



セルオートマトン モデリングとシミュレーション

2017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

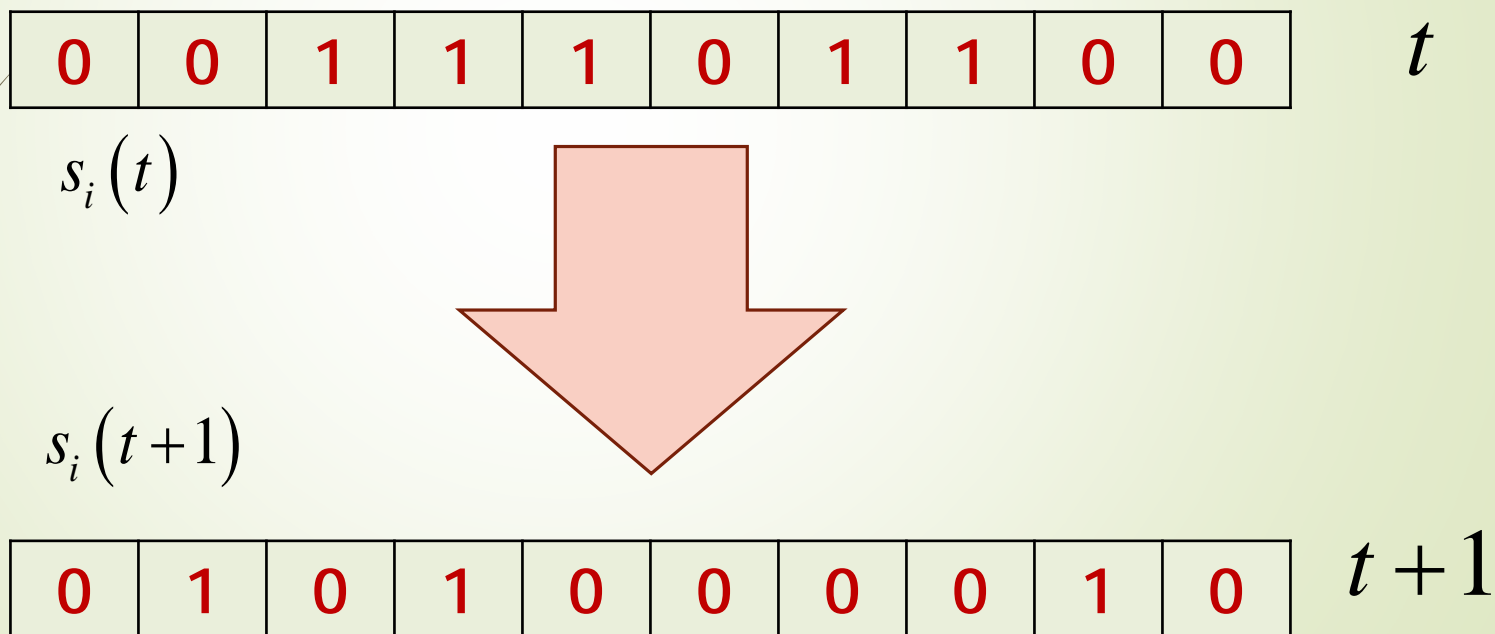
Cellular Automata

- 空間をセル(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
 - 自動機械

1次元セルオートマトンの例

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

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



1次元セルオートマトン

- ➡ 隣接する $2r + 1$ 個のセルの状態で、次の時刻の状態を決定

$$s_i(t+1) =$$

$$F(s_{i-r}(t-1), s_{i-r+1}(t-1), \dots, s_i(t-1), \dots, s_{i+r}(t-1))$$

