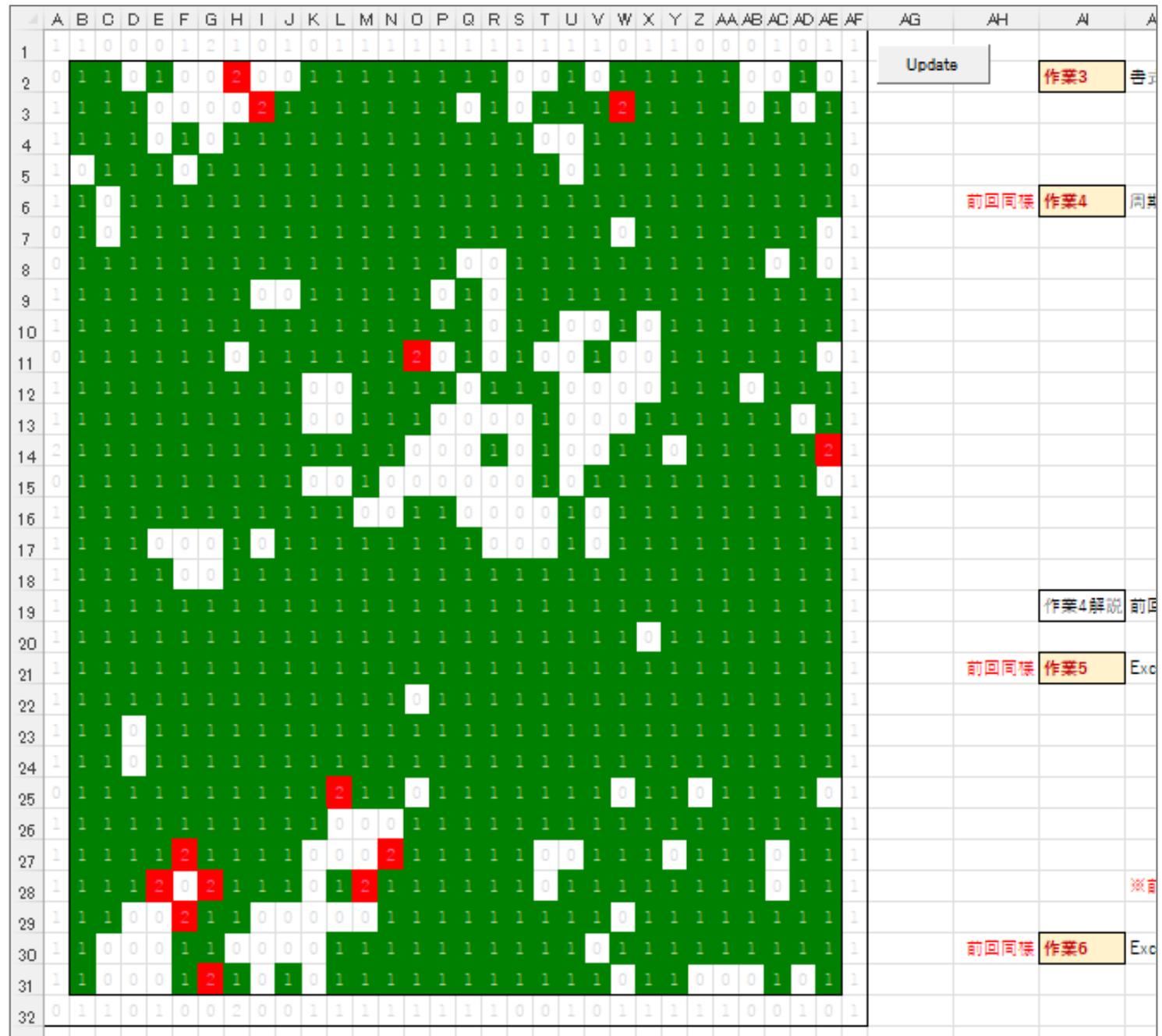
[注目セル]=0

# セルオートマトン

- 森林火災

- 実行例

- ✓ 発生確率  $p=0.05$
  - ✓ 発火確率  $q=0.001$
  - ✓ 延焼確率  $r=0.4$
- で実行



# セルオートマトン

## • 2次元セルオートマトンと液体分離シミュ

- 混ざり合わない2つの液体(水と油など)を1つの容器内で強引に混ぜ合わせた後, 時間と共に分離する様子をシミュレーションする

✓ セルの状態は-1.0~+1.0の連続量

例) 水と油の場合の状態量の意味

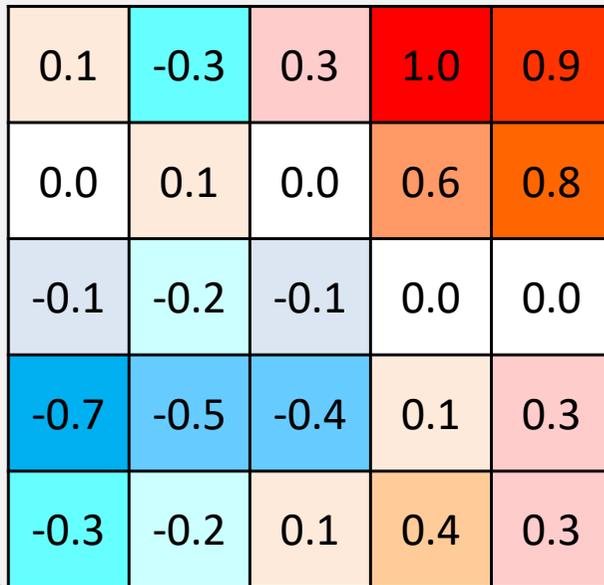
0.1	-0.3	0.3	1.0	0.9
0.0	0.1	0.0	0.6	0.8
-0.1	-0.2	-0.1	0.0	0.0
-0.7	-0.5	-0.4	0.1	0.3
-0.3	-0.2	0.1	0.4	0.3

容器を上から見た様子

状態量	
-1.0	水のみ液体
-0.7	水多量+油少量の混合液
-0.3	水やや多+油やや少の混合液
0.0	水と油が半々の混合液
+0.3	水やや少+油やや多の混合液
+0.7	水少量+油多量の混合液
+1.0	油のみ液体

# セルオートマトン

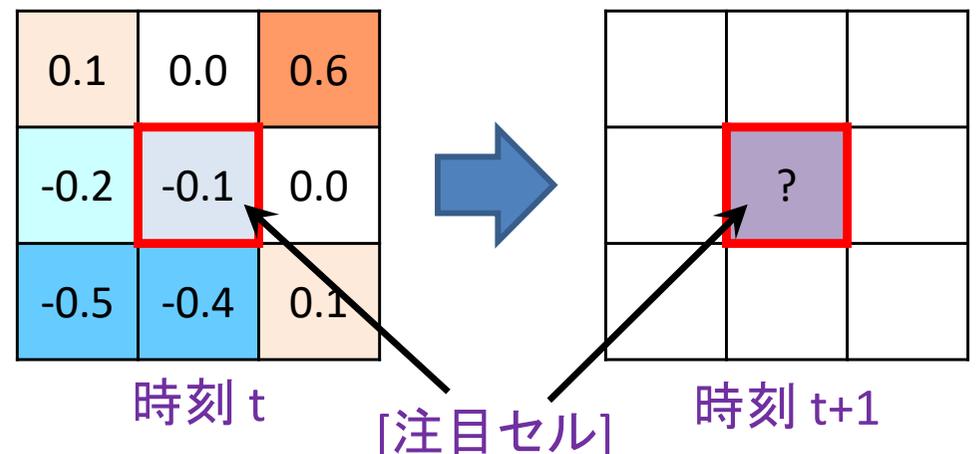
