

知の探究

3.セルオートマトン

堀田 敬介

セルオートマトン

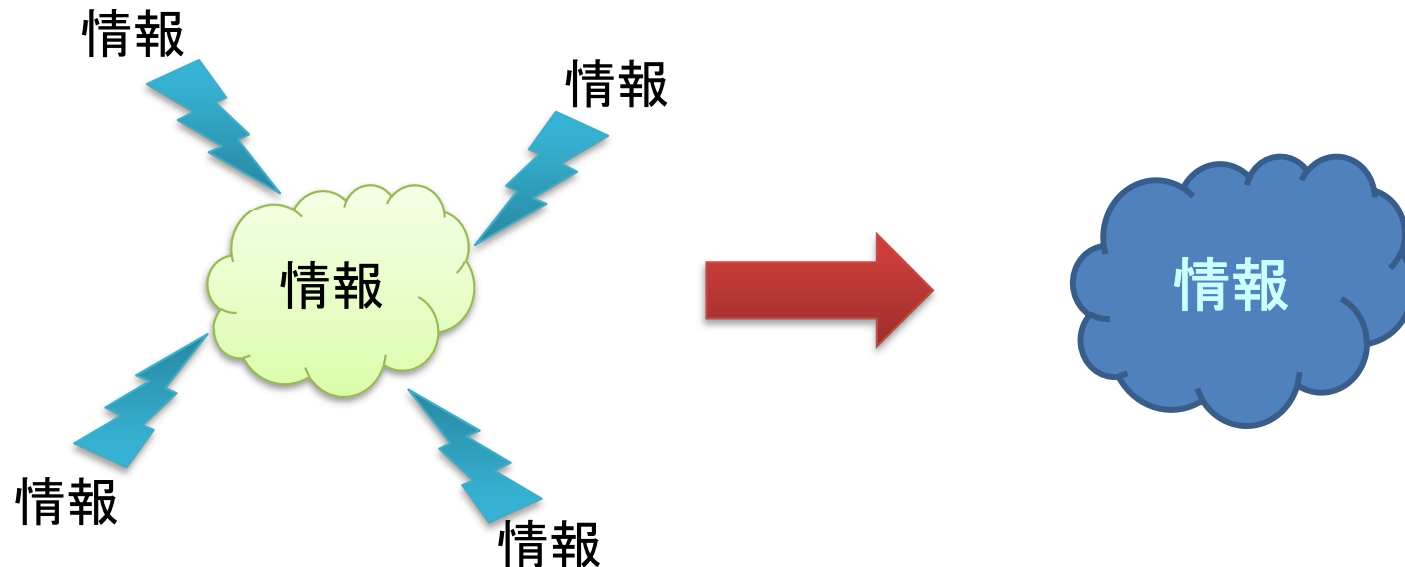
- Cellular Automata

- 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]

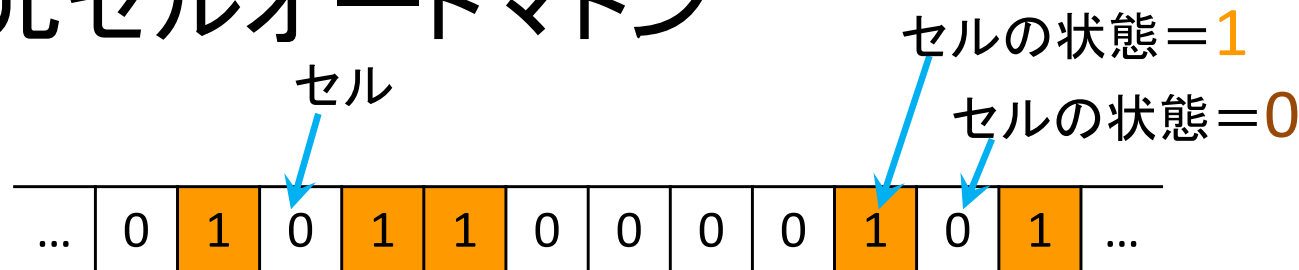
Cellular 細胞の
Automata 自動人形(からくり人形)
※Automaton(単数形)

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



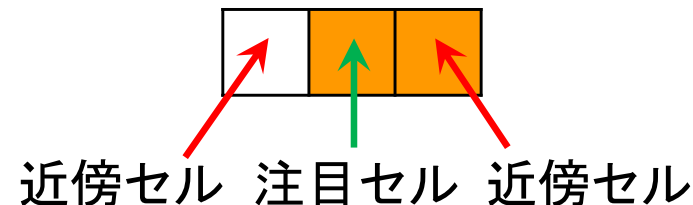
セルオートマトン

- 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

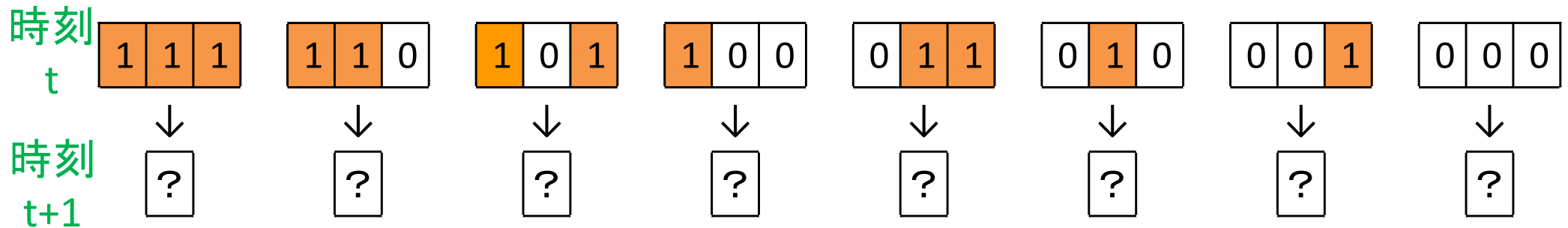
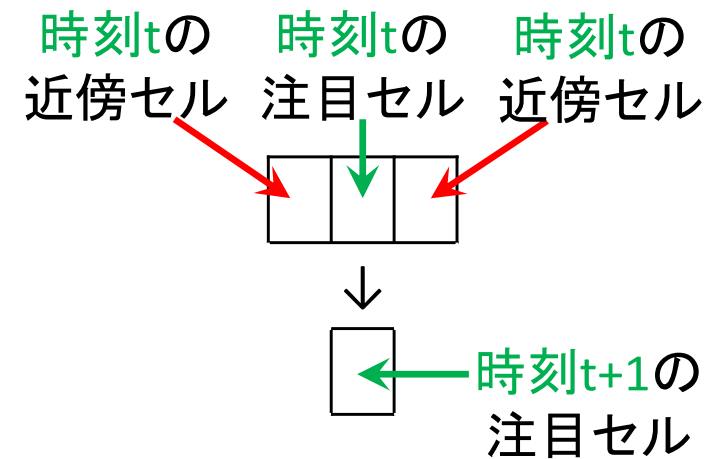