- ➡ 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？

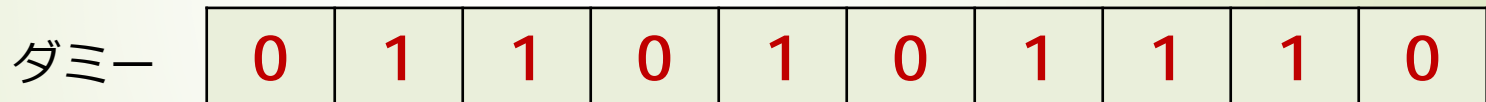
同期的更新(parallel update) 一般論

- 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、システムに書き込む

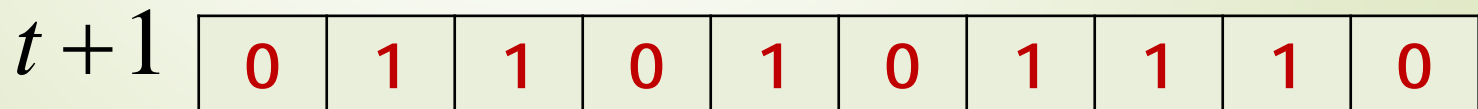
同期的更新の例



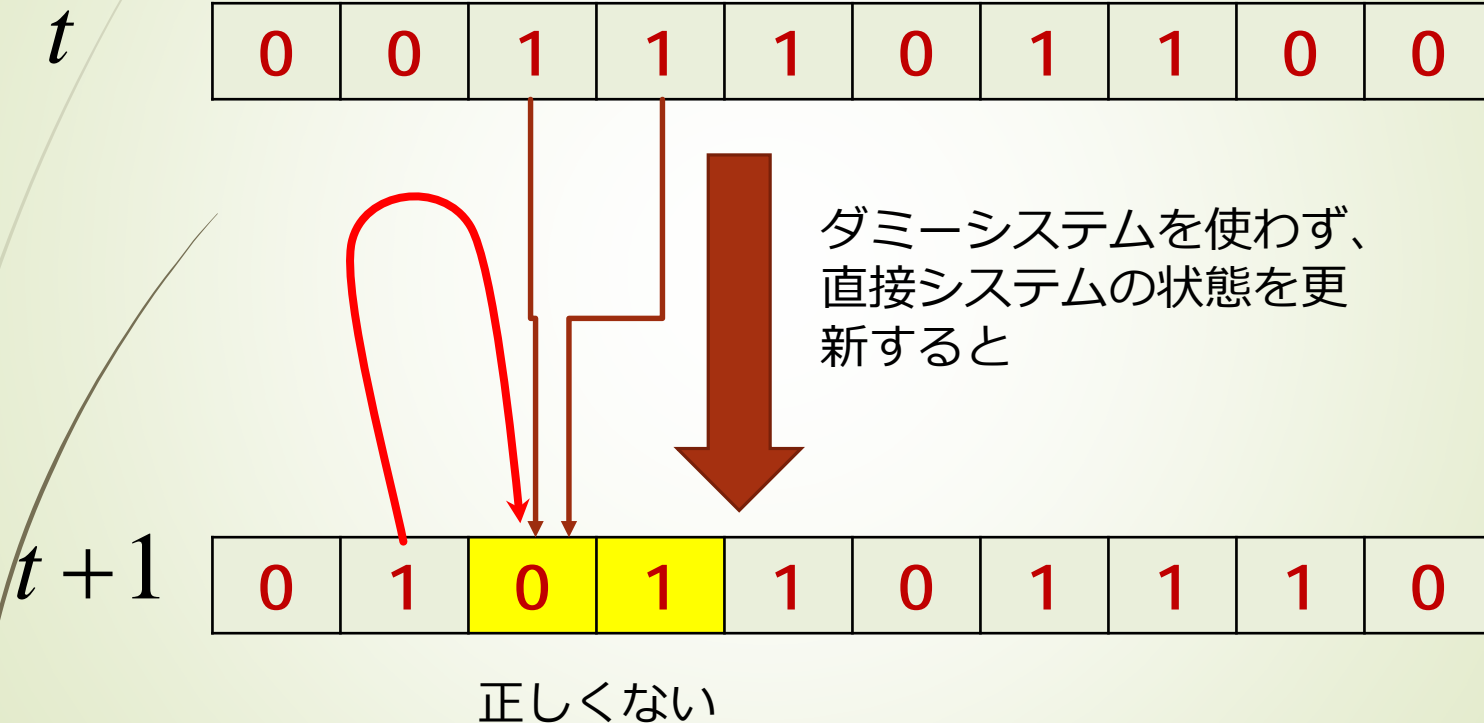
次の時刻の状態をダミーシステムに書き込む



ダミーシステムの状態を元のシステムに書き込む



例えば左端のセルから更新



最も簡単な1次元セルオートマトン

➡ 内部状態{0,1}

➡ $r = 1$

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

➡ 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

