

スマートインフォメディアシステム特論
Smart-Info Media System

第4回 セルオートマトン

4th week : Cellular Automata

延原 肇 (知能機能システム専攻)
Hajime Nobuhara @ iit

 筑波大学
University of Tsukuba

授業のページ (Class web page)
<http://www.nobuharaken.com>



研究テーマ一覧

資料や課題は、すべてここに掲載

[nobuharaken.com](http://www.nobuharaken.com)

2023年度スケジュール

[4/17] 第1回 ガイダンス (Guidance)
[4/24] 第2回 フラクタル (Fractal)
[5/08] 第3回 カオス (Chaos)
[5/15] 第4回 セルオートマトン (Cellular Automata)
[5/22] 第5回 パーコレーション (Percolation)
[5/29] 第6回 複雑ネットワーク (Complex Networks)
[6/05] 第7回 マルチエージェント (Multi-agents)
[6/12] 最終課題のため自習期間
[6/19] 最終課題のため自習期間
[6/26] 最終課題提出

4

課題はmanabaから提出すること



本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの数学的定義

Mathematical Aspects of cellular automata

セルオートマトンとカオスの縁

cellular automata and chaos

多次元セルオートマトン

Multi-dimensional cellular automata

渋滞学との関連

Cellular automata and traffic jams

本日の構成

オートマトンとは？

What is automata?

状態と遷移と入出力

States/Transition/Input and Output



状態 = 財布の中身

遷移 = 入出力によって財布の中身が変化



状態 = お腹

遷移 = 入出力によってお腹が変化

あらゆる対象を「状態」「遷移」「入出力」で記述

Our purpose is to describe the relation between
the input and output based on states and transition.

オートマトンとは？

What is Automata?

Automata = からくり人形, 自動人形

工学の言葉では？

有限オートマトン



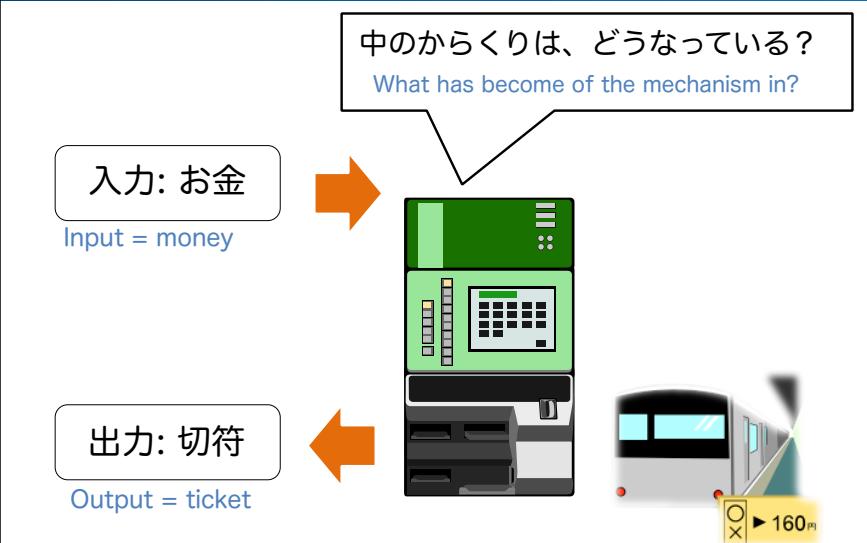
Finite Automata

有限個の離散的な内部状態を持ち、入力に応じて状態遷移し、出力する論理的な機械

Finite automata have an internal state of a discrete finite number, and they output the state changes in response to input

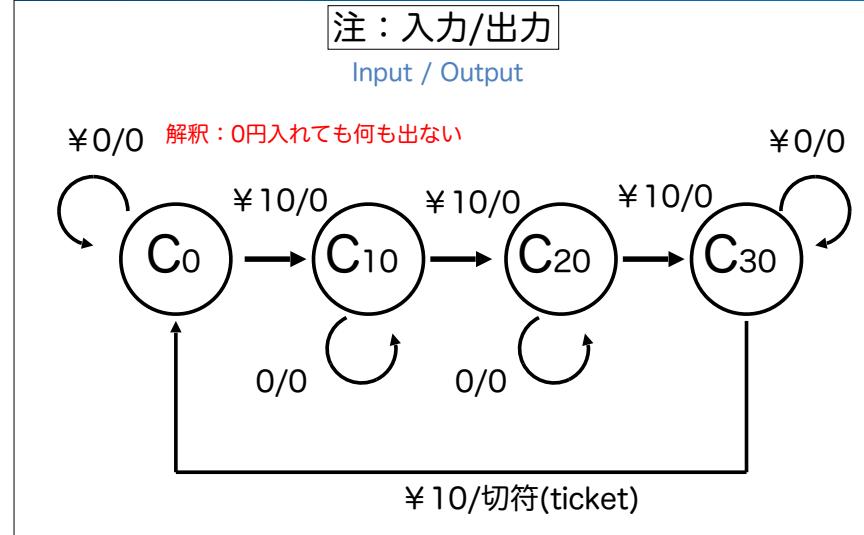
例：40円の切符販売機の有限オートマトン

Finite automaton of ticket vending machines of 40 yen



例：40円の切符販売機の有限オートマトン

Finite automaton of ticket vending machines of 40 yen



例：40円の切符販売機の有限オートマトン

Finite automaton of ticket vending machines of 40 yen

有限オートマトンの別表現

前の内部状態	状態遷移関数F State transition function		出力関数R Output function	
	入力0	入力10	入力0	入力10
C ₀	C ₀	C ₁₀	0	0
C ₁₀	クイズ	クイズ	0	0
C ₂₀	C ₂₀	C ₃₀	0	0
C ₃₀	C ₃₀	C ₀	クイズ	クイズ

