セルオートマトンとライフゲーム

1　セルオートマトン

概要

セルオートマトン (cellular automaton) の考え方は，1950 年頃に行なわれたノイマン (J. Neumann) とウラム (S. Ulam) の議論が起源であるとされています．
基本的なセルオートマトンの特徴は以下のようになっています．

- 世界はセルと呼ばれる格子で構成されている．
- 各セルは，0(死んでいる，白，などと呼ばれる場合もある) と 1(生きている，黒，などと呼ばれる場合もある) の 2 状態がある．
- 各セルの時刻 $t+1$ の状態は，時刻 t におけるそのセルの状態およびその近傍セルの状態によって定義 (ルールと呼ばれる) される．

セルを横 1 列にならべた 1 次元セルオートマトンや，平面状にならべた 2 次元セルオートマトンなどがあります．1 次元セルオートマトンを時間軸で展開したり，2 次元セルオートマトンならそれを直接見たりすれば，そこにはある 2 次元のパターンがあらわれます．セルオートマトン理論のおもしろさは，そのパターンが，局所的に定義されたルールからは簡単に予測できないところにあります．

セルオートマトンの例

1 次元セルオートマトンの一般的なモデルは以下のようになります．

$$c_i(t+1) = f(c_{i-r}(t), c_{i-r+1}(t), \ldots, c_i(t), \ldots, c_{i+r-1}(t), c_{i+r}(t))$$

ここで，$c_i(t)$ は，時刻 t における i 番目のセルの状態を表し，f は遷移のルールを，r は考慮する近傍の範囲 (いくつ隣りまで考慮するか) を表しています．

例として，$r=1$ の場合，すなわち，自身とその両隣のセルの状態によって次の時刻の状態が決定されるような 1 次元セルオートマトン

$$c_i(t+1) = f(c_{i-1}(t), c_i(t), c_{i+1}(t))$$

について詳しくみていきます．

自身とその両隣のセルの 3 つの状態から次の時刻の状態が決まるので，ルール f は，$\{(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(0,1,1)$, $(1,0,0)$, $(1,0,1)$, $(1,1,0)$, $(1,1,1)\}$ の 8 つの各状態に対して，0 または 1 を与えるものになり，このような f は，$2^8 = 256$ 通り 考えられます．

今，このような f のうち，

$$\begin{array}{ll} (0,0,0) \to 0 & (1,0,0) \to 1 \\ (0,0,1) \to 1 & (1,0,1) \to 0 \\ (0,1,0) \to 0 & (1,1,0) \to 0 \\ (0,1,1) \to 0 & (1,1,1) \to 0 \end{array}$$

で与えられるようなルールを考えます．このルールは，$f_{01001000}$，または 10 進数に直して，f_{72} と呼ばれます．時刻 t において，自身のセルが 0 で，かつ，隣のセルのうち，どちらか片方のセルが 1 の場合にのみ，時刻 $t+1$ における状態は 1 になります．

初期状態 $(t=0)$ として，

`00100000100101001000000000`

を与え，また，両端のセルは，それぞれもう一方の端に隣接しているものとします (周期境界条件)．これにルール f_{72} を適用すると，次の時刻 (t=1) のセルの状態は，

`01010010110001101000000000`

のようになり，さらに時間を経過させると，

`10001100000101000010000000`

`01010010010001001010000001`

`00001101101010110001000010`

のようになります．

図 1 に，このルールで $t=24$ まで変化させてできたパターンを示しました．ルールと初期状態とから，どんなパターンがでてくるか，さらに言えば，あるセル $c_i(t)$ が 0 なのか 1 なのかについての予測は，$t=0$ から，順番にシミュレーションさせてみる以外の方法では，かなり難しいことがわかると思います．

図 1: 1 次元セルオートマトンの時間発展の例 (f_{72})

ウルフラムクラス

1980 年代のなかばごろ，ウルフラム (S. Wolfram) は，$r = 2$ の 1 次元セルオートマトンについて調査し，そのパターンを分類しました．

$$c_i(t+1) = f(c_{i-2}(t), c_{i-1}(t), c_i(t), c_{i+1}(t), c_{i+2}(t))$$

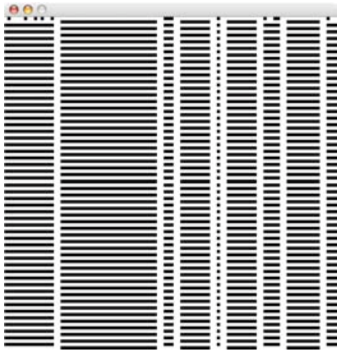
状態数の場合の数を減らすために，周りのセルについての状態和が用いられて，例えば，(0,0,0,1,1) と，(1,0,1,0,0) は同じ状態とされました．

$$c_i(t+1) = f(c_{i-2}(t) + c_{i-1}(t) + c_i(t) + c_{i+1}(t) + c_{i+2}(t))$$

状態和は，0 から 5 の 6 通りなので，ルールの総数は，$2^6 = 64$ 通り になります．ウルフラムは，これらのルールによって生成されるパターンと，微分方程式系 (離散力学系) の対応について考え，4 つのクラスに分類しました．