- 2次元セルオートマトンと液体分離シミュ
- 混ぜり合わない2つの液体(水と油など)を1つの容器内で強引に混ぜ合わせた後, 時間と共に分離する様子をシミュレーションする



0.1	-0.3	0.3	1.0	0.9
0.0	0.1	0.0	0.6	0.8
-0.1	-0.2	-0.1	0.0	0.0
-0.7	-0.5	-0.4	0.1	0.3
-0.3	-0.2	0.1	0.4	0.3

容器を上から見た様子

- ✓ ムーア近傍(8方向近傍)を使う
- ✓ 周期境界条件とする
- ✓ 時刻 $t$ の[注目セル]&[近傍セル8つ]の状態量から, 時刻 $t+1$ の[注目セル]の状態量が決まる



# セルオートマトン

## • 2次元セルオートマトンと液体分離シミュ

- 時刻 $t$ の[注目セル]の状態量 $s^t$ 及び[8近傍セル]の状態量 $a, b, c, d, w, x, y, z$ から, 時刻 $t+1$ の[注目セル]の状態量 $s^{t+1}$ を決定

$$s^{t+1} = \mu \cdot \tanh(s^t) + \delta \cdot T$$

$$T = \frac{1}{6}\{a + b + c + d\} + \frac{1}{12}\{w + x + y + z\} - s^t$$
$$= \frac{2a + 2b + 2c + 2d + w + x + y + z}{12} - s^t$$

