

知の探究

3.セルオートマトン

堀田 敬介

2025年10月20日(月)

セルオートマトン

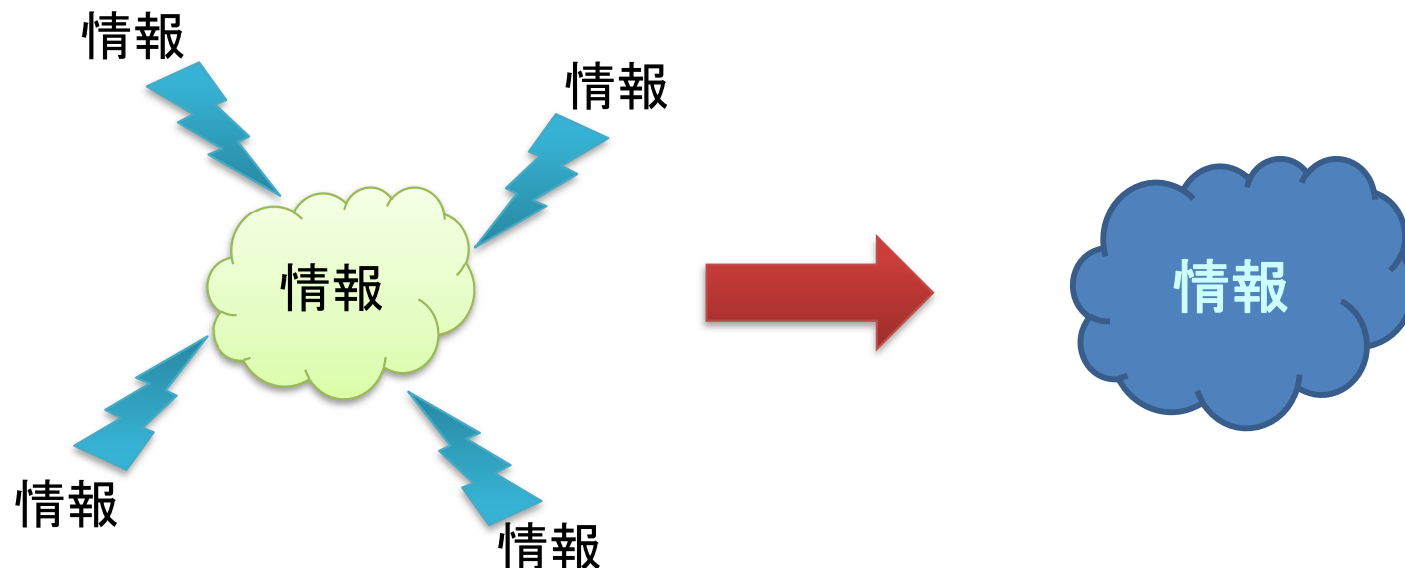
- Cellular Automata

- 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入力するシステム」 [1]

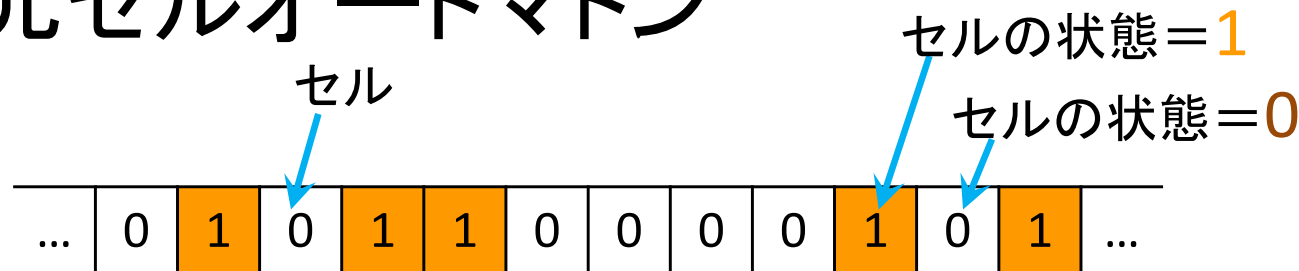
Cellular 細胞の
Automata 自動人形(からくり人形)
※Automaton(単数形)

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



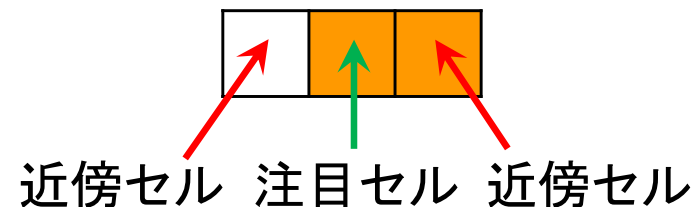
セルオートマトン

- 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

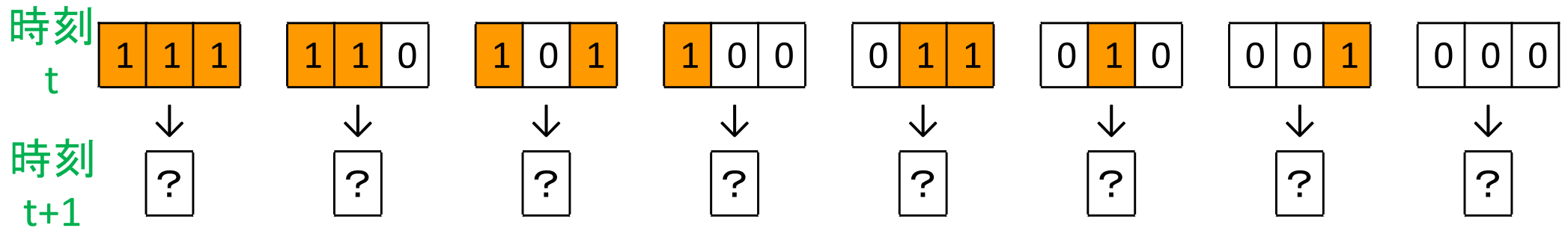
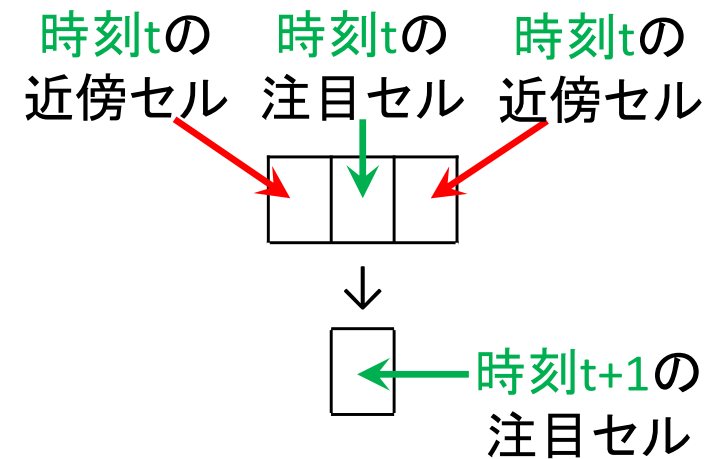


- 両端のセルの近傍の設定方法は2種類
 - 周期**境界条件 ...「左端セルの左側近傍は, 一番右端のセル」「右端セルの右側近傍は, 一番左端のセル」と考える(※ドーナツのイメージ)
 - 開放**境界条件 ...「左端セルの左側近傍は, なし」「右端セルの右側近傍は, なし」と考える

セルオートマトン

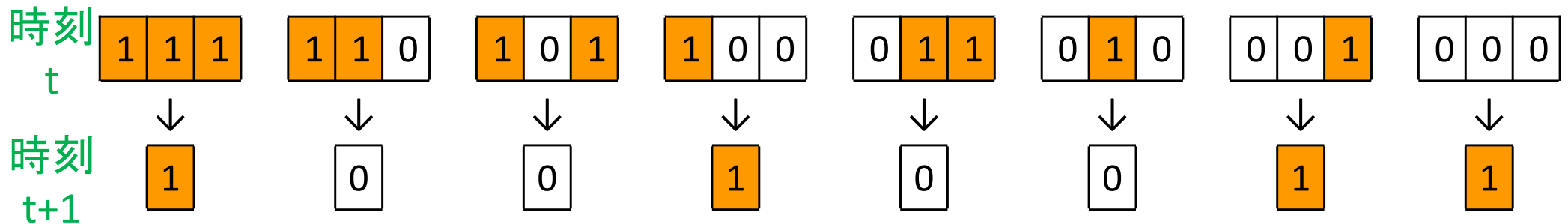
- 1次元セルオートマトン

- 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

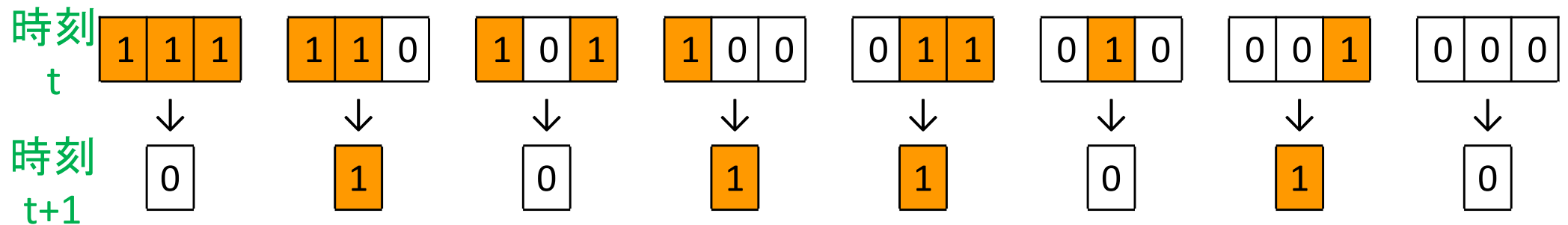
時刻 $t+1$ の考えられるパターンは全部で256個ある

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

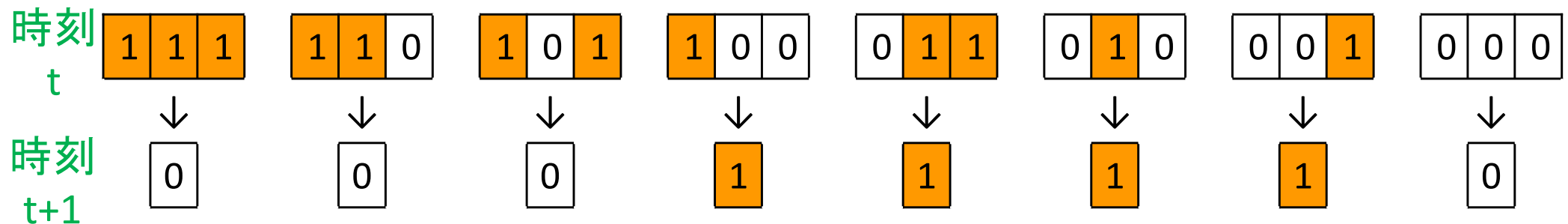
$(00000000)_2$ = 「ルール0」 $\sim$ $(11111111)_2$ = 「ルール255」