- 両端のセルの近傍の設定方法は2種類
 - 周期**境界条件 ...「左端セルの左側近傍は, 一番右端のセル」「右端セルの右側近傍は, 一番左端のセル」と考える(※ドーナツのイメージ)
 - 開放**境界条件 ...「左端セルの左側近傍は, なし」「右端セルの右側近傍は, なし」と考える

セルオートマトン

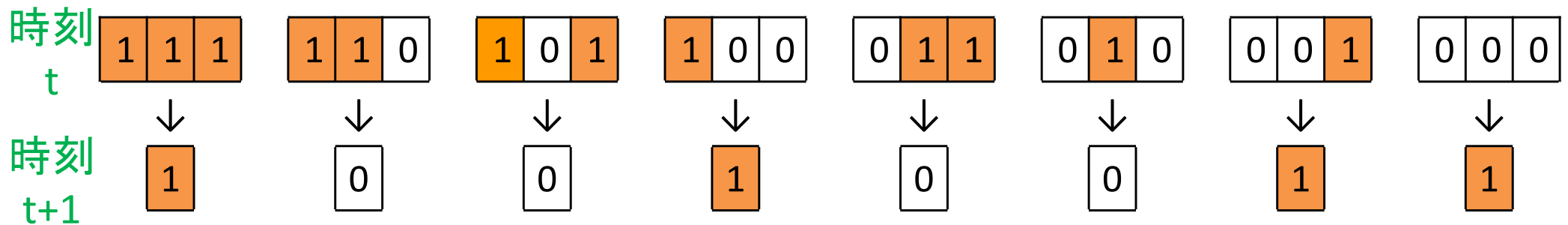
- 1次元セルオートマトン

- 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)
- 現在の状態は全部で8パターン(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

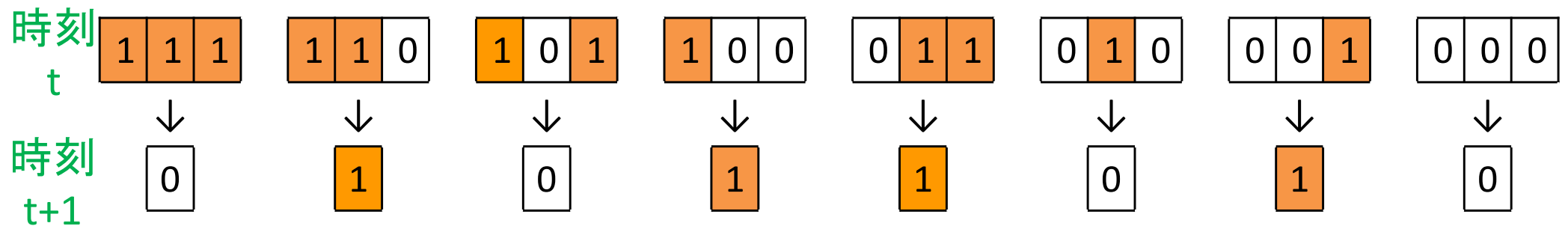
時刻 $t+1$ の考えられるパターンは全部で256個ある

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

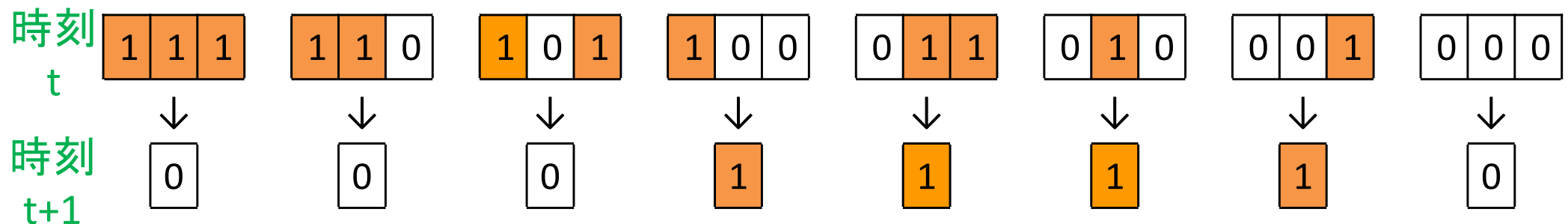
$(00000000)_2$ = 「ルール0」 $\sim$ $(11111111)_2$ = 「ルール255」

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

- 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

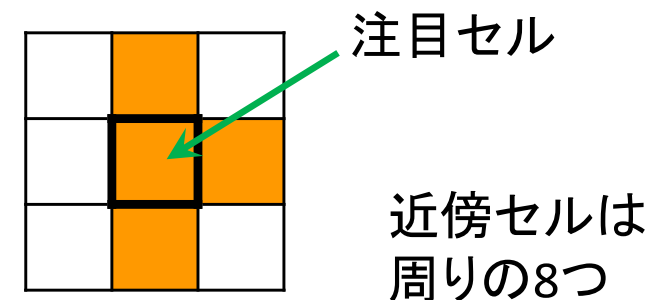
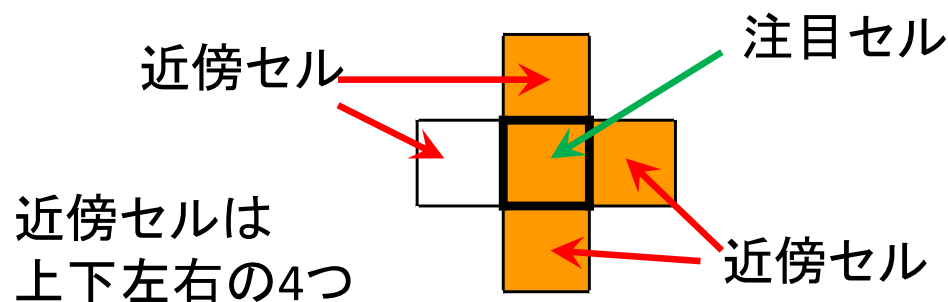
セルの状態=1
セルの状態=0