➡ $2^8 = 256$ 通り

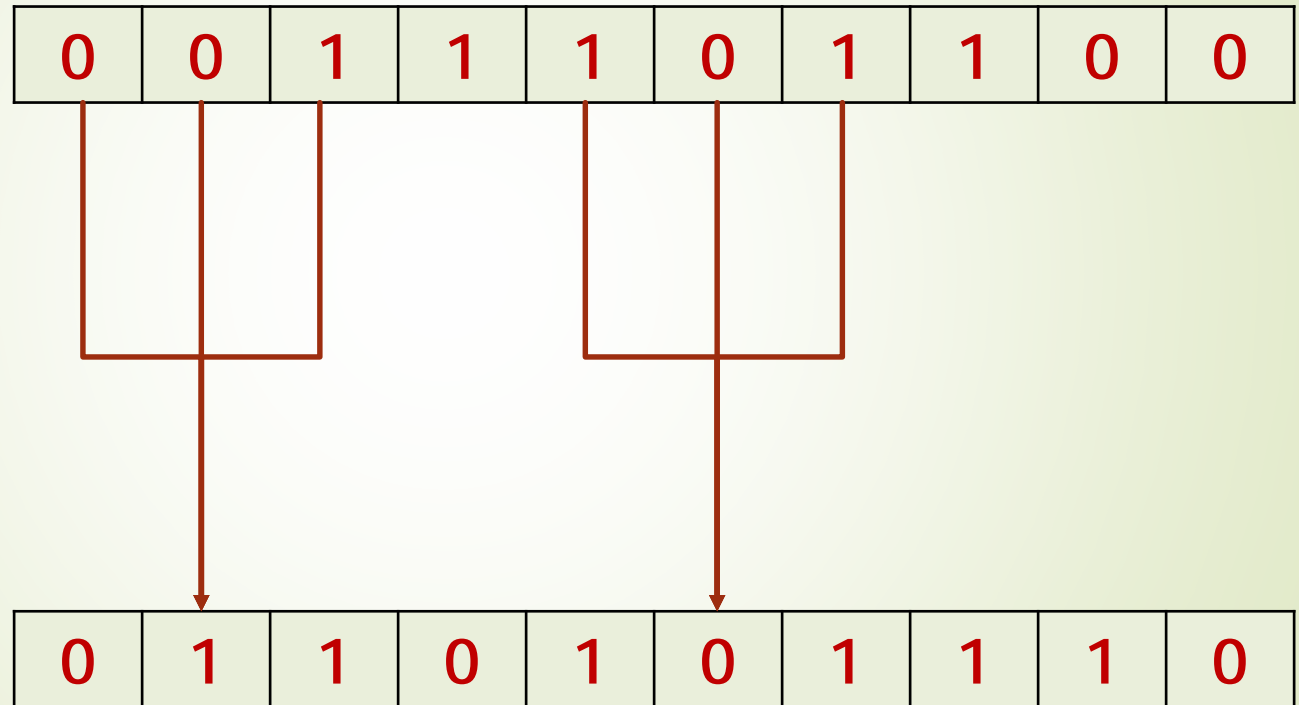
入力	111	110	101	100	011	010	001	000
出力	1	0	1	1	1	0	0	0

$$(10111000)_2 = 184$$

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

$$(01011010)_2 = 90$$

例：Rule-90 周期境界条件



実際にやってみよう

Rule-90 周期境界条件

0	0	0	0	1	0	0	0	0

実際にやってみよう

Rule-184 周期境界条件

1	0	1	1	1	0	1	0	1

Ruleから作ろう

➡ Rule 150

入力	111	110	101	100	011	010	001	000
出力								

➡ Rule 54

入力	111	110	101	100	011	010	001	000
出力								

実際にやってみよう

Rule-150 周期境界条件

0	0	0	1	0	1	0	0	0