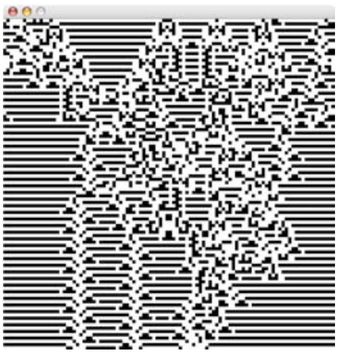
- クラス 1 - 線形系
  単一の平衡状態に収束する．つまり，全てのセルが，0 または 1 になってしまう．
- クラス 2 - 非線形系 (周期解)
  縞模様のような，周期的に無限に繰り返すパターンに収束する．
- クラス 3 - 非線形系 (カオス)
  単一の状態にも，周期的な状態にも収束せず，カオス的な振舞いをする．
- クラス 4 - 非線形系 (その他)
  予測しがたい，複雑なパターンで，周期，非周期が交互に現われたり，パターンが消えたりする．

クラス 1 の例 (f_8)



クラス 2 の例 (f_{32})

クラス 3 の例 (f_{12})



クラス 4 の例 (f_{40})

図 2: ウルフラムクラス

図 2 に，4 つのクラスの例を示します．クラス 1，クラス 2 は，秩序状態に収束します．その状態においては，任意の時刻のパターンを予測可能です．それに対し，クラス 3，クラス 4 は無秩序状態です．簡単で局所的なシステムが作りだすそのパターンは，微少時間後の予測は可能ですが，ある程度先の時刻でどのようになっているか予測することは非常に難しくなっています．

2　ライフゲーム

概要

ライフゲームは，1970 年にイギリスのコンウェイ (J. Conway) によって考案されました．最も古いコンピュータゲームのうちの一つであるとされていて，生物の誕生，死滅や地球環境，あるいは文明の興亡を単純化したシミュレーションモデルであるということができます．

安定なパターンを探したり，周期的なパターン探したり，ライフゲームの面白さは，簡単なルールから生成される出現パターンの予測の困難さ，遷移の複雑性，意外性にあります．

ライフゲームのルール

前章との関連で言えば，ライフゲームは 2 次元のセルオートマトン系です．コンウェイは，ルールを作る際に次の 3 つのことを考えました．

1. 人口が限りなく増加することが，簡単に証明できるような初期配置がないようにしたい．
2. 「一見して」無限に増えそうな初期配置があってほしい．
3. 適当ないくつかの初期配置があって，成長し，かなりの期間変化し，次の 3 つのどれかの状態に収束する．
    - 完全に死滅する．
    - 安定なパターンに固定される．
    - 周期的にいくつかのパターンを繰りかえす振動状態になる．

あるセルの次の時刻における状態は，そのセルの状態と 8 近傍 (上下左右および，斜め隣り) の状態によって決まり，次のようなルールで定義されます．

1. 現在そのセルが生きていて，かつ，8 近傍のうち 2 つまたは 3 つのセルが生きているなら，生き残る．
2. 現在そのセルが生きていて，かつ，8 近傍の生きているセルが 1 つ以下 (過疎)，あるいは，4 つ以上 (過密) なら，死亡する．
3. そのセルが生きていなくて (死んでいて)，かつ，8 近傍の生きているセルがちょうど 3 つなら，新たにそのセルに命が与えられる (誕生)．

このような単純なルールでありながら，ライフゲームの上で展開されるさまざまなパターンの複雑さは，現在でも，多くの人々を魅了しています．

有名なパターンの例

図 3 に，数多く知られている安定なパターンの中から，代表的なものを 5 つ示しました．これらは，それぞれ単独に出現した場合，どんなに時間が経過しても，決してそのパターンに変化が起こりません．例えば，一番

左のパターンについて考えてみます．このパターンは「tub(桶)」と呼ばれるパターンです．生存しているセル (黒) は 4 つありますが，それらそれぞれについての近傍の生存セルを数えてみると，2 つずつあることがわかります．ルール 1 により，これらは生き残ります．また，生きていないセル (白) について調べてみると，ちょうど 3 つの生存セルを近傍にもつようなセルはないので，ルール 3 によって新しく命が与えられるセルも存在しないことがわかります．

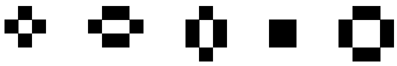


図 3: 安定なパターンの例

図 4 には，周期的に振動するパターンを示しました．上の 2 つは周期 2 でもとのパターンにもどり，下のパターンは，周期 8 の例です．もちろん，これ以外にも周期的なパターンはたくさん存在します．興味があれば，いろいろ探してみてください．

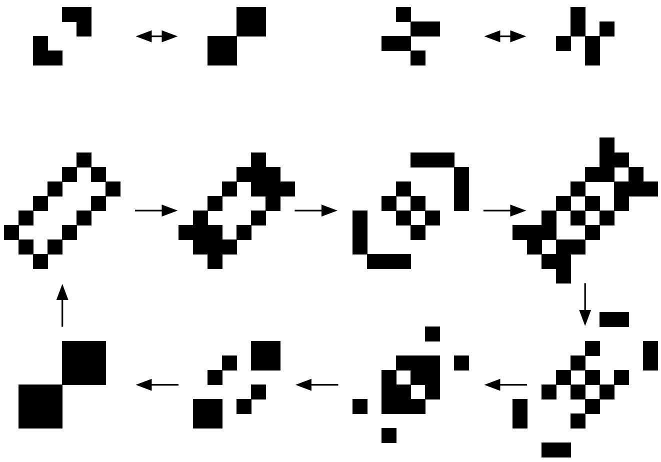


図 4: 周期的なパターンの例

最後に，図 5 は,「glider(グライダー)」と呼ばれるパターンです．このパターンはとても興味深い「4 ステップで右下に 1 つずつ移動する」という動き方をします．

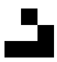


図 5: グライダー