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」

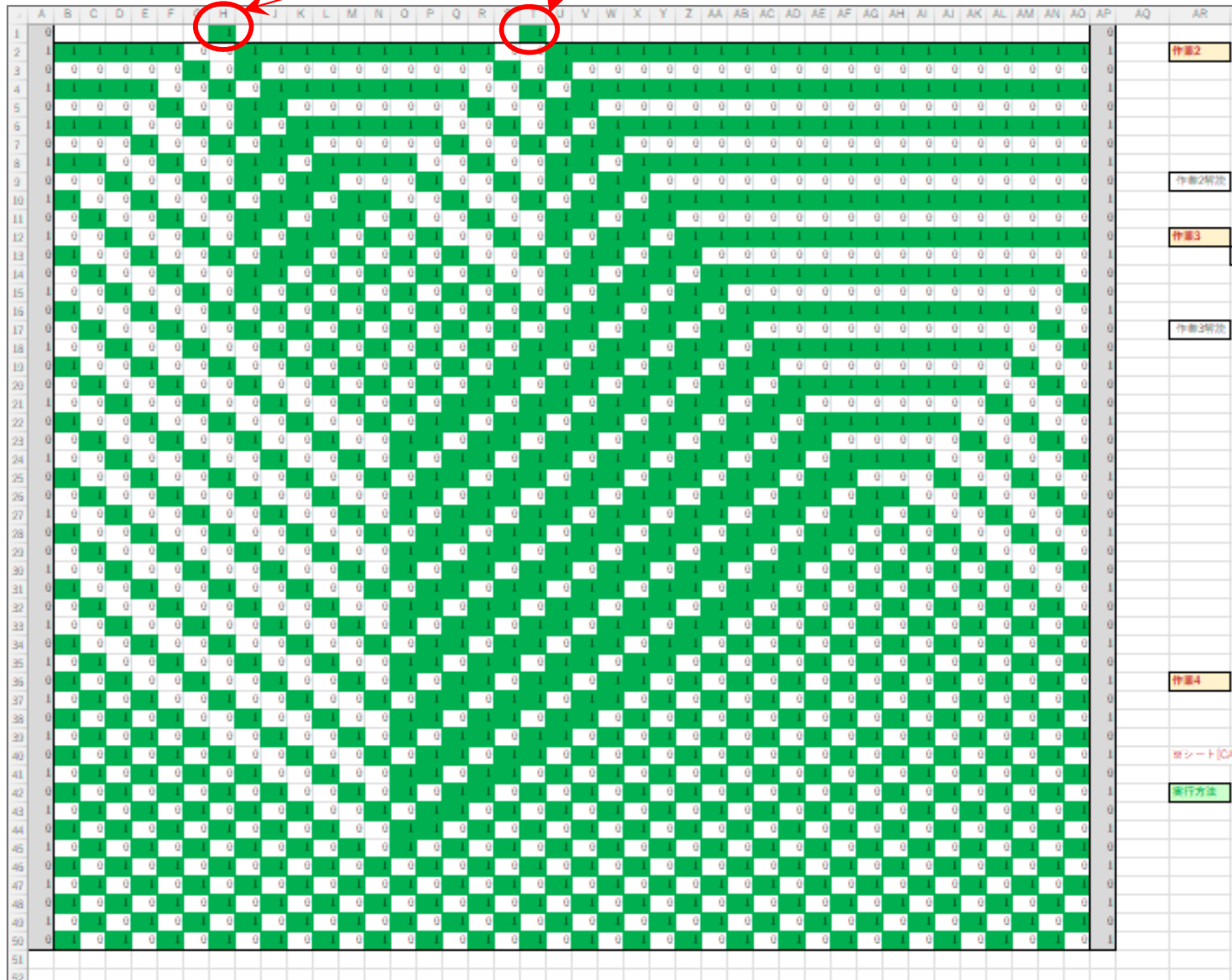


➡ $(00011110)_2 = (30)_{10}$ より, これは「ルール30」

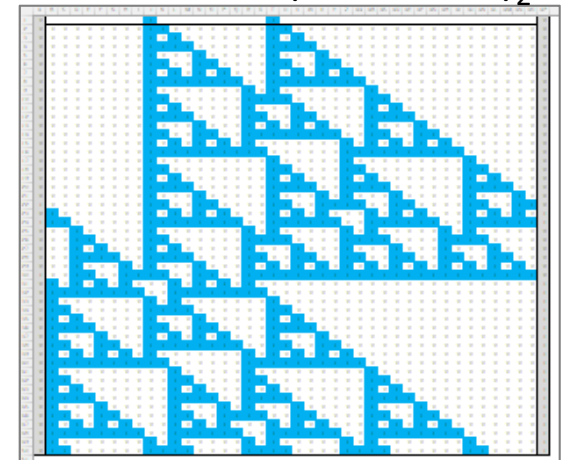
セルオートマトン

• 実行例)

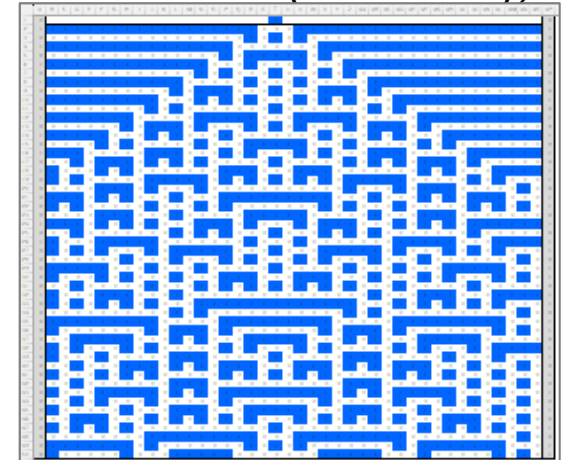
- ✓ 初期状態(時刻 $t=0$ の状態)は, セル[H1]と[T1]の2箇所が1でそれ以外は0
- ✓ 適用規則はルール57(= $(0011\ 1001)_2$)



- ✓ 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

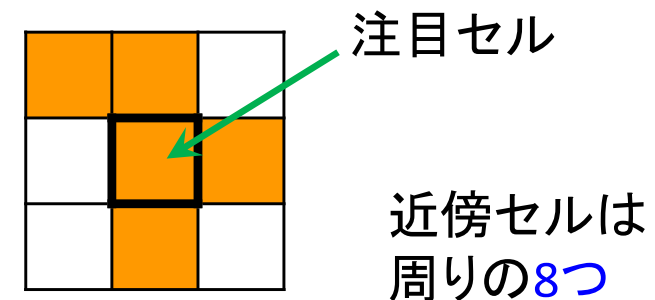
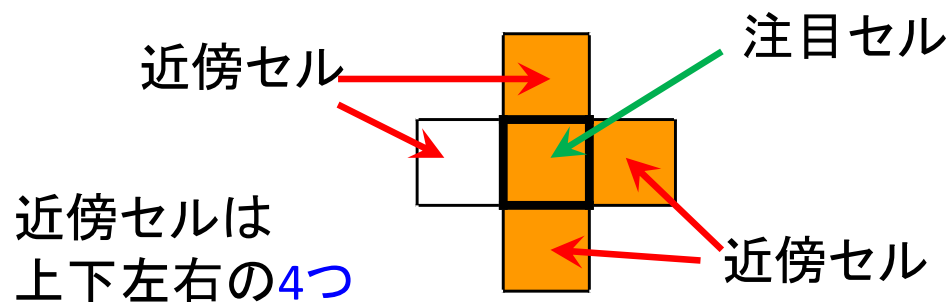
セルの状態=1
セルの状態=0

※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



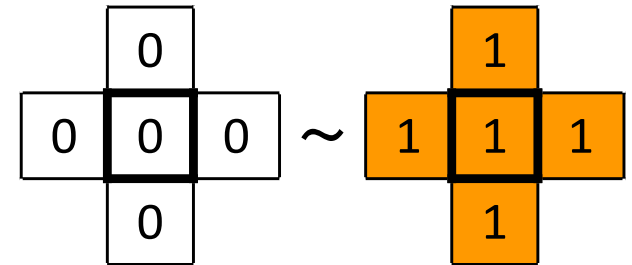
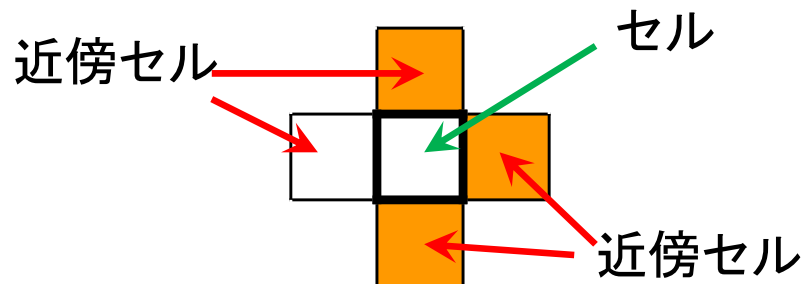
セルオートマトン

- 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

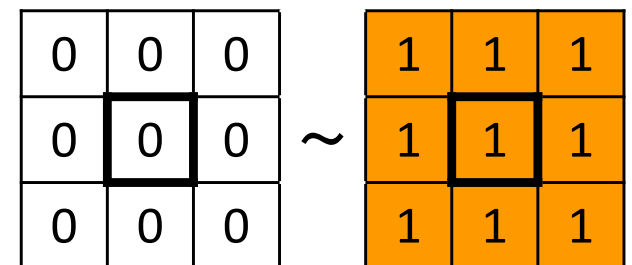
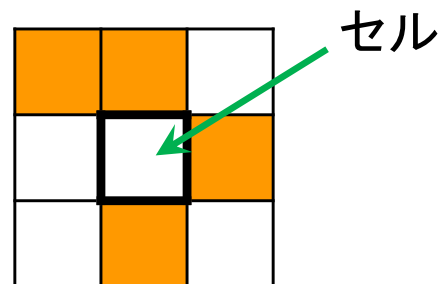
- ノイマン近傍の場合、全部で32パターン 何故か？

$$(32 = 2^5)$$



- ムーア近傍の場合、全部で512パターン 何故か？

$$(512 = 2^9)$$



セルオートマトン

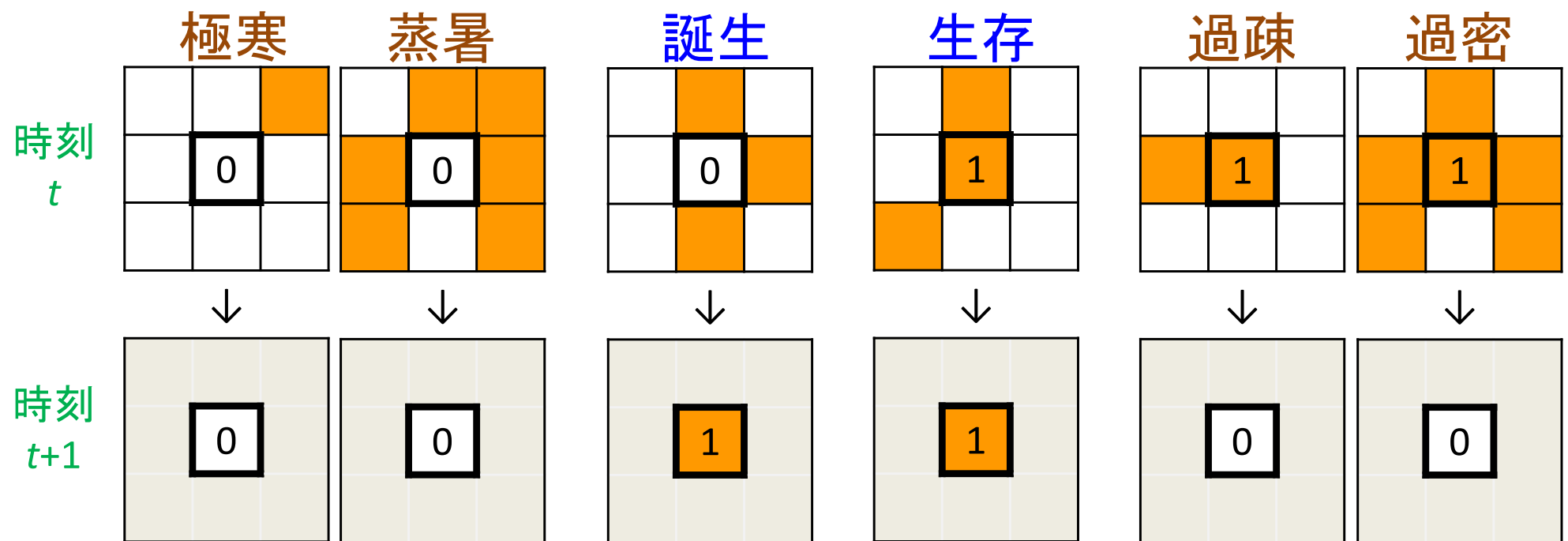
- 2次元セルオートマトンとライフゲーム

- 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4つに分類)
- 状態の更新ルールは次の4つ
 1. 極寒/蒸暑...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2-3つが生(1) → 1
 4. 過疎/過密...自セル=生(1) & 近傍セルの0-1,4-8つが生(1) → 0