※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

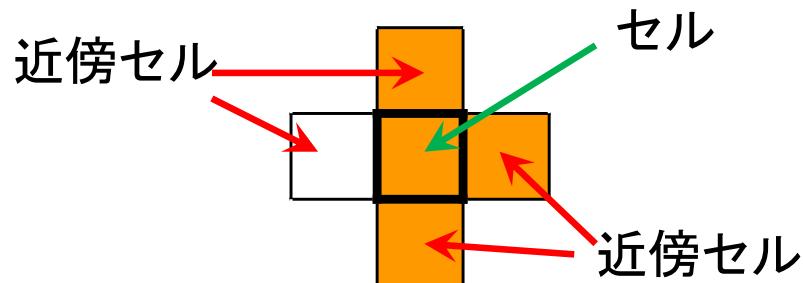


セルオートマトン

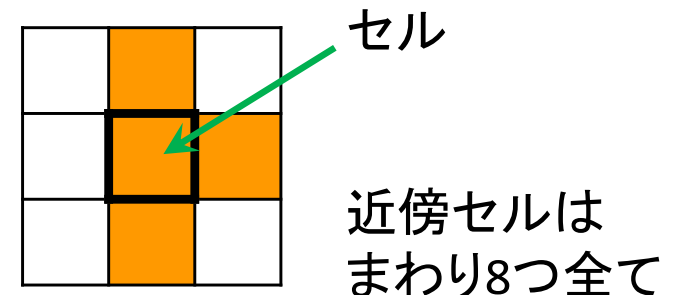
- 2次元セルオートマトン

- 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)
- 現在の状態は
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

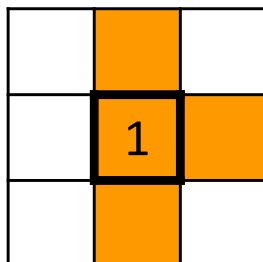
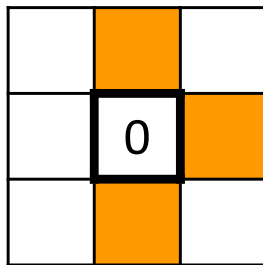
セルオートマトン

- ライフゲーム

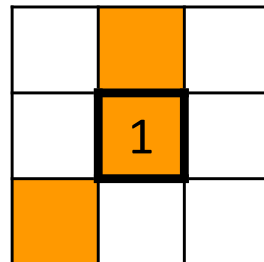
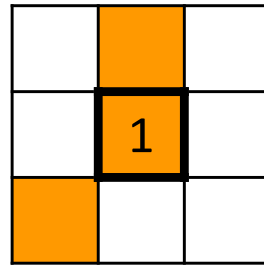
1. 瞑想...自セル=死(0) & 近傍セルの0-2,4-8つが生(1) → 0
2. 誕生...自セル=死(0) & 近傍セルの3つが生(1) → 1
3. 生存...自セル=生(1) & 近傍セルの2 or 3つが生(1) → 1
4. 過疎...自セル=生(1) & 近傍セルの0 or 1つが生(1) → 0
5. 過密...自セル=生(1) & 近傍セルの4~8つが生(1) → 0

例)

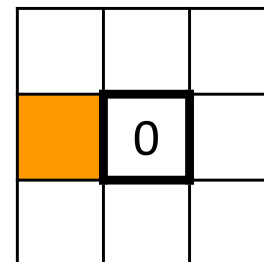
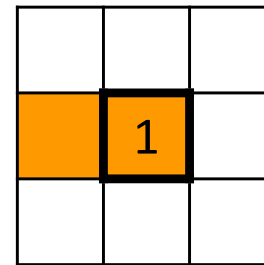
誕生



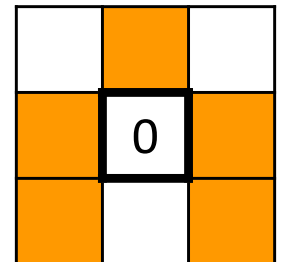
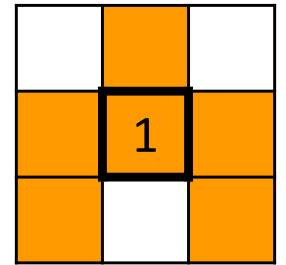
生存