※ $\tanh()$  は双曲線正接関数 Hyperbolic tangent function 入力値を[-1.0~1.0]の範囲の数値に変換する

※ $\mu$ と $\delta$ はパラメータ(事前に決める定数)  
ex)  $\mu = 1.3$ ,  $\delta = 0.5$

※ $T$ は「8近傍の状態の加重平均」と「注目セルの状態」との差

$s^t \in [-1.0, 1.0]$  と,

$[\tanh(-1), \tanh(1)] = [-0.761, 0.761]$  より

$\mu = 1.3$  に対し,

$[\mu \cdot \tanh(-1), \mu \cdot \tanh(1)] = [-0.990, 0.990]$

$a, b, c, d, w, x, y, z, s^t \in [-1.0, 1.0]$  より,

$T \in [-2.0, 2.0]$  なので,

$\delta = 0.5$  に対し,  $\delta \cdot T \in [-1.0, 1.0]$

従って,  $s^{t+1} \in [-1.0, 1.0]$  である

※  $s^{t+1}$  の第1項と第2項にある  $s^t$  の正負が逆であることに注意

時刻  $t$

$w$	$a$	$x$
$b$	$s^t$	$d$
$y$	$c$	$z$



時刻  $t+1$

	$s^{t+1}$	

# セルオートマトン

- 液体分離  
- 実行例)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH		
1	0.5	-0.5	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.6	0.6	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.5	-0.5	1.3 = $\mu$			
2	0.5	-0.5	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.6	0.1	0.7	0.9	1.0	1.0	1.0	1.0	0.9	0.8	0.5	-0.5	0.5 = $d$			
3	0.4	-0.5	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-0.8	-0.7	-0.7	0.6	0.8	0.9	1.0	1.0	1.0	1.0	0.9	0.8	0.4	-0.5				
4	0.4	-0.6	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.8	-0.7	-0.6	-0.6	-0.5	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.4	-0.5				
5	0.2	-0.7	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.8	-0.7	-0.6	-0.5	-0.5	-0.5	-0.5	-0.4	-0.3	-0.3	0.3	0.6	0.8	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.2	-0.5				
6	-0.1	-0.8	-0.9	-1.0	-1.0	-1.0	-0.9	-0.8	-0.8	-0.7	0.1	0.3	0.4	0.4	0.5	0.5	0.5	0.6	0.8	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.2	-0.5				
7	-0.1	-0.8	-0.9	-1.0	-1.0	-0.9	-0.8	-0.7	0.1	0.6	0.7	0.8	0.8	0.8	0.8	0.9	0.9	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	-0.5	-0.5			
8	-0.1	-0.9	-1.0	-1.0	-0.9	-0.8	-0.7	0.3	0.7	0.9	0.9	0.9	0.9	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.2	-0.6	-0.5				
9	-0.1	-0.9	-1.0	-1.0	-0.9	-0.7	0.5	0.7	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.2	-0.8	-0.5				
10	-0.1	-0.9	-1.0	-0.9	-0.8	-0.5	0.4	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	-0.2	-0.8	-0.5			
11	-0.1	-1.0	-1.0	-0.9	-0.8	-0.4	0.5	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.3	-0.2	-0.9	-1.0			
12	-0.1	-1.0	-1.0	-0.9	-0.8	-0.5	0.6	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	-0.1	-0.7	-0.9	-1.0			
13	-0.1	-1.0	-1.0	-0.9	-0.8	-0.4	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.3	-0.2	-0.9	-1.0			
14	-0.1	-1.0	-1.0	-0.9	-0.8	-0.2	0.7	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.4	-0.2	-0.9	-1.0			
15	-0.1	-1.0	-1.0	-0.9	-0.8	-0.4	0.4	0.7	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.4	-0.2	-0.9	-1.0			
16	-0.1	-1.0	-1.0	-0.9	-0.9	-0.7	-0.4	0.1	0.4	0.5	0.5	0.6	0.6	0.6	0.6	0.7	0.8	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.2	-0.7	-0.9	-1.0				
17	-0.1	-0.9	-0.9	-1.0	-1.0	-0.9	-0.9	-0.8	-0.7	-0.5	-0.5	-0.4	-0.4	-0.4	-0.3	0.0	0.3	0.6	0.8	0.9	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.2	-0.2	-0.6	-0.8	-0.5			
18	-0.1	-0.8	-0.9	-0.9	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-0.8	-0.8	-0.8	-0.8	-0.8	-0.7	-0.6	-0.5	-0.3	0.3	0.7	0.9	0.9	1.0	1.0	1.0	1.0	0.9	0.9	0.6	0.0	-0.6	-0.8	-0.5		
19	0.2	-0.4	-0.7	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.8	-0.5	0.1	0.7	0.9	1.0	1.0	1.0	1.0	0.9	0.8	0.6	0.2	-0.4			
20	0.7	0.5	0.5	-0.6	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.8	-0.5	0.4	0.8	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.5				
21	0.9	0.8	0.6	-0.1	-0.7	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.7	-0.5	0.7	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.8				
22	0.9	0.9	0.7	0.2	-0.6	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.4	0.6	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9				
23	1.0	0.9	0.7	0.2	-0.7	-0.9	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.5	0.4	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9				
24	0.9	0.9	0.6	-0.2	-0.8	-0.9	-1.0	-1.0	-0.9	-0.9	-0.7	-0.6	-0.7	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.6	0.3	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9				
25	0.9	0.7	0.2	-0.6	-0.9	-0.9	-1.0	-1.0	-0.9	-0.8	-0.6	-0.5	-0.5	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.7	0.0	0.7	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7				
26	0.8	0.5	-0.1	-0.8	-0.9	-1.0	-1.0	-1.0	-0.9	-0.9	-0.7	-0.5	-0.6	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.5	0.6	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5					
27	0.7	0.2	-0.6	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.8	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.5	0.5	0.9	0.9	1.0	1.0	1.0	1.0	0.9	0.7	0.2					
28	0.6	-0.2	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	0.4	0.8	0.9	1.0	1.0	1.0	1.0	0.9	0.6	0.2					
29	0.6	-0.4	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	0.3	0.8	0.9	1.0	1.0	1.0	1.0	0.9	0.6	0.4					
30	0.5	-0.5	-0.8	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.9	-0.5	0.4	0.8	0.9	1.0	1.0	1.0	0.9	0.5	0.5					
31	0.5	-0.5	-0.9	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.5	0.6	0.9	1.0	1.0	1.0	0.9	0.9	0.5	0.5					
32	0.5	-0.5	-0.9	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.6	0.1	0.7	0.9	1.0	1.0	1.0	0.9	0.8	0.5	-0.5				