セルオートマトン

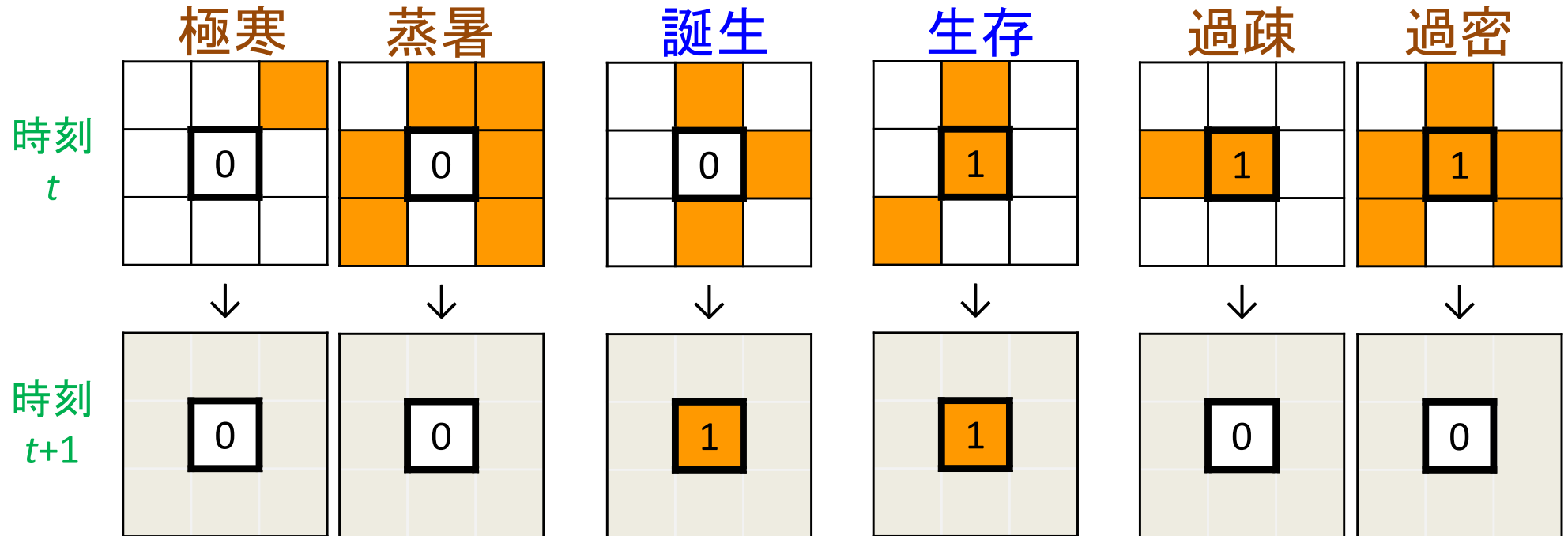
- ライフゲームのルール

1. 極寒/蒸暑...自セル=死(0)&近傍セルの0-2,4-8つが生(1) $\rightarrow 0$
2. 誕生 ...自セル=死(0) & 近傍セルの3つが生(1) $\rightarrow 1$
3. 生存 ...自セル=生(1) & 近傍セルの2-3つが生(1) $\rightarrow 1$
4. 過疎/過密...自セル=生(1)&近傍セルの0-1,4-8つが生(1) $\rightarrow 0$



セルオートマトン

- ライフゲームのルール



- ゲームのルールを状態遷移表で表現 (Excel用)

		S = 時刻 t の近傍セルの状態の和 (= 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

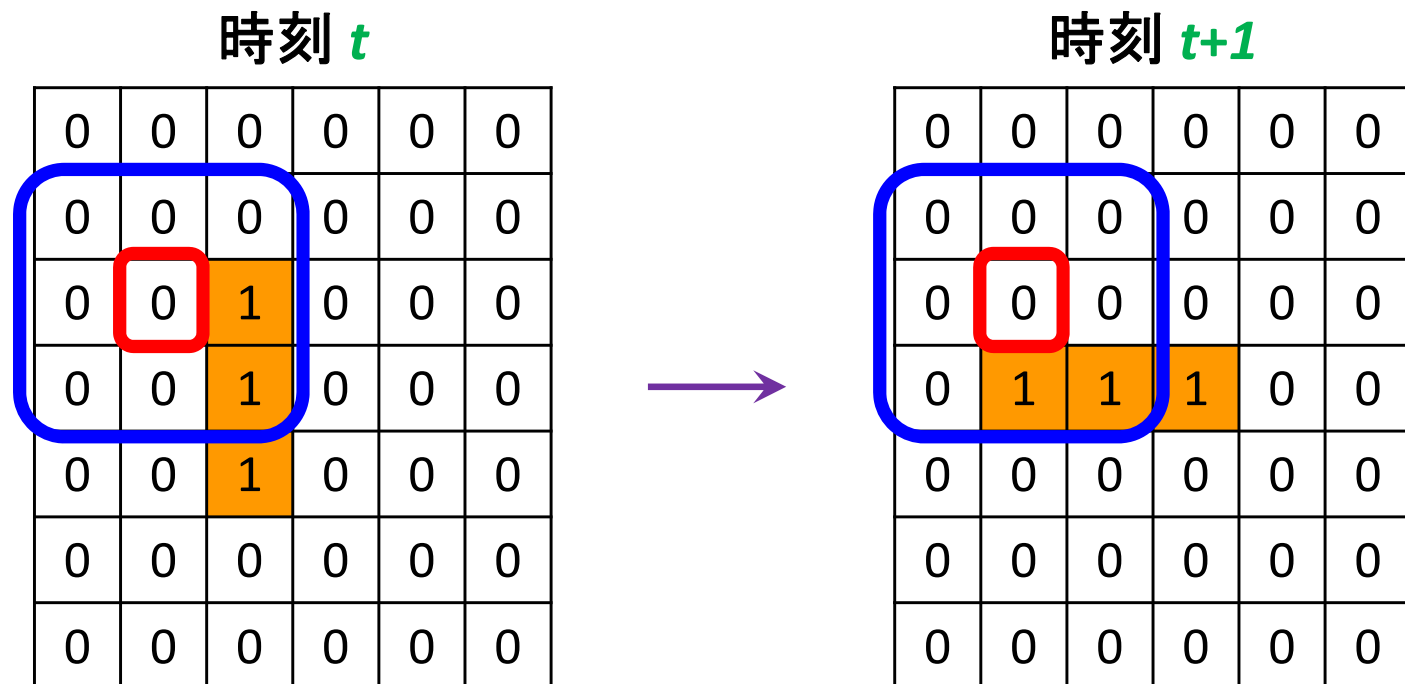
セルオートマトン

- ゲームのルールを状態遷移表で表現(Excel用)

		S = 時刻 t の近傍セルの状態の和(= 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

- 実行例)



- Life Game :実行例

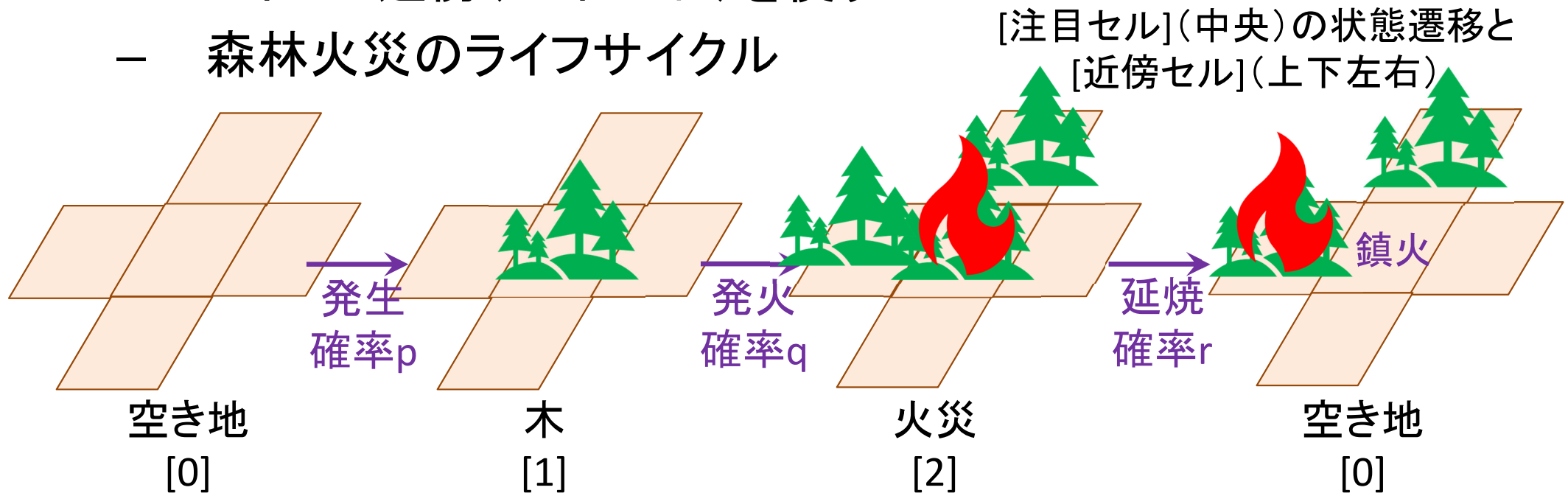
- ✓ 上部に[グライダー銃], 左中央に[ペンタデカスロン], 左下に[パルサー] を配置して 実行

[illegible]