過疎



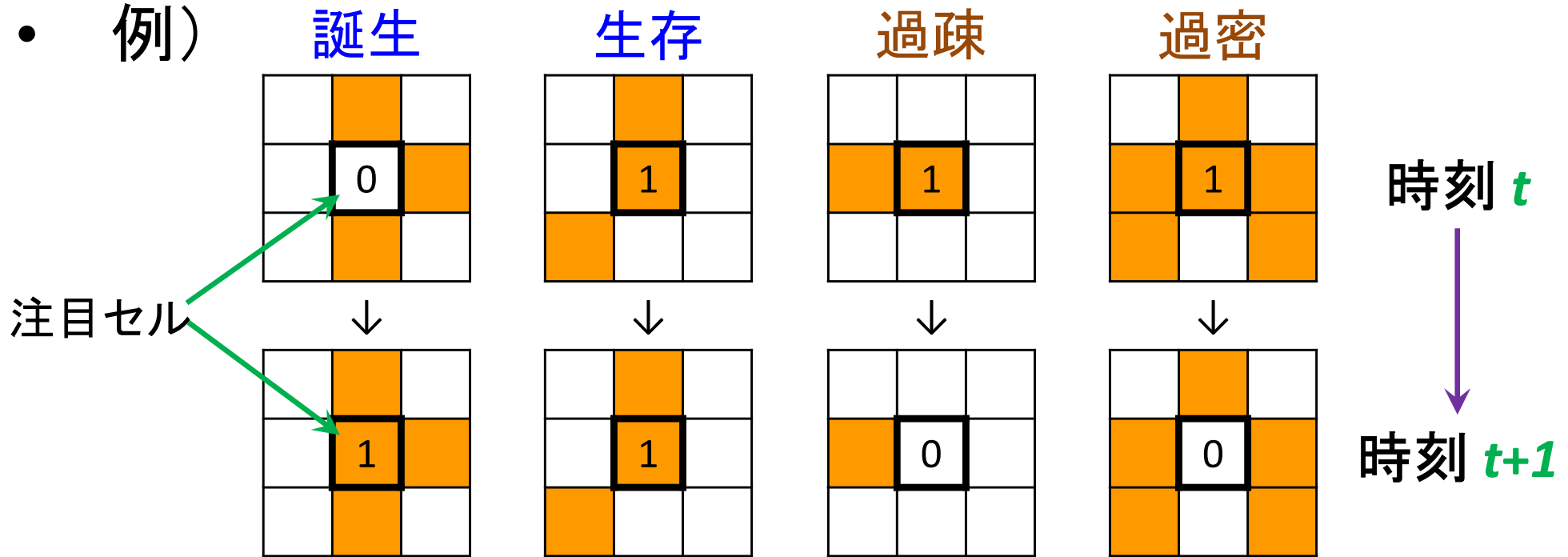
過密



※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

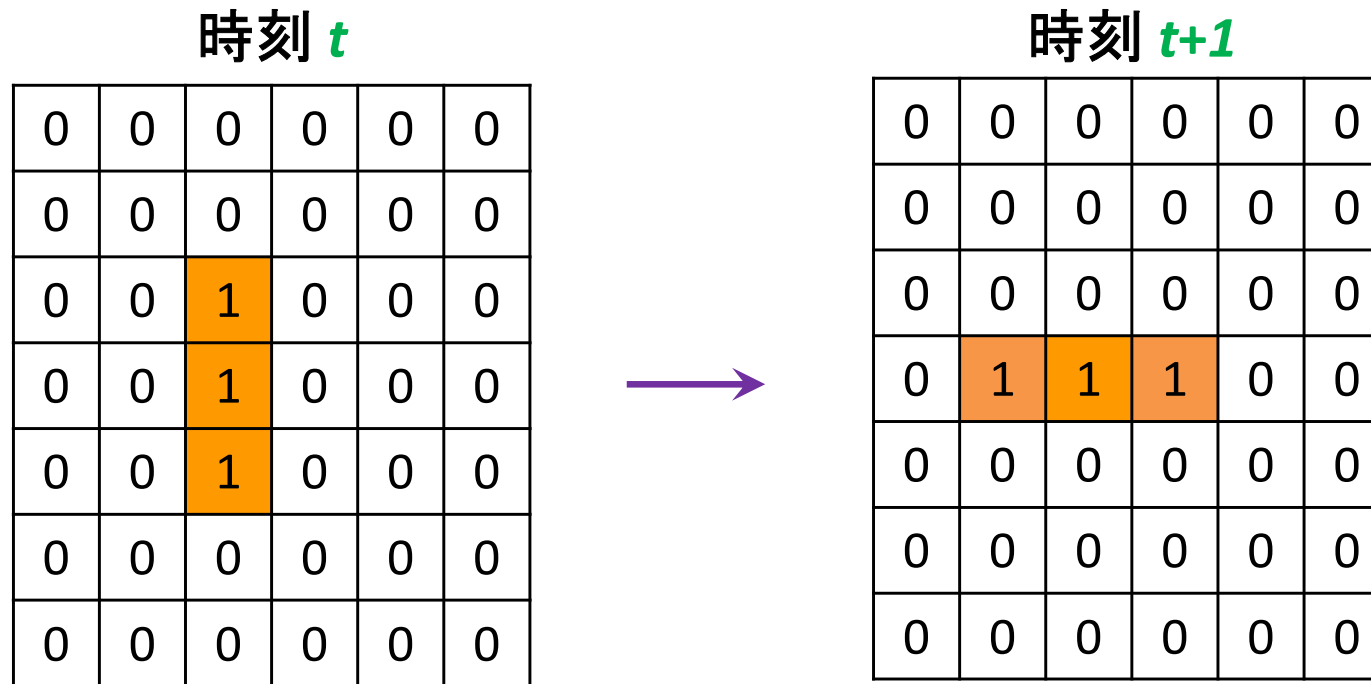
– S = 近傍セルの状態の合計 (= 生存数) とすると

		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)



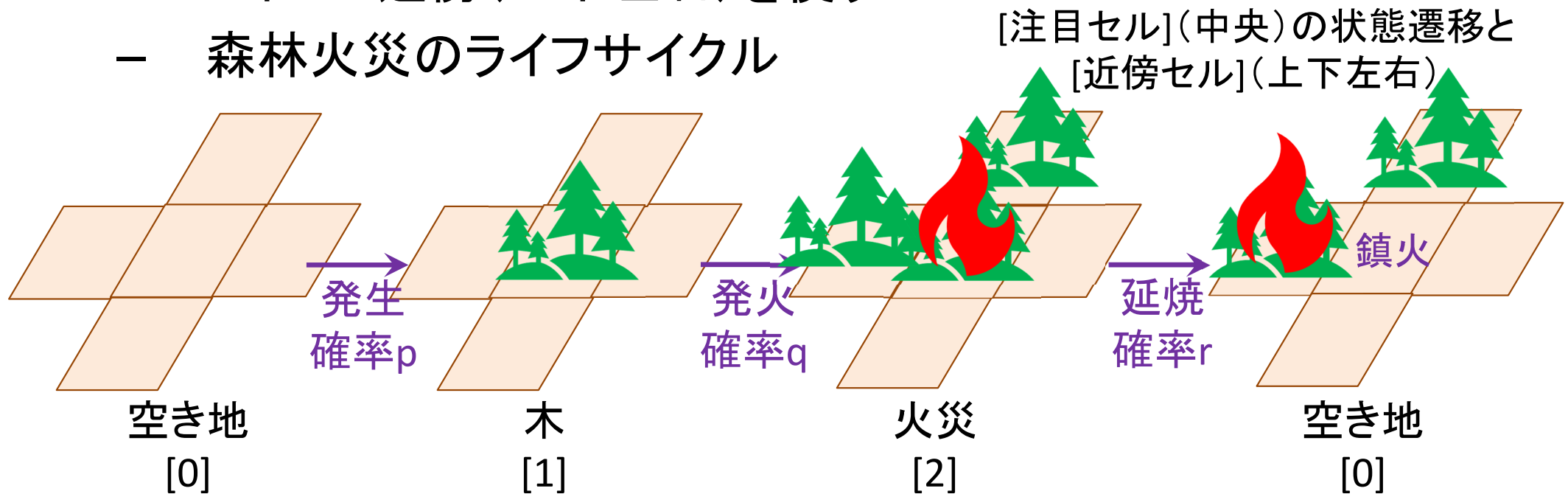
- ルールをまとめた「時刻 $t \rightarrow t+1$ 」の状態遷移表