# セルオートマトン

## • 交通流・交通渋滞シミュレーション(1)

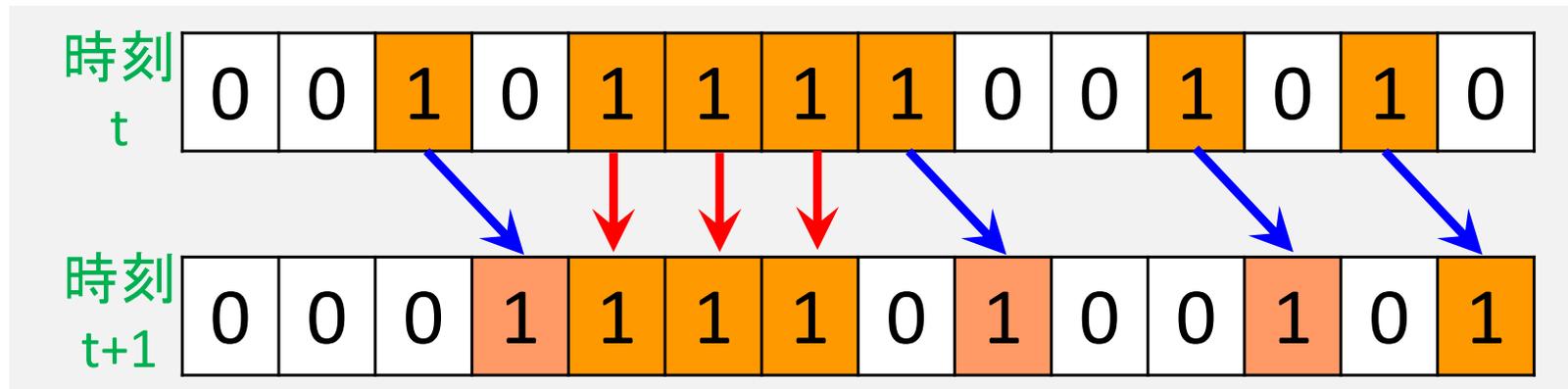
### – 車両の進行規則

1. 速度は一定で, 前方が空いている時は前へ進む
2. 前方が詰まっている(渋滞)時はその場に留まる



### – 1次元セルオートマトンで実現

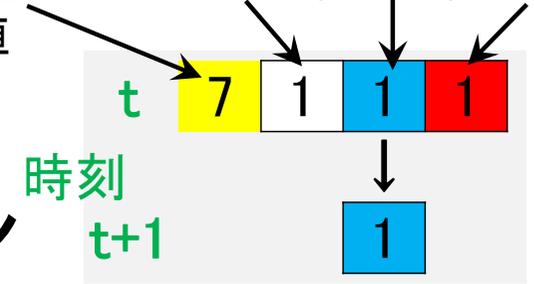
- 道路上に車両が居る状態を 1, 居ない状態を 0とする



# セルオートマトン

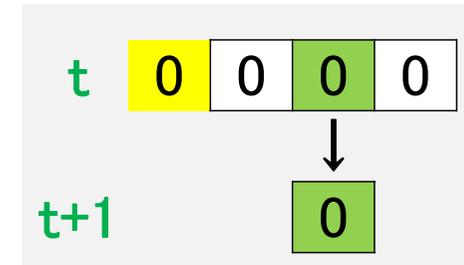
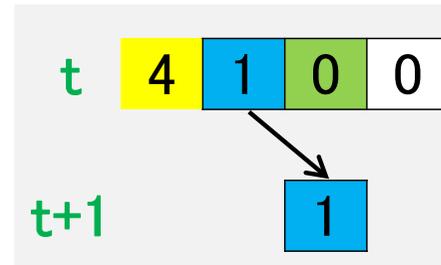
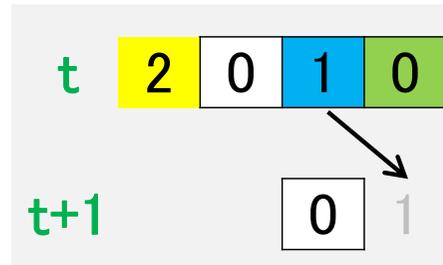
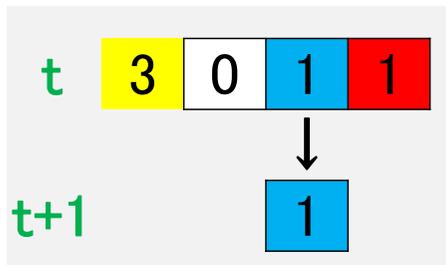
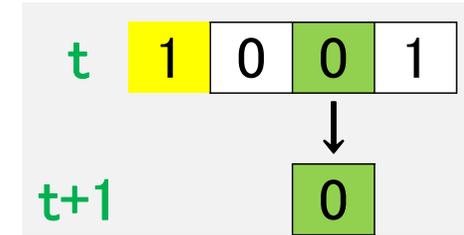