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

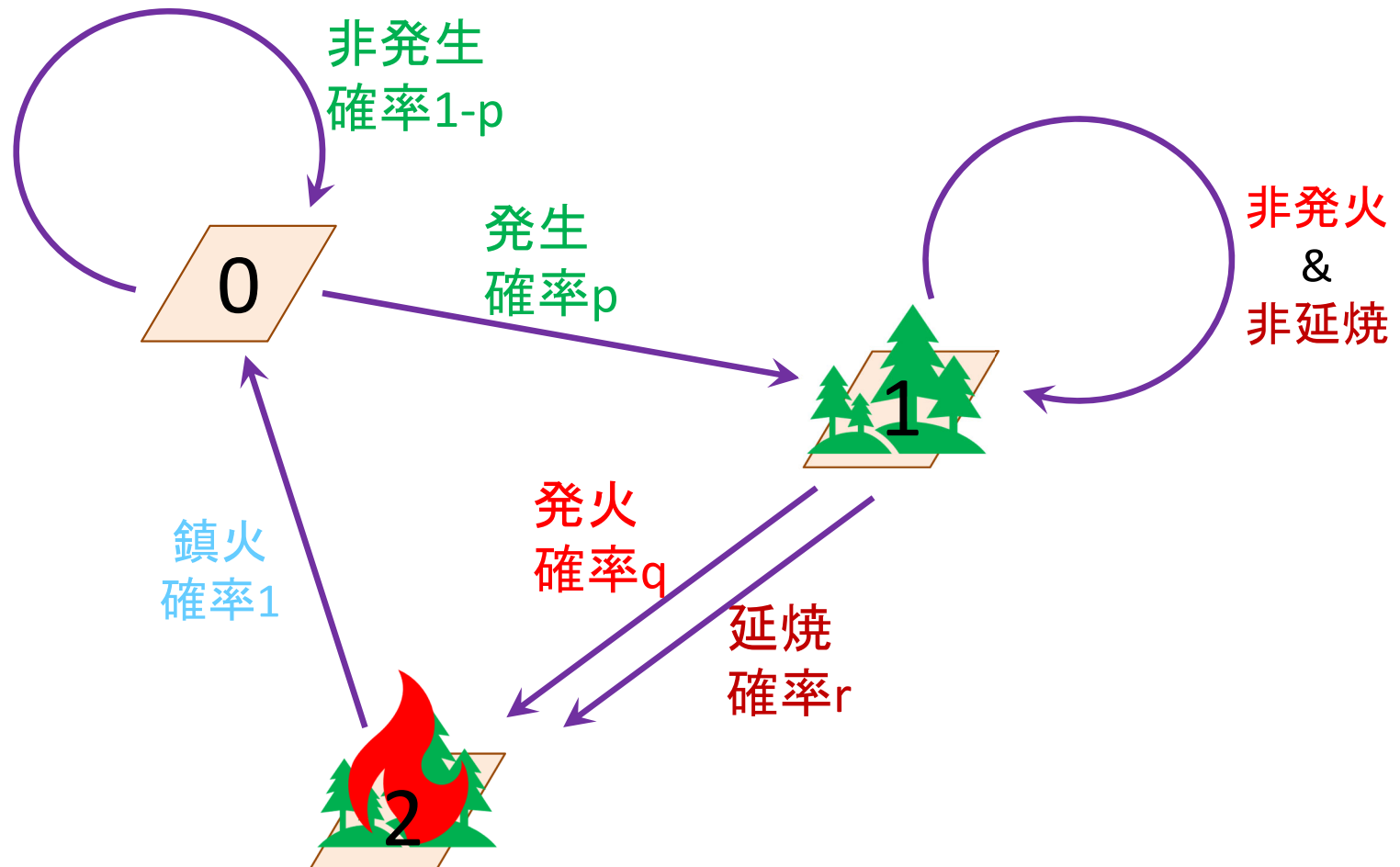


- 空き地[0]の状態では, 一定確率 p で木が発生
- 木[1]の状態かつ近傍全部非火災では, 一定確率 q で発火
- 火災[2]状態の近傍の木[1]は, 一定確率 r で延焼
- 火災[2]状態の木は, 時刻1期後に燃え尽き(鎮火)空き地[0]に

セルオートマトン

- 森林火災シミュレーション

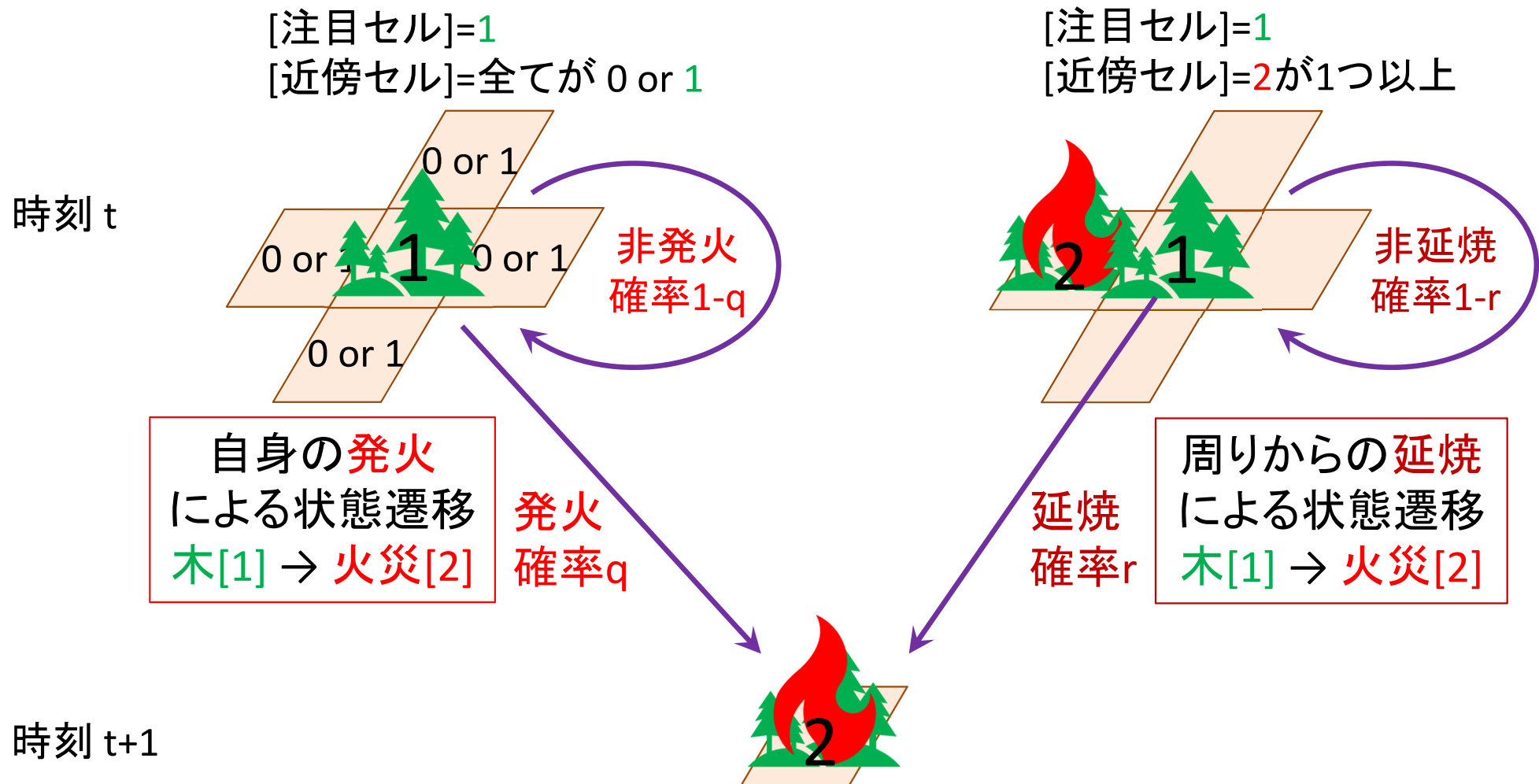
- 状態遷移図state transition diagram, life cycle



セルオートマトン

- 森林火災シミュレーション

- 状態遷移図の(木[1] → 火災[2])部分の2パターン詳細



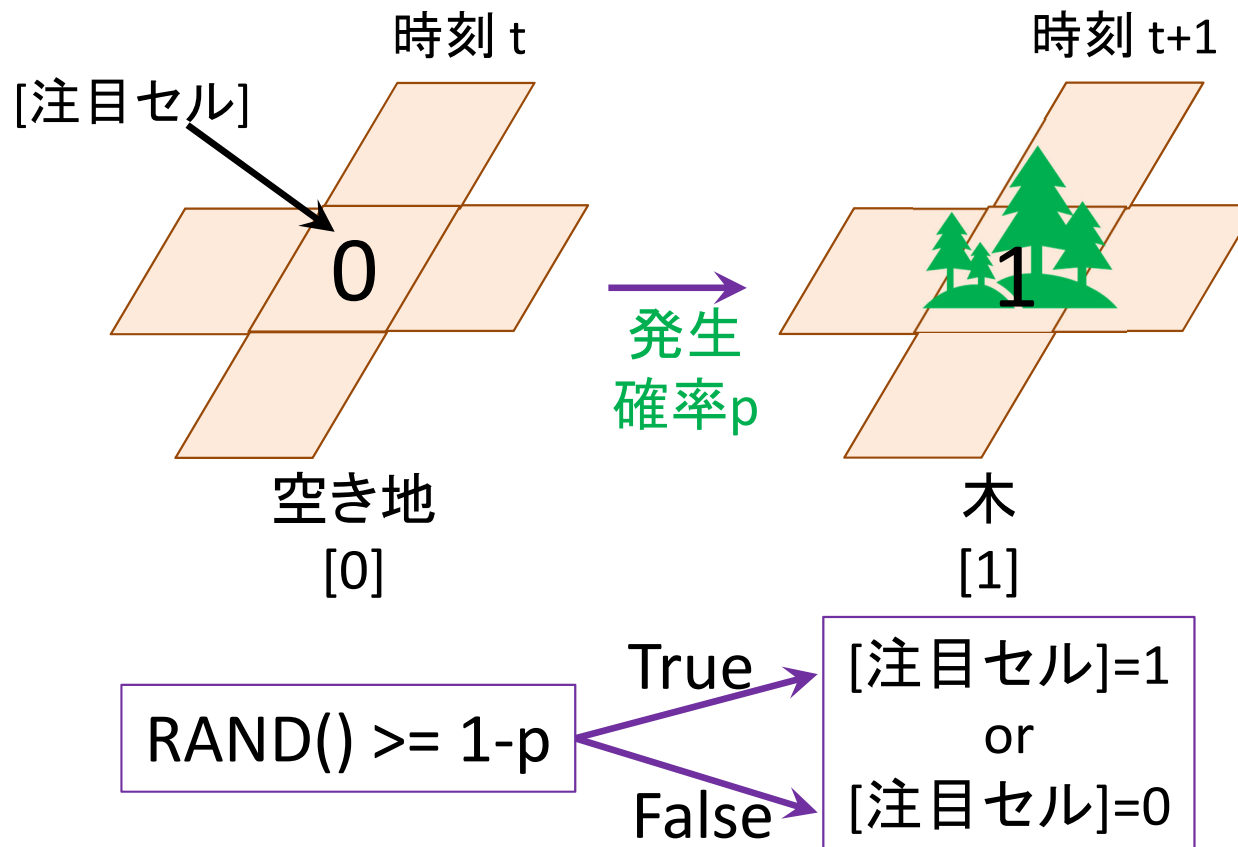
セルオートマトン

RAND() は, $[0,1)$ -一様擬似乱数

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

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

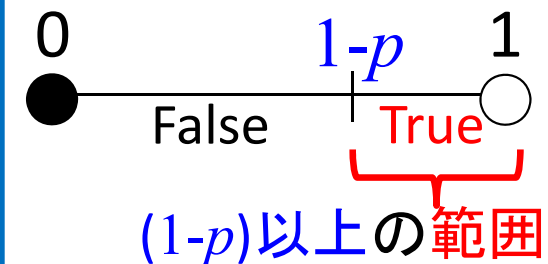


この仕組みを実現する式

時刻 $t+1$ の [注目セル] = $IF(RAND() \geq 1-p, 1, 0)$

$RAND() < p$ と同じ
(後の都合で以下を使用)