		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

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



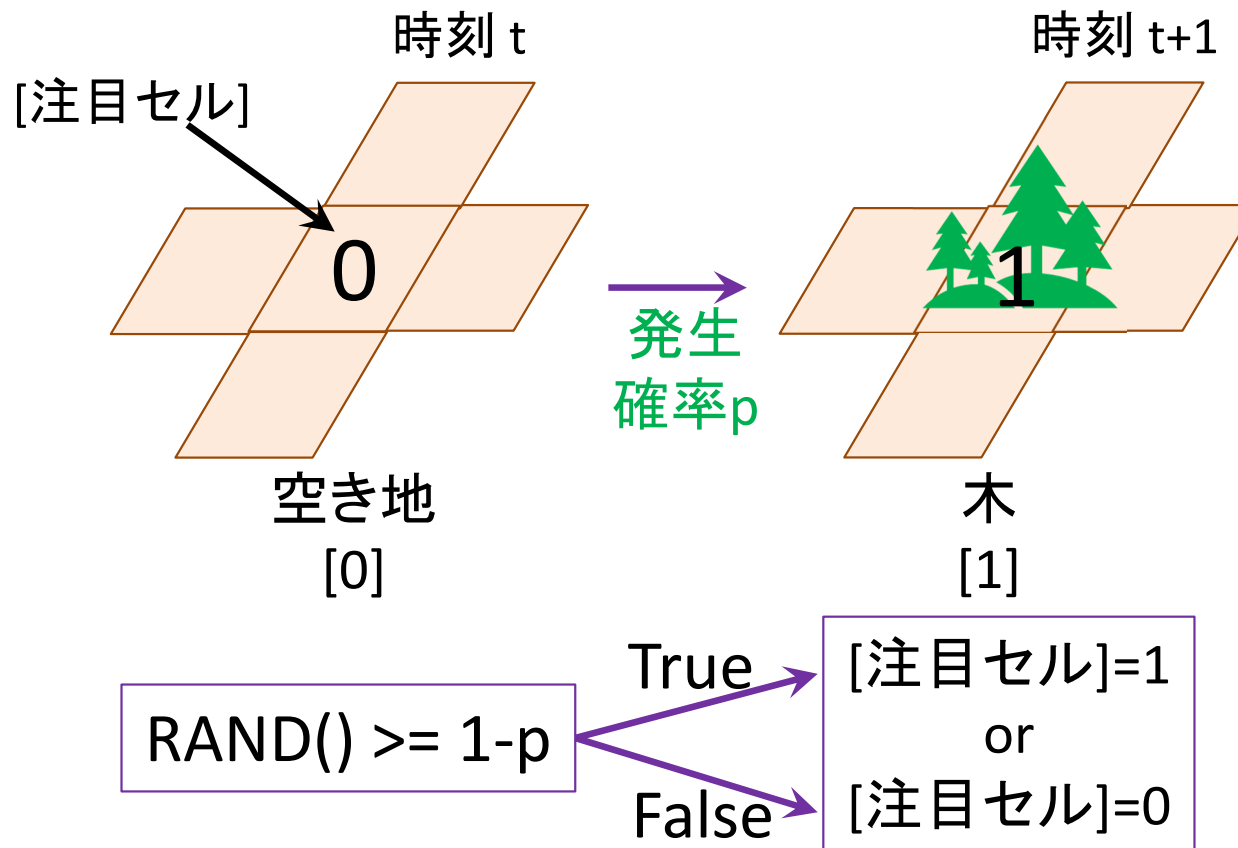
- 空き地[0]の状態では, 一定確率 p で木が発生
- 木[1]の状態かつ近傍全部非火災では, 一定確率 q で発火
- 火災[2]状態の近傍の木[1]は, 一定確率 r で延焼
- 火災[2]状態の木は, 時刻1期後に燃え尽き(鎮火)空き地[0]に

セルオートマトン

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

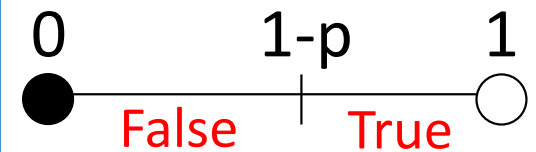


この仕組みを実現する式

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

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

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



($1-p$)以上の範囲

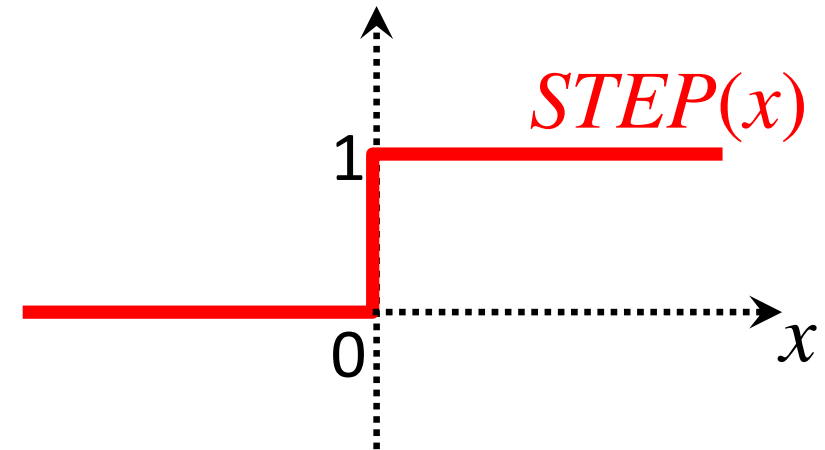
例えば $p=0.05$ なら,
生成した乱数値がこ
の範囲に入る確率が
0.05 (5%) となる

補足：ステップ関数

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

- ステップ関数(階段関数)

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



↑階段を横から見てるイメージ

ー IF文 を STEP関数を使って等価な式に書き換える

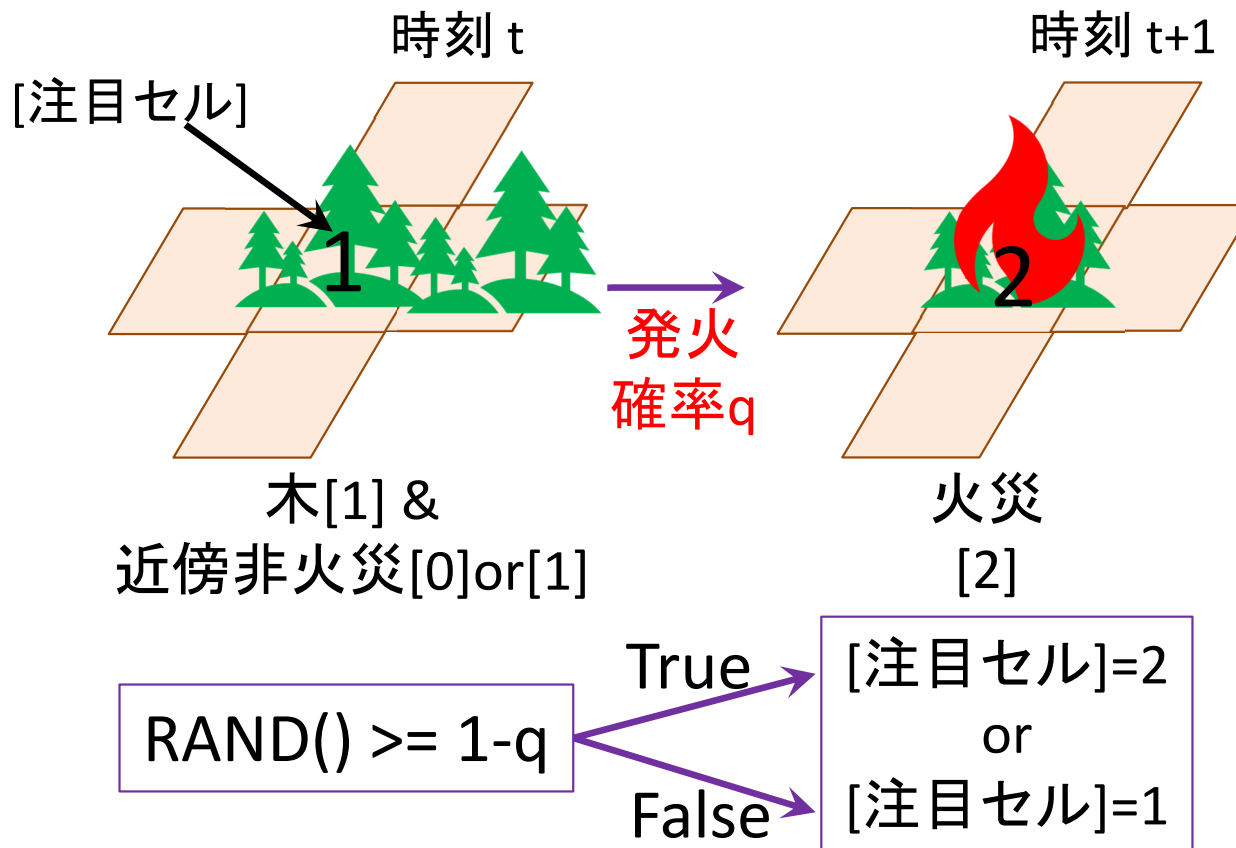
$$\text{IF}(\text{RAND}() \geq 1-p, 1, 0)$$