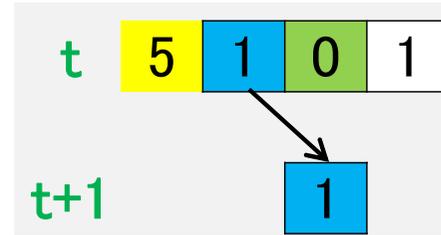
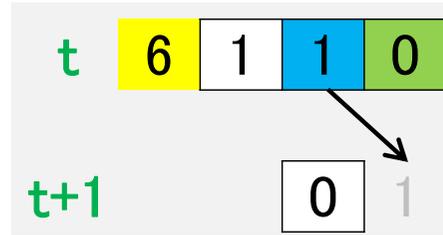
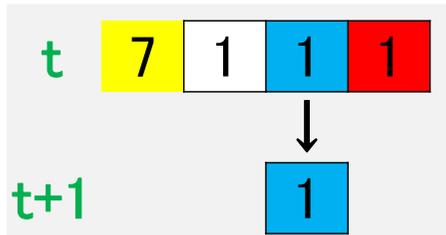
※3桁の2進数(111)を10進数に変換した値

[左近傍 | 注目 | 右近傍]



## 交通流・交通渋滞シミュレーション

「車両の進行規則」を1次元セルオートマトンの「ルール」へ

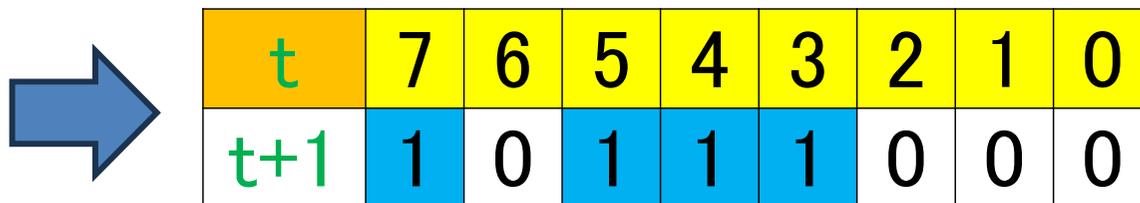


前方(右近傍)が詰まっている(渋滞)ので「注目セル」の車は動かない(動けない)

前方(右近傍)が空いているので「注目セル」の車が前方に進み、道路が空く

前方(注目セル)が空いているので、「左近傍」の車が前方(注目セル)に進む

変化なし ※空き道路(注目セル)が空きのまま



ルール184

(= (1011 1000)<sub>2</sub>)