$RAND() \geq 1-p$



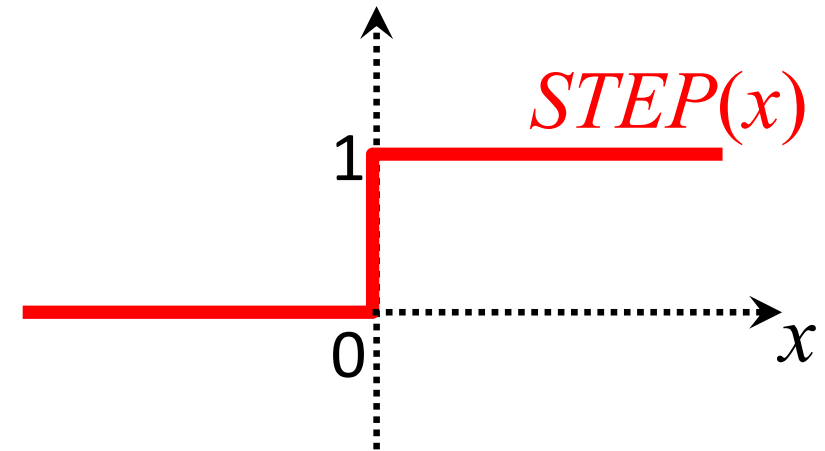
例えば, $p=0.05$ なら,
 $1-p=0.95$ より, 生成した
乱数値がこの範囲に
入る確率が 0.05 (5%)

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

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

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

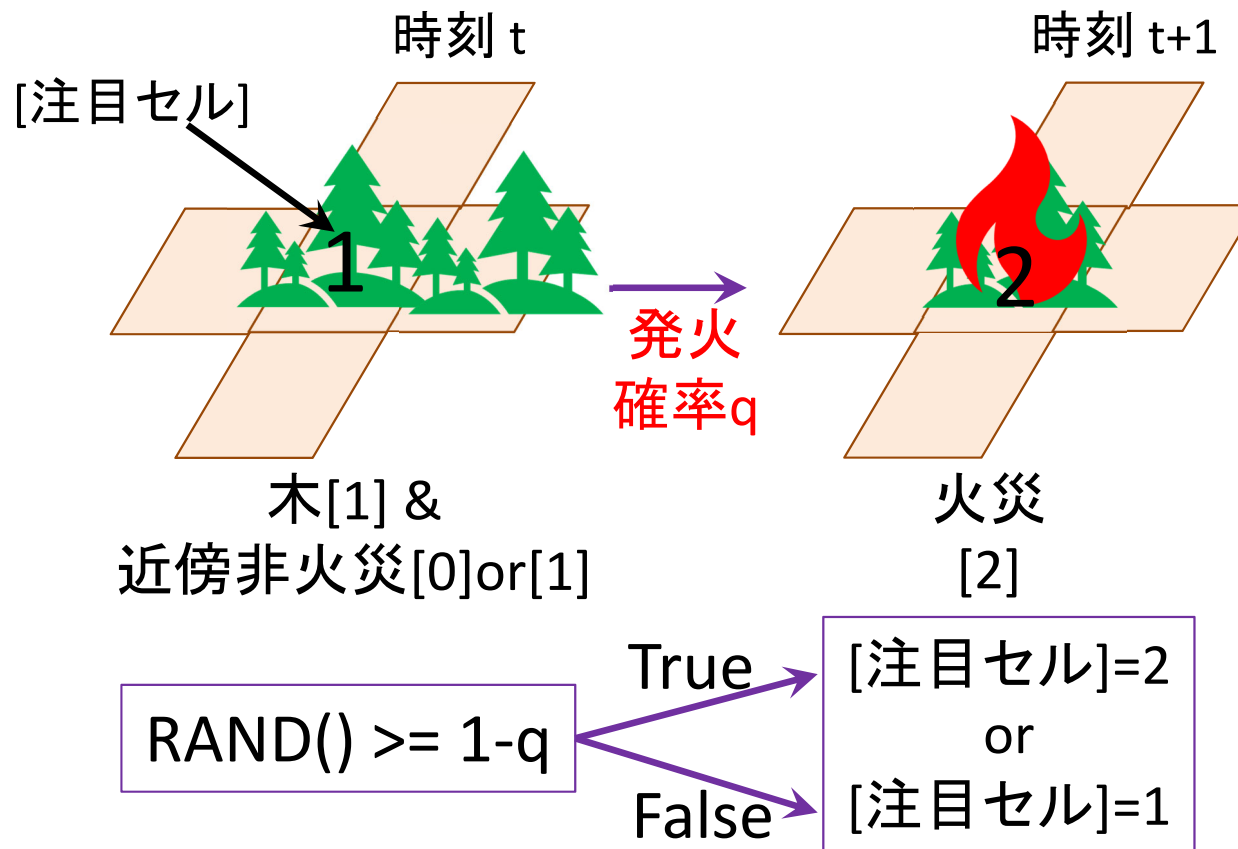
$$= \text{IF}(\text{RAND()} - 1 + p \geq 0, 1, 0)$$

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

※素直に $\text{IF}(\text{RAND()} < p, 1, 0)$ を使わないのはSTEP関数を使いたいため
(IF文の条件式を非負にしたいため)