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

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

セルオートマトン

- 森林火災シミュレーション: 発火(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



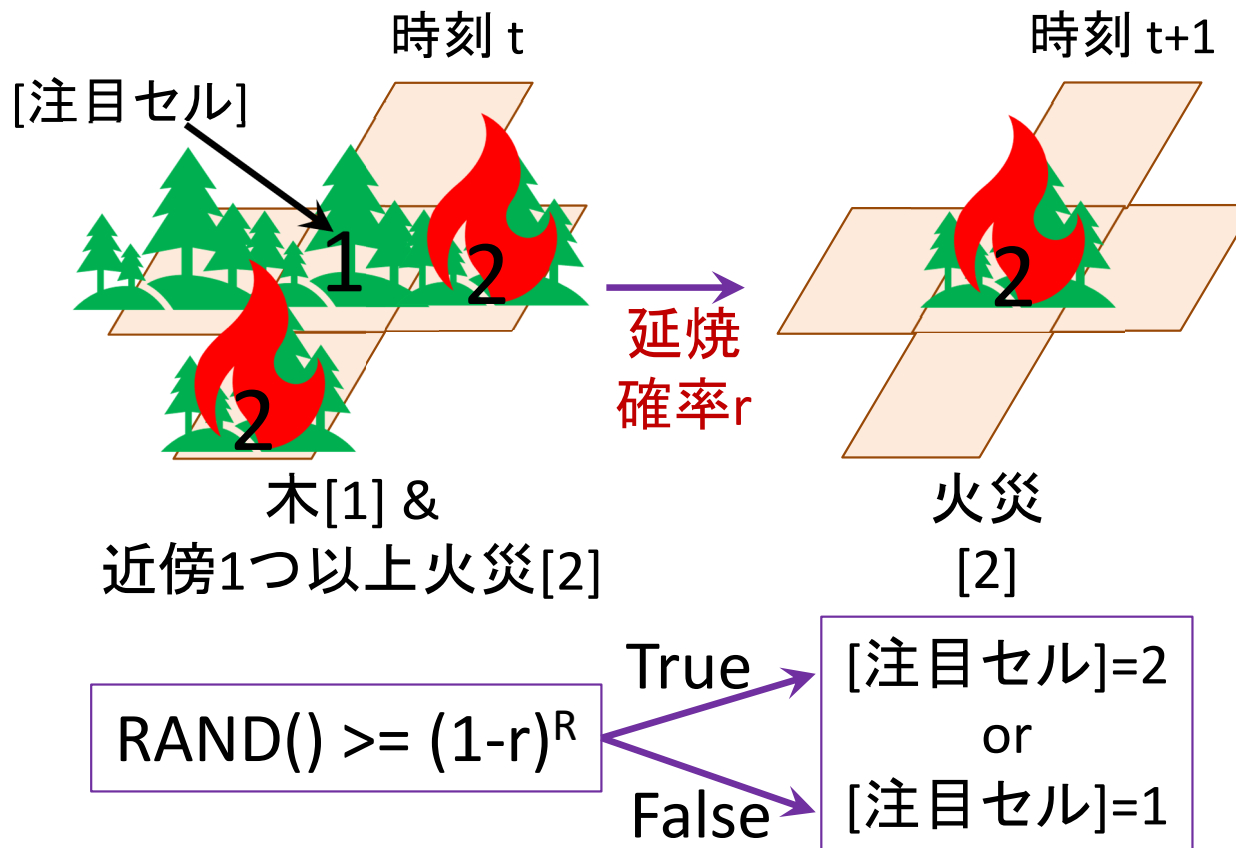
この仕組みを実現する式

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: **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**