交通シミュレーションの基本モデル



# セルオートマトン

## • 交通流・交通渋滞シミュレーション(2)

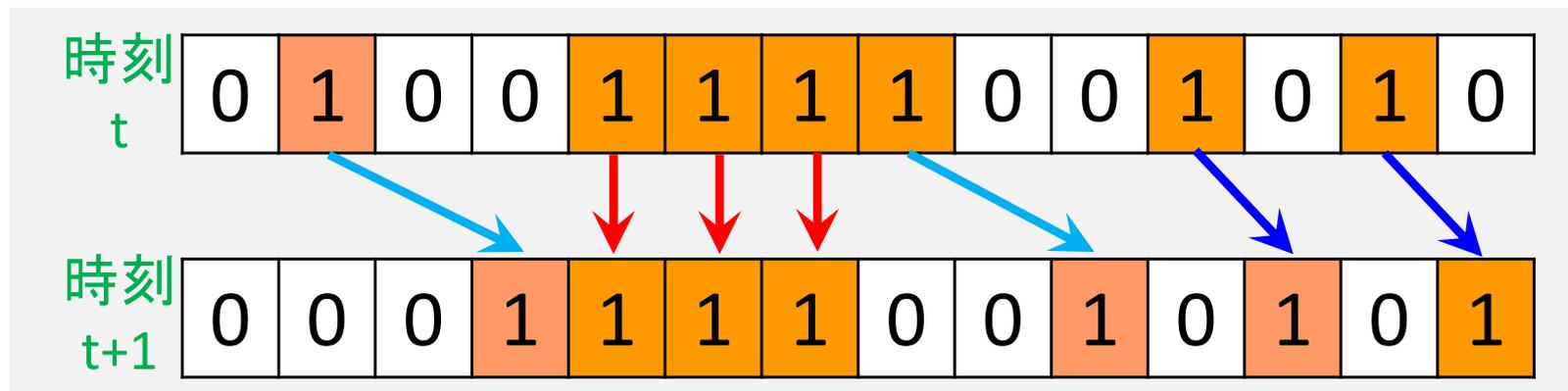
### – 車両の進行規則

1. 速度は道路状況により, 前方のセルが1つ空いている時は1つ進み, 2つ空いている時は加速して2つ進む
2. 前方が詰まっている(渋滞)時はその場に留まる



### – 1次元セルオートマトンで実現

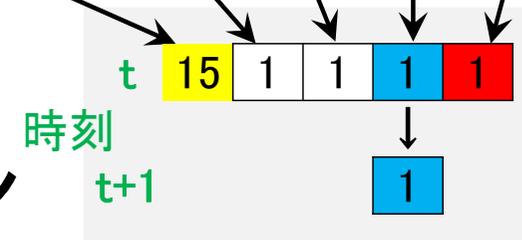
- 道路上に車両が居る状態を1, 居ない状態を0とする



# セルオートマトン

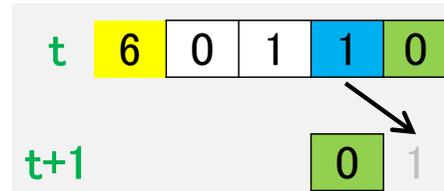
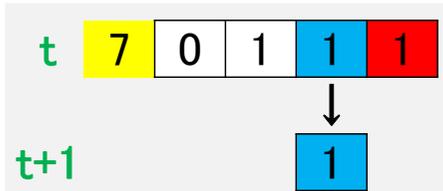
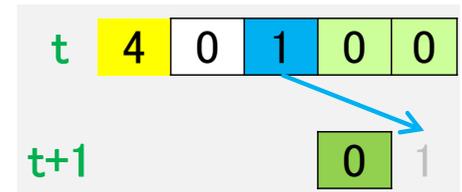
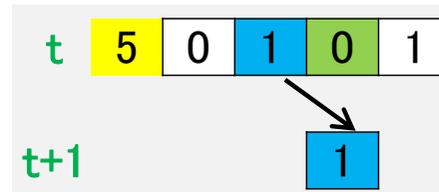
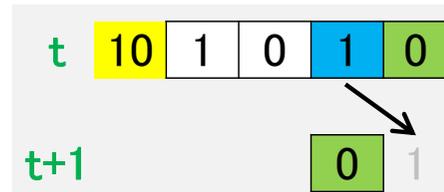
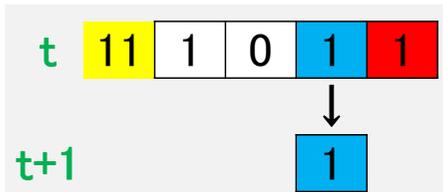
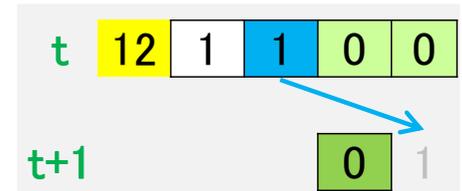
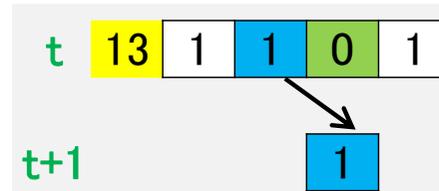
※4桁の2進数(1111)を10進数に変換した値