セルオートマトン

- 森林火災シミュレーション: 自身の(自然)発火
 - [注目セル]=1(木)&近傍全て非火災では, 一定確率 q で発火

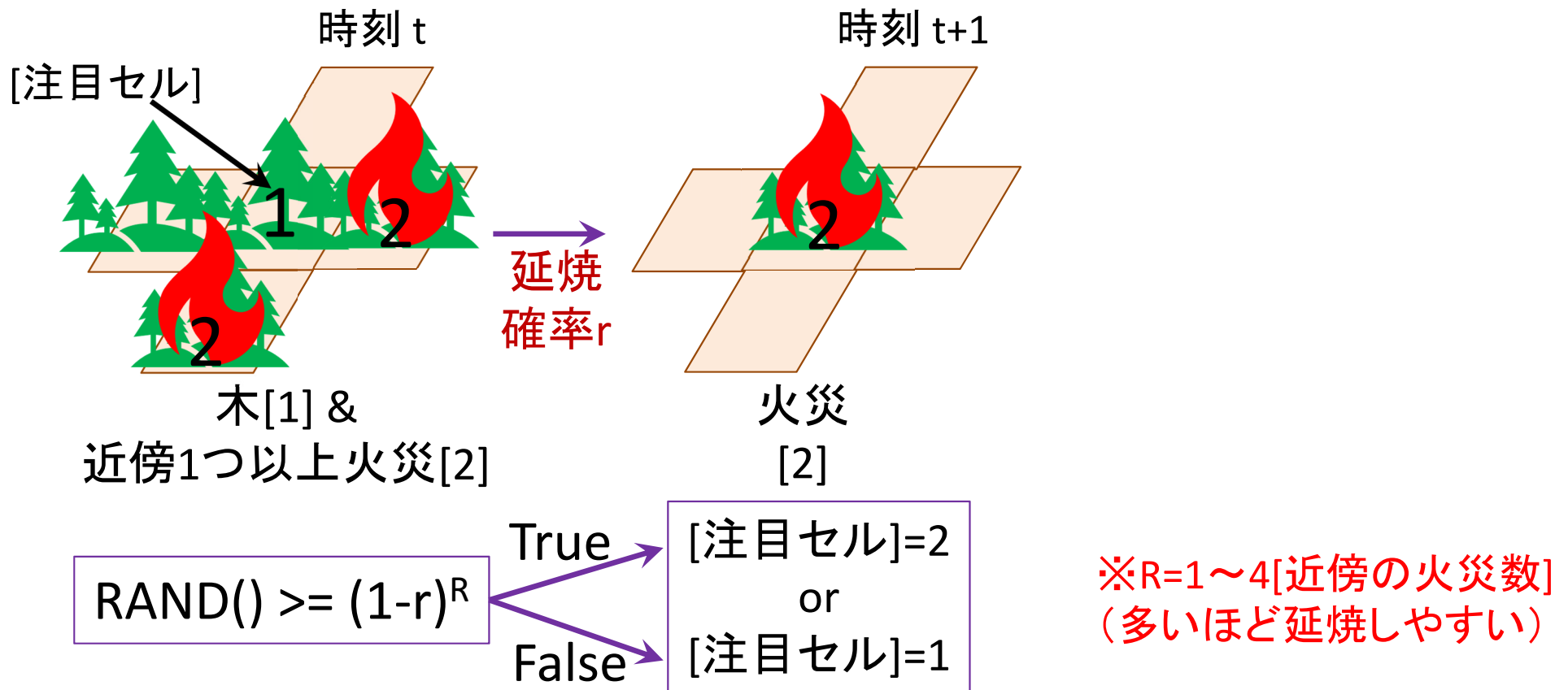


この仕組みを実現する式

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

セルオートマトン

- 森林火災シミュレーション: 周りからの延焼
 - [注目セル]=1(木)&近傍1つ以上火災では, 一定確率 r で延焼

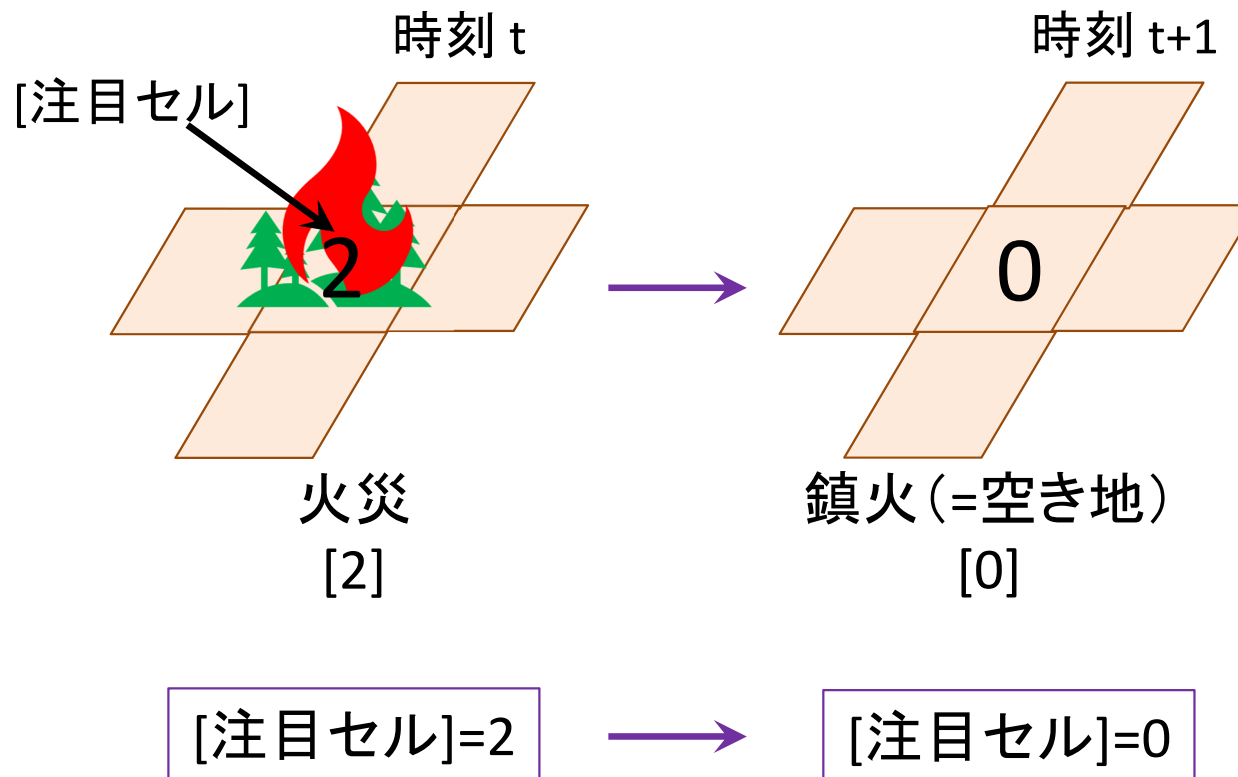


この仕組みを実現する式

時刻 $t+1$ の[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



セルオートマトン

● 森林火災

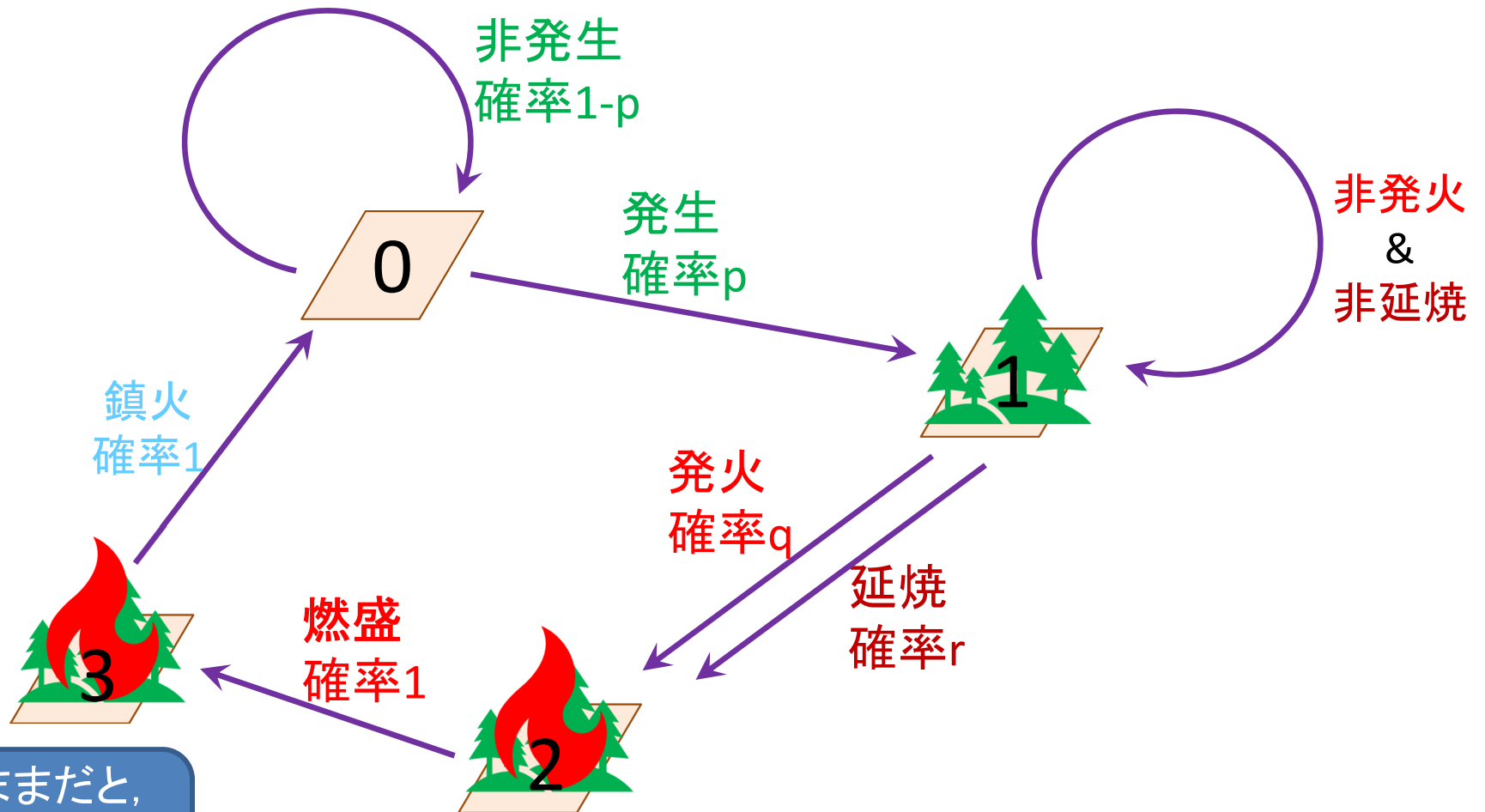
ー 実行例

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

	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	AI	AJ	
1	1	1	0	0	0	1	2	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1				
2	0	1	1	0	1	0	0	2	0	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0	0	1	0	1		Update	作業3	寺	
3	1	1	1	1	0	0	0	0	2	1	1	1	1	1	1	1	0	1	0	1	1	1	2	1	1	1	1	0	1	0	1	1					
4	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1					
5	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0				
6	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	前回同様	作業4	同	
7	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1				
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1				
9	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1				
11	0	1	1	1	1	1	1	0	1	1	1	1	1	1	2	0	1	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	1				
12	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1				
13	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	1	0	1	1				
14	2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0	0	1	1	0	1	1	1	1	1	2	1					
15	0	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1				
16	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1				
17	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1				
18	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		作業4解説	前回		
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1					
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	前回同様	作業5	Exc	
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
23	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
24	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
25	0	1	1	1	1	1	1	1	1	1	1	2	1	1	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1					
26	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
27	1	1	1	1	2	1	1	1	1	0	0	0	2	1	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1				
28	1	1	1	1	2	0	2	1	1	1	0	1	2	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1			※前		
29	1	1	1	0	0	2	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1				
30	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	前回同様	作業6	Exc	
31	1	1	0	0	0	1	2	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	1						
32	0	1	1	0	1	0	0	2	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0	0	1	0	1						

セルオートマトン

- 演習) 森林火災シミュレーション
 - 状態遷移図を以下に変更してみよう



状態が2のままだと、
時刻を記憶しなければ
ならないので面倒

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



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

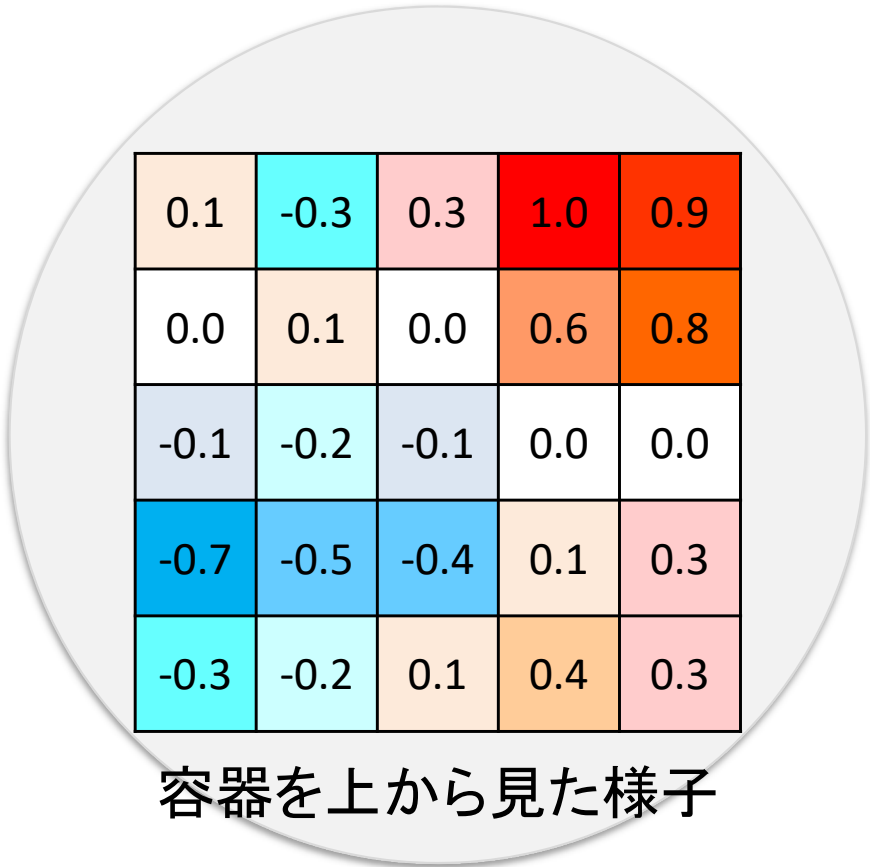
容器を上から見た様子

例) 水と油の場合の状態量の意味

状態量	
-1.0	水だけの液体
-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	油だけの液体

セルオートマトン

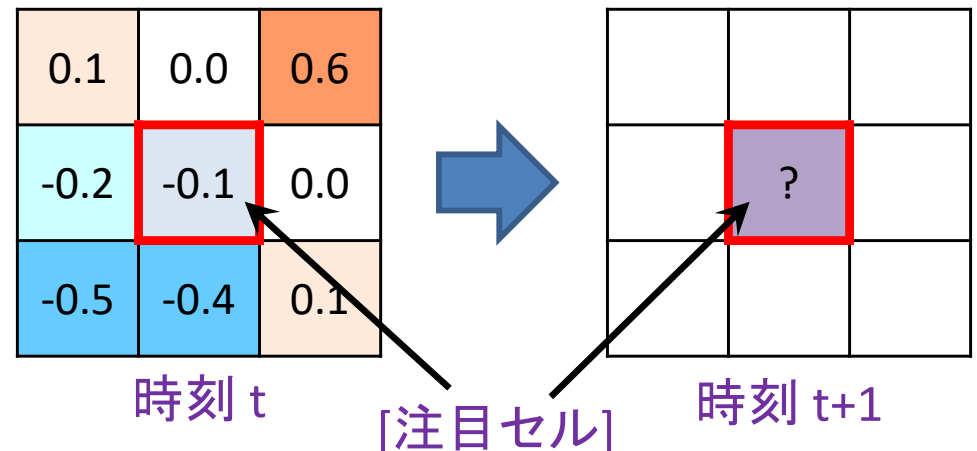
- 2次元セルオートマトンと液体分離シミュ
- 混ざり合わない2つの液体(水と油など)を1つの容器内で強引に混ぜ合わせた後, 時間と共に分離する様子をシミュレーションする



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 $[-1.0 \sim 1.0]$ の範囲の数値に変換する