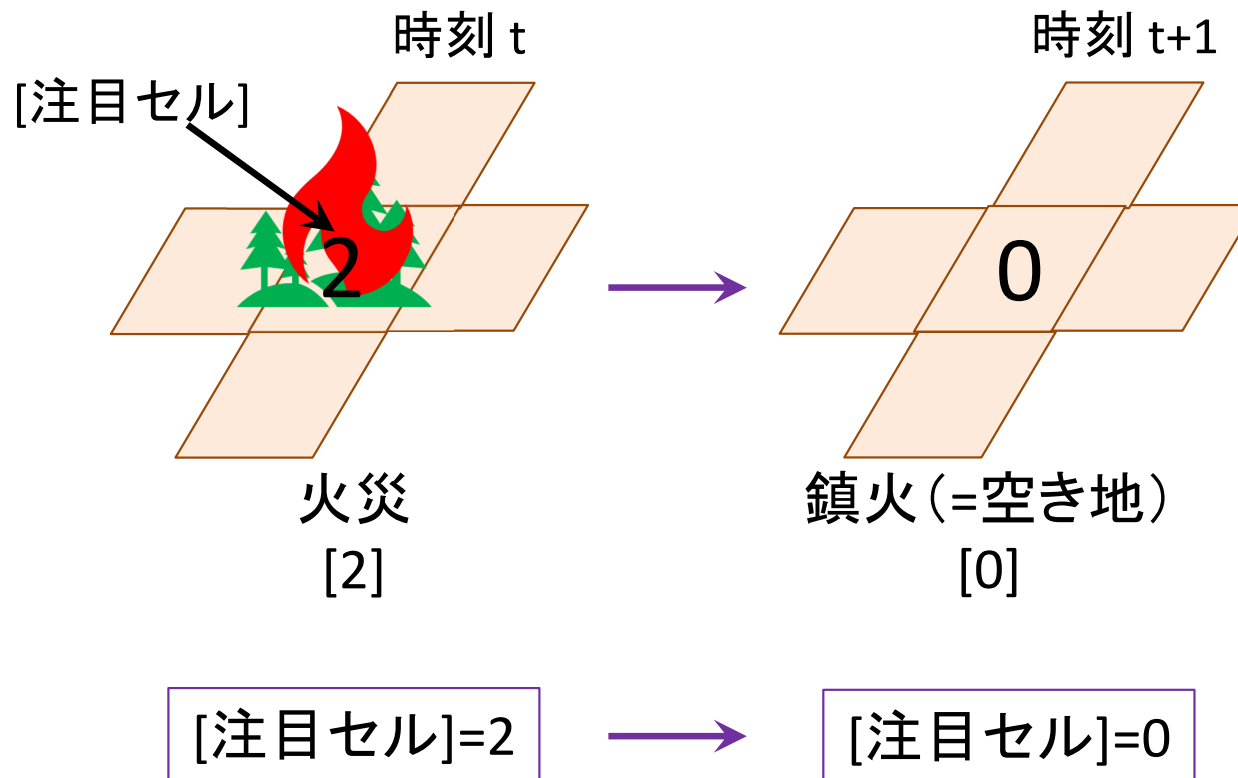
※ R =近傍の火災数1~4
(多いほど延焼しやすい)