[左左近傍 | 左近傍 | 注目 | 右近傍]

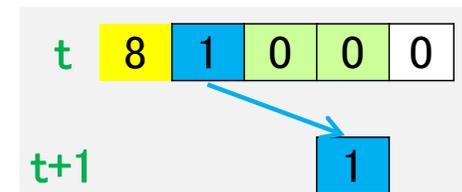
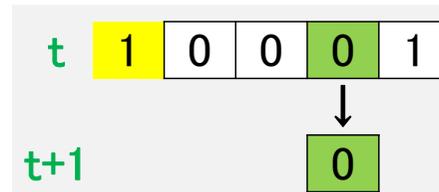
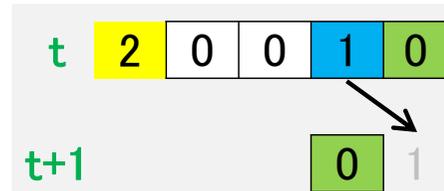
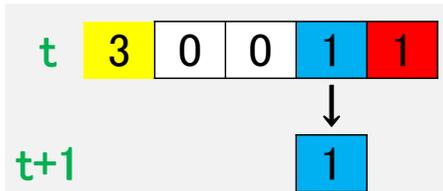
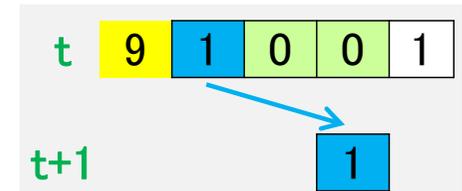


## 交通流・交通渋滞シミュレーション

「車両の進行規則」を1次元セルオートマトンの「ルール」へ

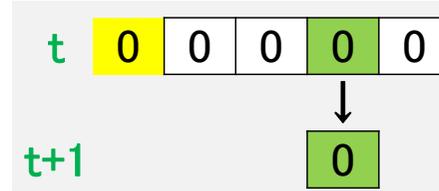


前方(注目セル)が空いているので、「左近傍」の車が前方(注目セル)に進む



前方(右近傍)が詰まっている(渋滞)ので「注目セル」の車は動かない(動けない)

前方(右近傍)が空いているので「注目セル」の車が前方に進み、道路が空く

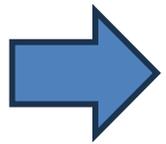


変化なし※空きのまま

前方が2マス空いているので加速して2マス前方に進む

# セルオートマトン

- 交通流・交通渋滞シミュレーション(2)
  - 「車両の進行規則」を1次元セルオートマトンの「ルール」へ



t	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
t+1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0

ルール43944 ( $= (1010\ 1011\ 1010\ 1000)_2$ )



# セルオートマトン

## • 交通流・交通渋滞シミュレーション(3)

### – 車両の進行規則

1. 速度は一定で, 前方が空いている時は前へ進む
2. 前方が詰まっている(渋滞)時はその場に留まるが, 右横と右前が両方空いていれば車線変更(右前へ移動)する

※車線変更は右車線へのみ



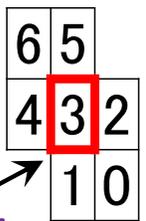
### – 2次元セルオートマトンで実現

- 道路上に車両が居る状態を 1, 居ない状態を 0とする
- 例) 2車線

時刻	0	1	0	0	1	1	1	0	0	1	0	1	0
t	0	1	0	0	1	0	0	1	1	1	0	0	1
						車線変更							
時刻	0	0	1	0	1	0	1	0	1	0	0	1	0
t+1	0	0	1	0	0	1	1	1	1	0	1	0	0