※ μ と δ 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,

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

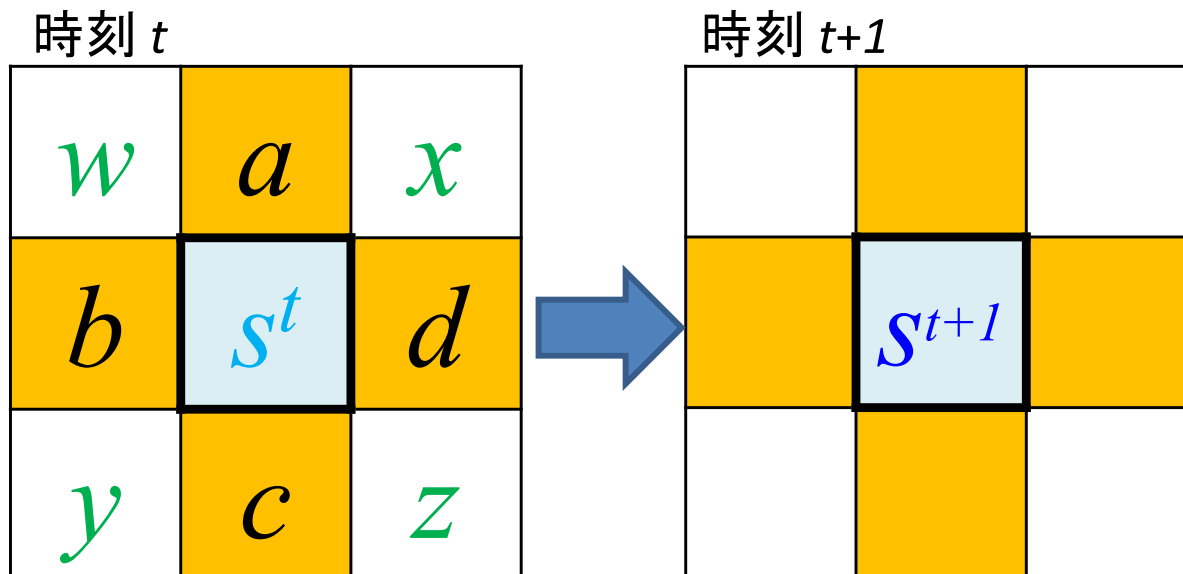
$a, b, c, d, w, x, y, z, s^t \in [-1.0, 1.0]$ より,

$T \in [-2.0, 2.0]$ 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



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 = μ	
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	-0.9	-0.8	-0.6	0.1	0.7	0.9	1.0	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	-0.6		
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	0.6	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.2	-0.6		
6	-0.2	-0.8	-0.9	-1.0	-1.0	-1.0	-0.9	-0.8	-0.6	-0.3	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	-0.5	-0.6		
7	-0.5	-0.8	-0.9	-1.0	-1.0	-0.9	-0.8	-0.5	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	-0.6	
8	-0.6	-0.9	-1.0	-1.0	-0.9	-0.8	-0.5	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	1.0	0.9	0.8	0.2	-0.6	-0.9	
9	-0.8	-0.9	-1.0	-1.0	-0.9	-0.7	0.0	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	1.0	0.9	0.6	-0.5	-0.8	-0.9	
10	-0.8	-0.9	-1.0	-0.9	-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	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	-0.5	-0.8	-0.9
11	-0.9	-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	1.0	0.9	0.8	0.3	-0.6	-0.9	-1.0
12	-0.9	-1.0	-1.0	-0.9	-0.8	-0.3	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	1.0	0.9	0.7	-0.5	-0.7	-0.9	-1.0
13	-0.9	-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	1.0	0.9	0.6	-0.5	-0.8	-0.9	-1.0
14	-0.9	-1.0	-1.0	-1.0	-0.9	-0.6	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	-0.8	-0.9	-1.0
15	-0.9	-1.0	-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	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	-0.4	-0.8	-0.9	-1.0
16	-0.9	-1.0	-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	0.7	0.8	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	-0.5	-0.7	-0.9	-1.0
17	-0.8	-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	1.0	0.9	0.8	0.2	-0.6	-0.8	-0.9	
18	-0.6	-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	0.7	0.9	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.6	0.0	-0.6	-0.8	-0.9
19	0.2	-0.4	-0.7	-0.8	-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	-0.5	0.1	0.7	0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.6	0.2	-0.4	
20	0.7	0.5	0.0	-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	1.0	0.9	0.9	0.7	0.5	
21	0.9	0.8	0.6	-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	-1.0	-0.9	-0.7	-0.5	0.7	0.9	1.0	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	-0.9	-0.9	-0.9	-1.0	-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	-0.9	-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	-0.5	-0.8	-0.9	-1.0	-1.0	-1.0	-0.9	-0.9	-0.7	-0.5	-0.5	-0.9	-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	-0.9	-0.9	-0.8	-0.8	-0.9	-1.0	-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	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	1.0	0.9	0.6	-0.5		
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	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	-0.9	-0.9	-0.5	0.4	0.8	0.9	1.0	1.0	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	-0.9	-0.8	-0.4	0.6	0.9	1.0	1.0	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	-0.9	-0.8	-0.6	0.1	0.7	0.9	1.0	1.0	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)

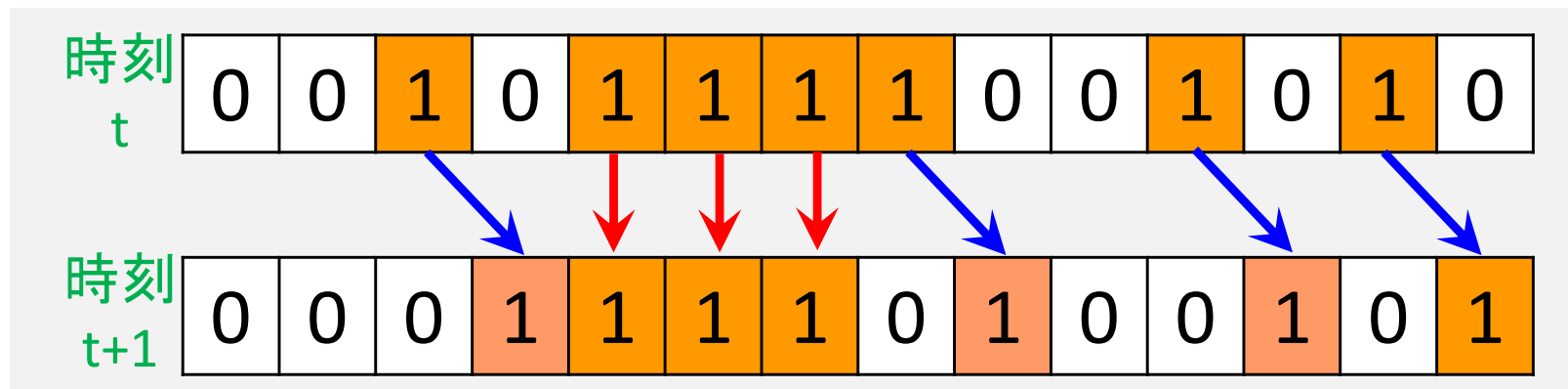
– 車両の進行規則

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

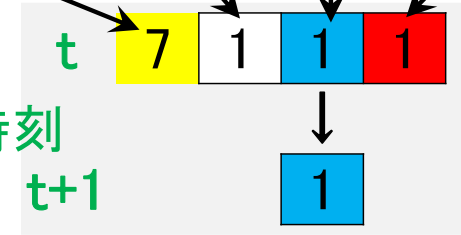


セルオートマトン

※3桁の2進数(111)を
10進数に変換した値

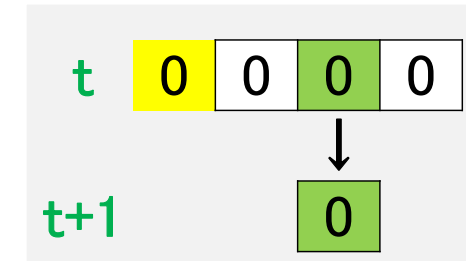
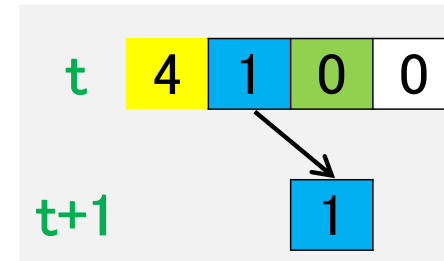
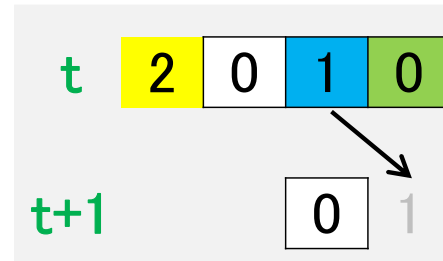
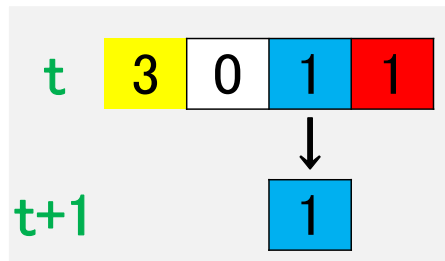
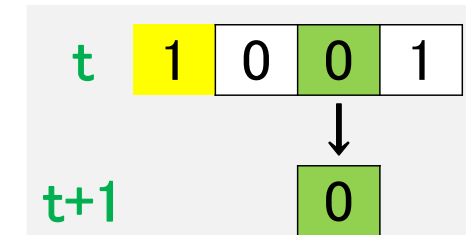
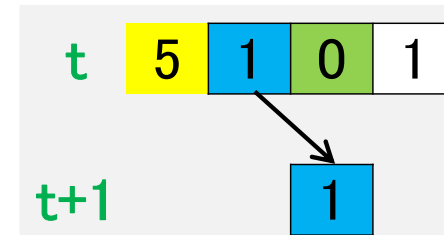
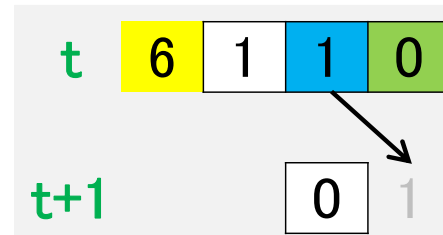
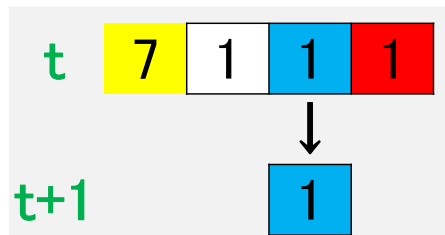
[左近傍 | 注目 | 右近傍]

時刻
t
t+1



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

t	7	6	5	4	3	2	1	0
t+1	1	0	1	1	1	0	0	0

ルール184

(= (1011 1000)₂)

交通シミュレーション
の基本モデル

- 交通流・交通渋滞シミュレーション(1)

- ※初期配置をセル[B11:AY13] の状態からスタート

下1車線は、車両占有率が高く(62%),
渋滞が解消されない

[illegible]

セルオートマトン

• 交通流・交通渋滞シミュレーション(2)