この仕組みを実現する式

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

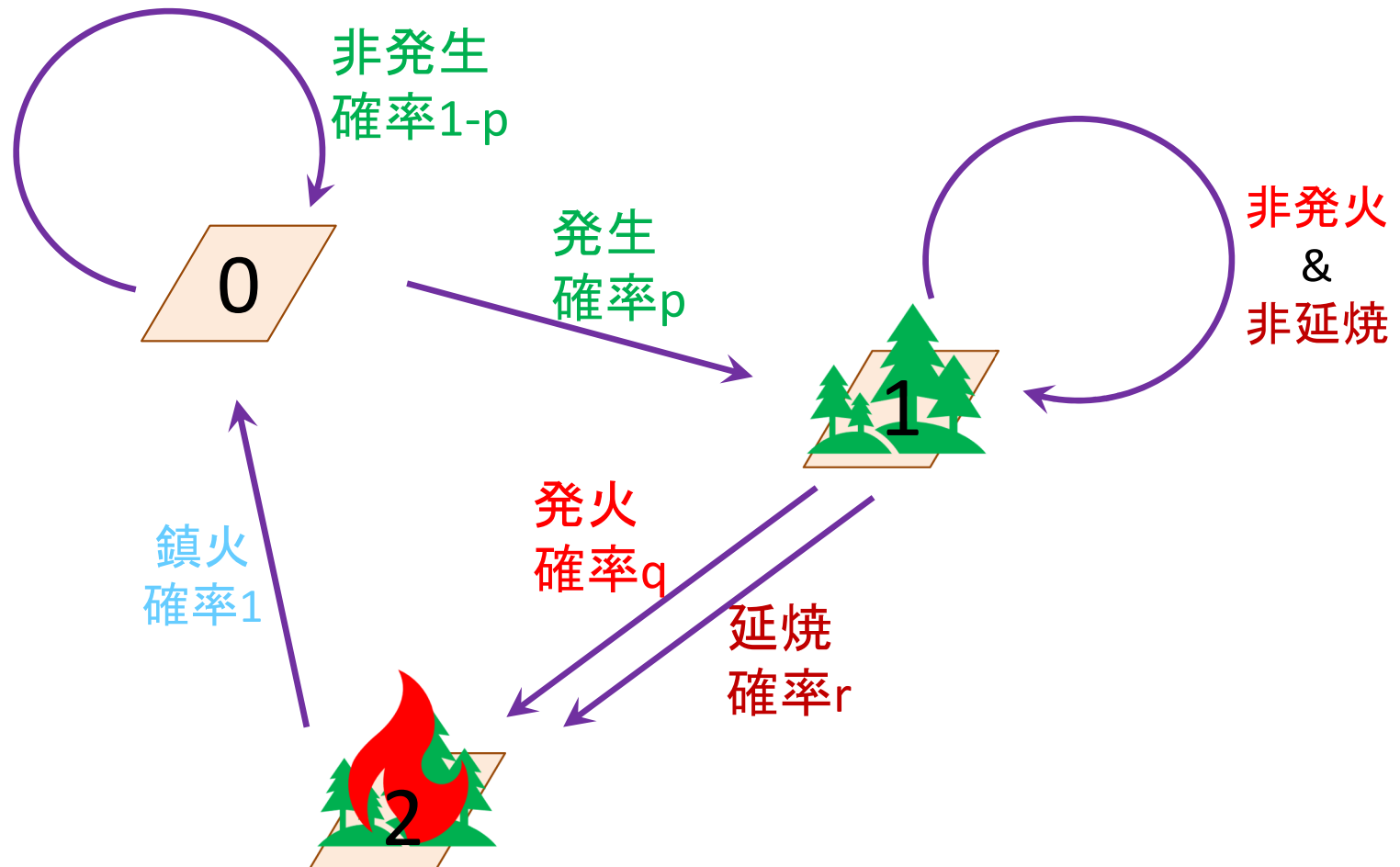
セルオートマトン

- 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

- 森林火災シミュレーション
 - 状態遷移図, life cycle



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

例) 水と油の場合の状態量の意味

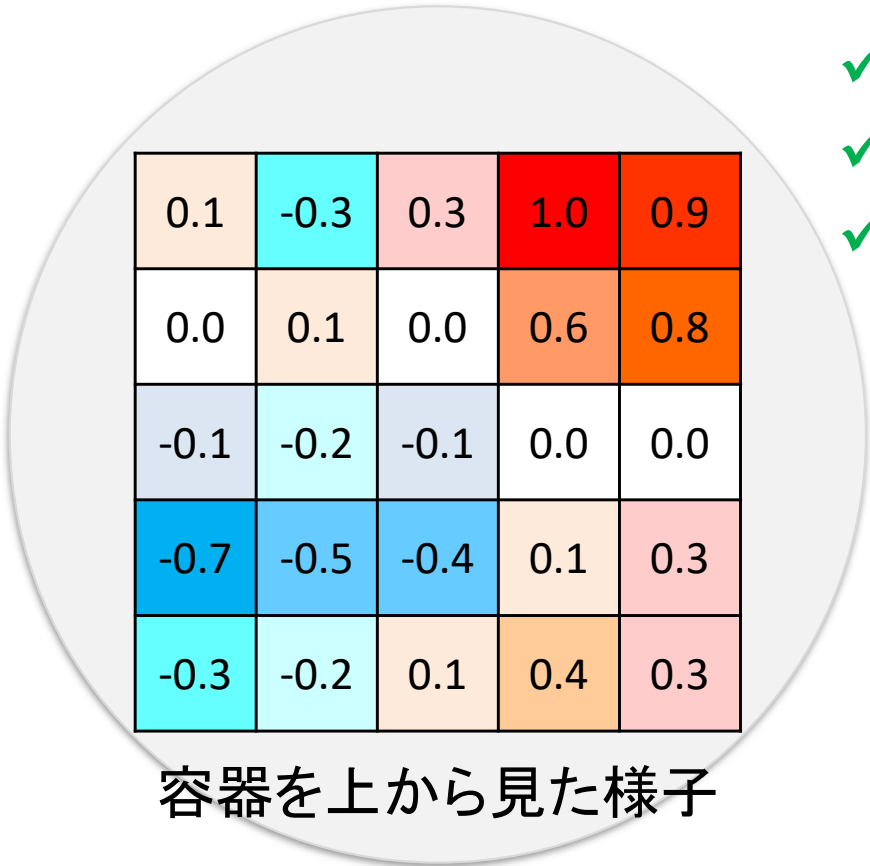
状態量	
-1.0	水だけの液体
-0.7	水多量+油少量の混合液
-0.3	水やや多+油やや少の混合液
0.0	水と油が半々の混合液
+0.3	水やや少+油やや多の混合液
+0.7	水少量+油多量の混合液
+1.0	油だけの液体

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

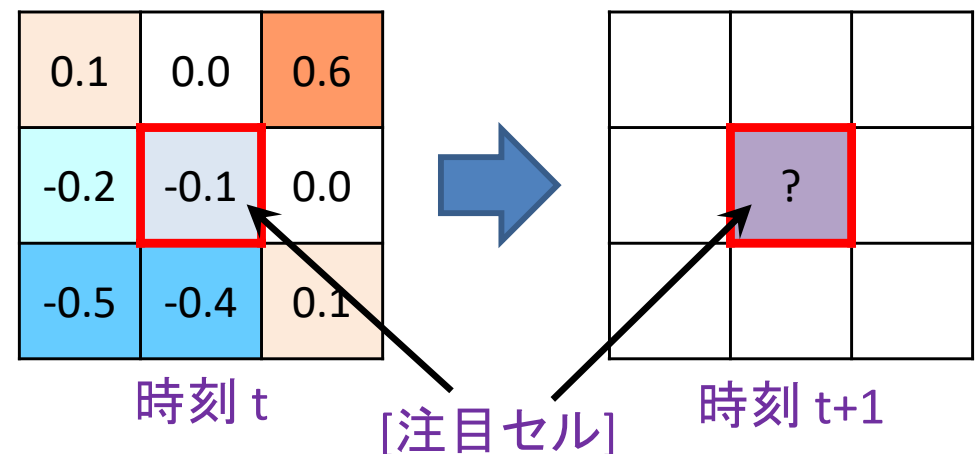
- 混ざり合わない2つの液体(水と油など)を1つの容器内で強引に混ぜ合わせた後, 時間と共に分離する様子をシミュレーションする

- ✓ セルの状態は-1.0~+1.0の連続量
- ✓ 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



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



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

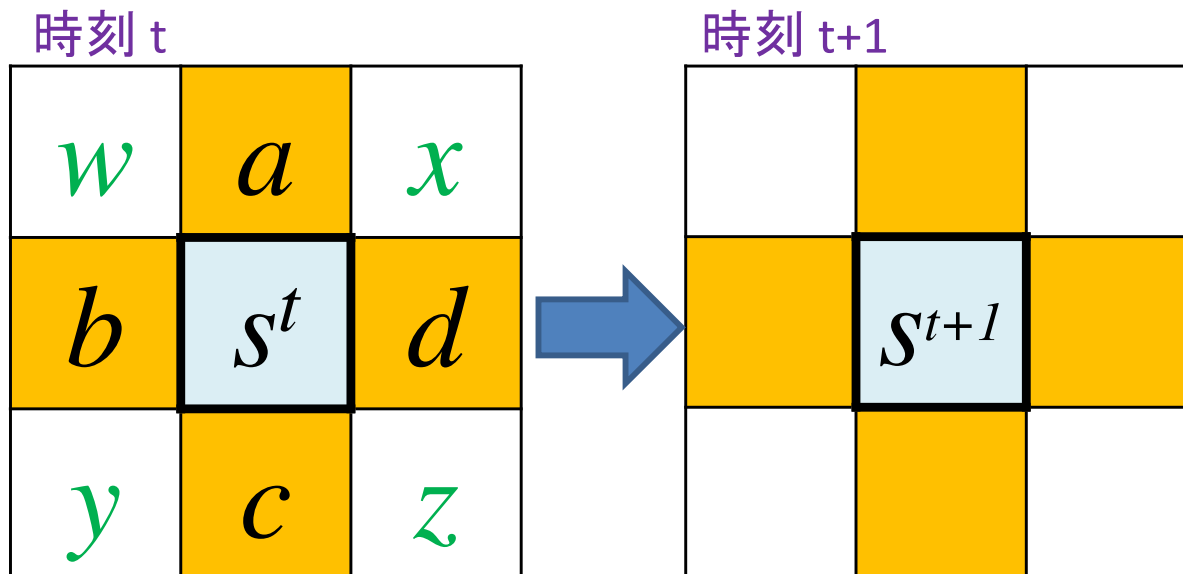
$$s^{t+1} = \mu \cdot \tanh(s^t) + \delta \cdot T$$

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

※ $\tanh()$ 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



$s^t \in [-1.0, 1.0]$ より,

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

故に $\mu=1.3$ に対し

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

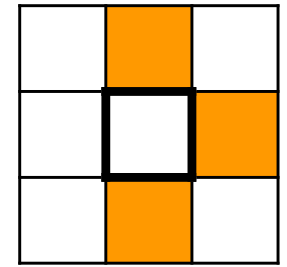
$a, b, c, d, w, x, y, z, s^t \in [-1.0, 1.0]$ より,

$T \in [-2.0, 2.0]$

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

※第1項と第2項の s^t の正負が逆であることに注意する

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)