# セルオートマトン

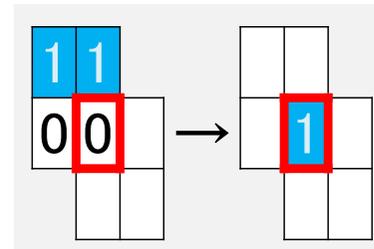
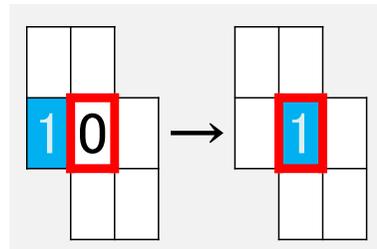
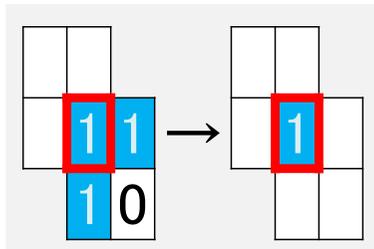
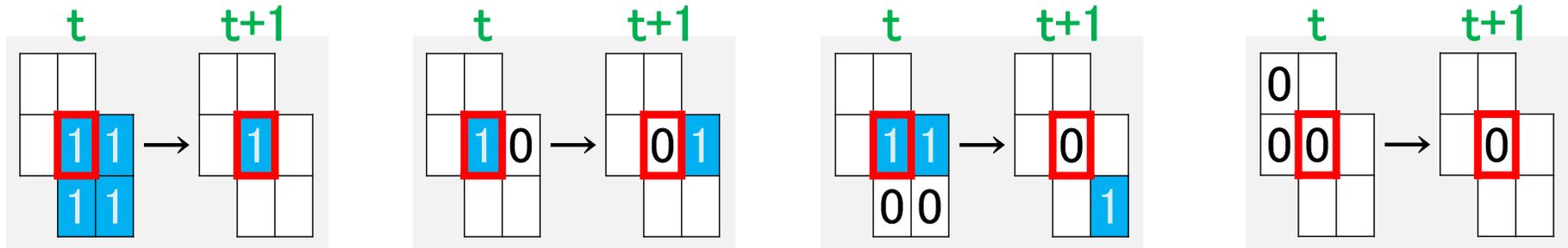
2次元セルオートマトンだが、近傍は「ノイマン近傍+斜め2カ所」とする  
 ※[6543210]順の[7桁の2進数]で表現



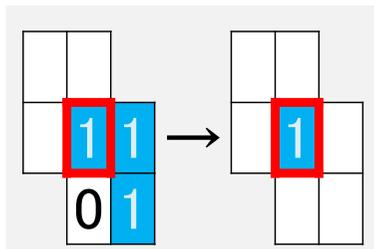
## 交通流・交通渋滞シミュレーション(3)

[注目セル]

「車両の進行規則」を2次元セルオートマトンの「ルール」へ



変化なし  
 ※空きのまま



前方が空いている  
 ので前方に進む

前方が詰まっている  
 (渋滞)が、右車線が  
 空いているので  
右前に車線変更して  
 進む

前方(右近傍)も右車線  
 も詰まっている(渋滞)  
 ので「注目セル」の車は  
動かない(動けない)

2<sup>7</sup>=128パターンを  
 8つに分類

空欄の状態は [0 / 1]  
 どちらの場合も含む

# セルオートマトン

前ページの渋滞で動けないに該当

例)  $110 (= (1101101)_2)$

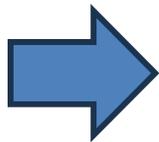
	t			t+1	
	1	1			
	0	1	1		
		0	1		

→

			1		

- 交通流・交通渋滞シミュレーション

「車両の進行規則」を2次元セルオートマトンの「ルール」へ



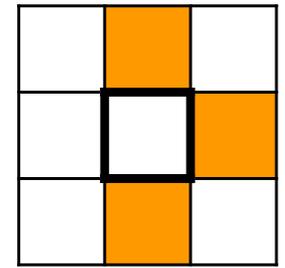
t	t+1	t	t+1	t	t+1	t	t+1	t	t+1	t	t+1	t	t+1	t	t+1	t	t+1	t	t+1
127	1	111	1	95	1	79	1	63	1	47	1	31	1	15	1				
126	1	110	1	94	1	78	1	62	1	46	1	30	1	14	1				
125	1	109	1	93	1	77	1	61	1	45	1	29	1	13	1				
124	0	108	0	92	0	76	0	60	0	44	0	28	0	12	0				
123	0	107	0	91	0	75	0	59	0	43	0	27	0	11	0				
122	0	106	0	90	0	74	0	58	0	42	0	26	0	10	0				
121	0	105	0	89	0	73	0	57	0	41	0	25	0	9	0				
120	0	104	0	88	0	72	0	56	0	40	0	24	0	8	0				
119	1	103	1	87	1	71	0	55	1	39	0	23	1	7	0				
118	1	102	1	86	1	70	0	54	1	38	0	22	1	6	0				
117	1	101	1	85	1	69	0	53	1	37	0	21	1	5	0				
116	1	100	1	84	1	68	0	52	1	36	0	20	1	4	0				
115	1	99	1	83	1	67	0	51	1	35	0	19	1	3	0				
114	1	98	1	82	1	66	0	50	1	34	0	18	1	2	0				
113	1	97	1	81	1	65	0	49	1	33	0	17	1	1	0				
112	1	96	1	80	1	64	0	48	1	32	0	16	1	0	0				

$2^7 =$   
128パターン





# 参考文献



- [1] 北栄輔・脇田佑希子「Excelで学ぶセルオートマトン」  
オーム社(2011)
- [2] 岡 瑞起他「作って動かすALife」オライリージャパン(2018)
- [3] 岡 瑞起「ALIFE | 人工生命」BNN(2022)