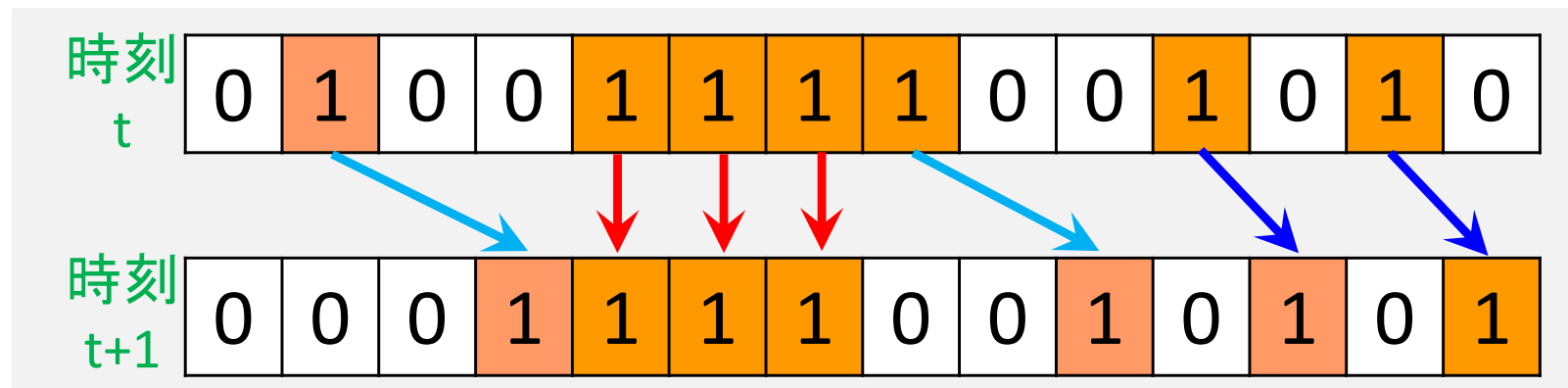
– 車両の進行規則

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

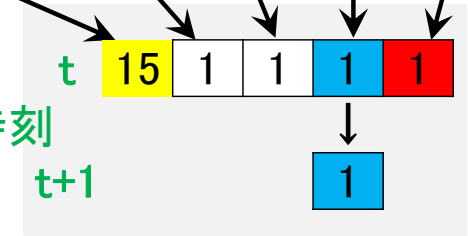


セルオートマトン

※4桁の2進数(1111)を
10進数に変換した値

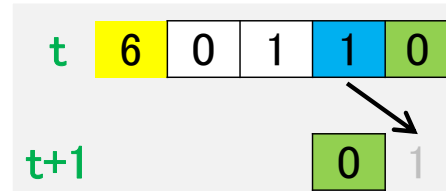
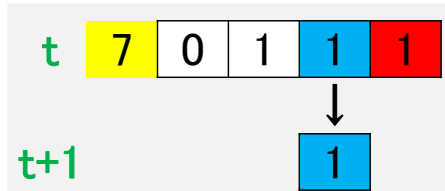
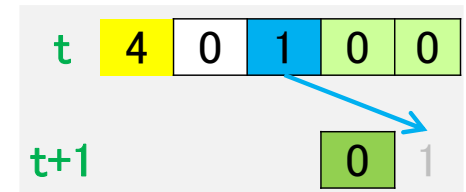
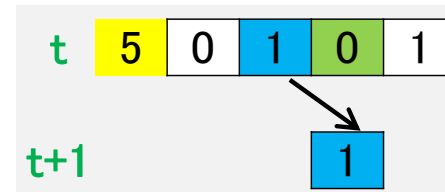
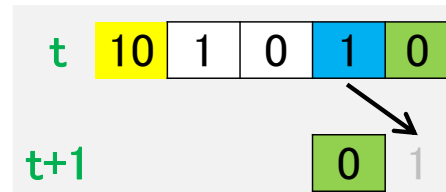
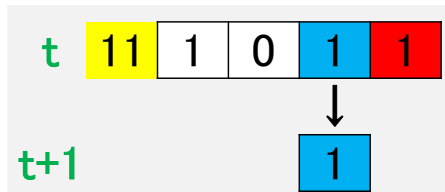
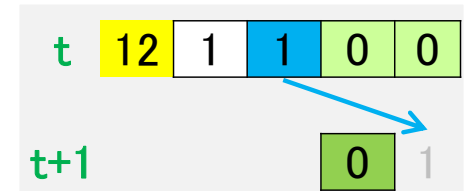
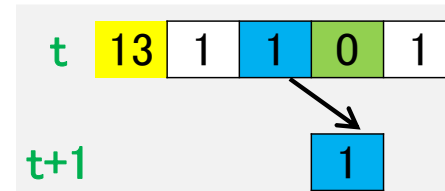
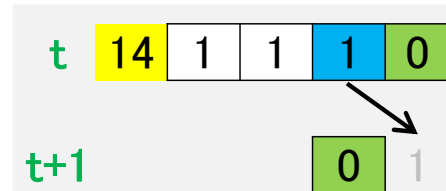
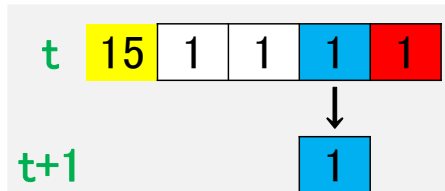
[左左近傍 | 左近傍 | 注目 | 右近傍]

時刻
t
t+1

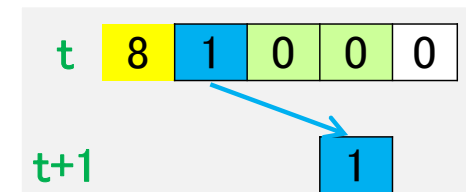
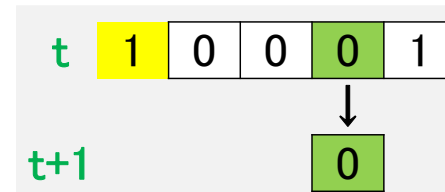
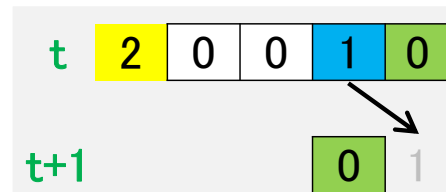
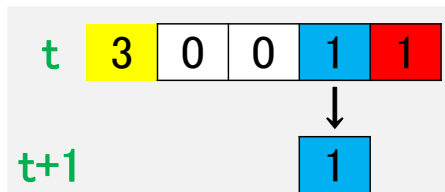
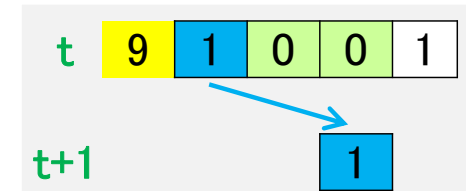


交通流・交通渋滞シミュレーション

「車両の進行規則」を1次元セルオートマトンの「ルール」へ

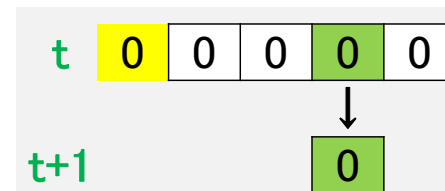


前方(注目セル)が空いているので、「左近傍」の車が前方(注目セル)に進む



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

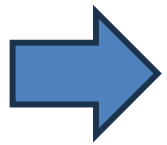


変化なし※空きのまま

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(2)

– 実行例) 3車線道路(1次元セルオートマトン×3)

※車線は各々独立で相互干渉なし(車線変更しない)

※初期配置をセル[C11:AY13]の状態からスタート

上車線は車両の道路占有率が低く(22%), 渋滞が解消されて各車両が等速(加速した速度)で進むようになる. 中央車線は, 占有率が低い(36%)が, 一部の渋滞は解消されない

	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	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB			
1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	11	22%	
2	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0	18	36%
3	1	0	0	1	0	1	1	1	1	1	1	0	1	1	0	0	1	0	0	1	0	1	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	0	0	31	62%			
4																																																									
5																																																									
6																																																									
7																																																									
8																																																									
9																																																									
10	初期配置例																																																1	道路							
11	1									1					1						1																																11	22%			
12	1	1	1	1	1	1	1	1	1																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	18	36%	
13				1	1	1	1	1	1																	1	1	1	1	1	1	1																								31	62%

Update

下1車線は、車両占有率が高く(62%),
渋滞が解消されない

Update

下1車線は, 車両占有率が高く(62%), 渋滞が解消されない

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

車線変更

セルオートマトン

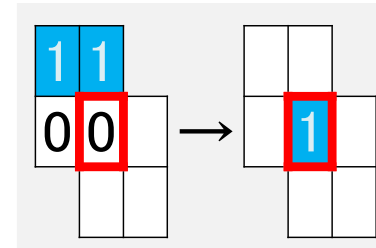
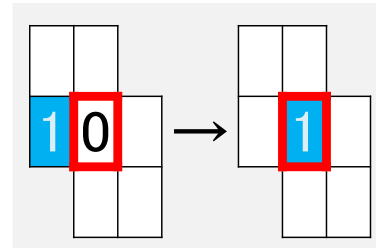
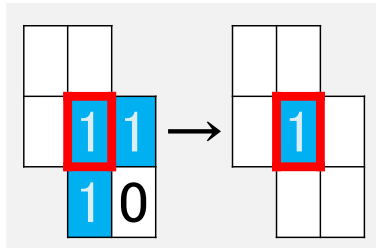
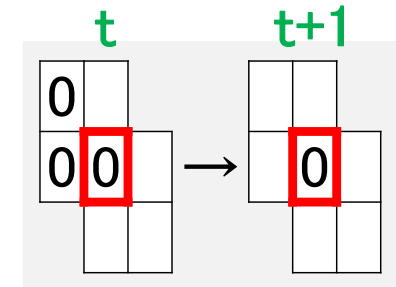
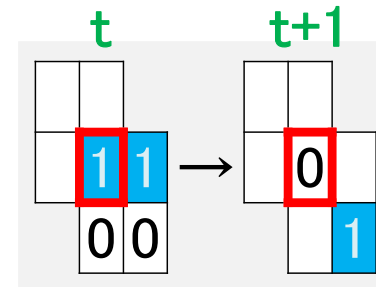
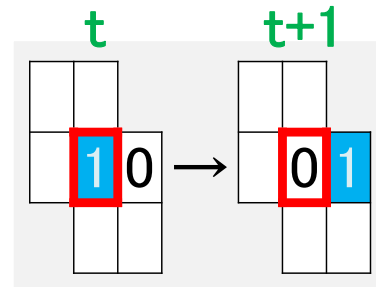
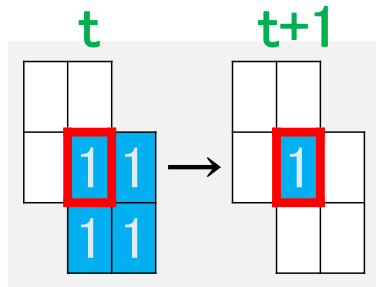
2次元セルオートマトンだが、近傍は「ノイマン近傍 + 斜め2カ所」とする
 ※[6543210]順の[7桁の2進数]で表現

6	5	
4	3	2
	1	0

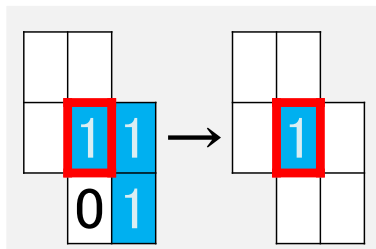
[注目セル]

交通流・交通渋滞シミュレーション(3)

「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)

$2^7=128$ パターンを
 8つに分類

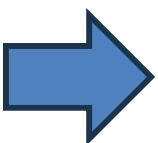
空欄の状態は [0 / 1]
 どちらの場合も含む

セルオートマトン



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
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

例は3車線だが、
2車線で実行す
れば左右への車
線変更になる

右への車線変更
が無限に可能

- ※初期配置セル[B16:AY18] で開始し, 21ステップ後

各車線の車両占有率はいずれも50%を超えるので、全車線で一部渋滞は解消されない

[illegible]

セルオートマトン

右への車線変更
が無限に可能

- 交通流・交通渋滞シミュレーション(3)

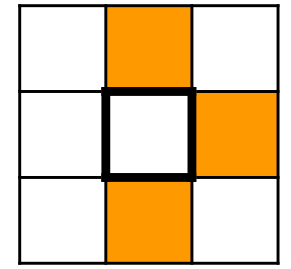
一 実行例2) 3車線道路(上下も周期境界条件)

初期配置(上)から時刻を1ずつ進めたところ(中/下)

上(左)車線, 中央車線の渋滞は, 先頭車以外は下(右)車線へ一斉に車線変更する様子

[illegible][illegible][illegible]

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)