有限オートマトンの計算可能性

世の中のたいていのものは
有限オートマトンで記述可能



世の中のたいていのものは
コンピュータで計算可能

自己複製（増殖）オートマトン

Self-replicating automata

従来のオートマトンは**静的**

(内部状態、出力関数などは**固定**)

Conventional automata are static, that is, fixed internal state, such as output function.

「(静的・固定の) コンピュータに
自己修復機能を与えられないか?」

Neumann was trying to develop the self-replicating computer?



ジョン・ノイマン
(John von Neumann)

セルオートマトン研究の動機

This is the motivation of cellular automata

自己複製（増殖）オートマトン

Self-replicating automata

ノイマンの理想：

周囲の脳細胞から栄養をもらい、細胞分裂を行い、死滅した細胞に変わって次々に新しい脳細胞が役割を担ってゆくような機能の実現



Necessary functions:

To get information nutrients from the surrounding brain cells, sometimes, cell division, and a new brain cells play a role in succession on behalf of the cells were died.

本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

3分でわかる

セルオートマトン

3-min super-understandable-introduction



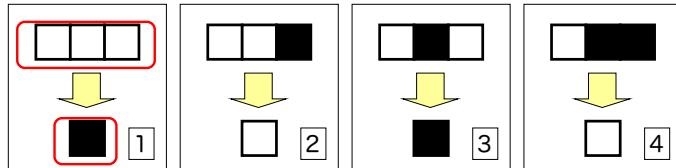
セルオートマトンの例 (1)

Example of cellular automata (1)

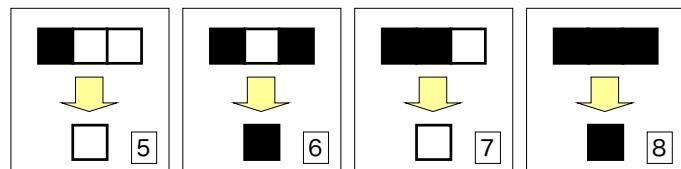
準備するもの=状態遷移ルール

Preliminary = state transition rules

全ての組み合わせ (白黒3つなので8状態) : 3 cells and white/black = 8 states



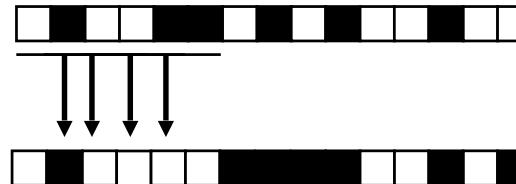
こちらは自由: we can define freely.



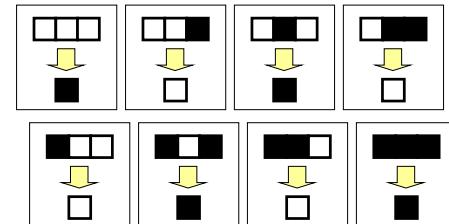
セルオートマトンの例 (2)

Example of cellular automata (2)

初期状態 (initial state)



端っここの処理は
折り返して考える



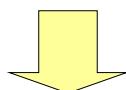
セルオートマトンの例 (3)

Example of cellular automata (3)

初期状態 (initial state)



次の状態 (next state)



与えられた遷移規則に従い各セルの状態を更新

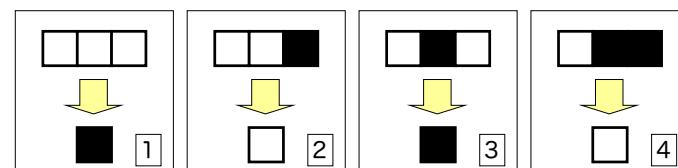
Update the state of each cell in accordance with the transition rules

自己修復との関連

Self-repairing and cellular automata

状態遷移ルール

Preliminary = state transition rules



次の状態は、両隣の状態に左右されている

The next state is dependent on the state of both sides

周囲の脳細胞から栄養をもらい、細胞分裂を行い、死滅した

細胞に変わって次々に新しい脳細胞が役割を担ってゆくような機能

Cellular automata = get information nutrients from the surrounding brain cells, sometimes, cell division, and a new brain cells play a role in succession on behalf of the cells were died.

以上、おしまい。めでたし、めでたし。

つまり、ほぼセルオートマトンは理解できました！

That is, we could understand cellular automata! (almost, 3min)



セルオートマトンの応用対象

Applications of cellular automata



これらはすべて
セルオートマトンで記述可能



These phenomena can be modeled by the cellular automata.

セルオートマトンの計算能力

Computational capability of cellular automata

こんな簡単な計算モデルでどんな対象を計算できる？

Actually, the cellular automata are quite simple computational model, so we are wondering what kind of computation can be done?



本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの歴史 (1)

History of C.A. (1)

ジョン・冯・ノイマン

(John von Neumann, 1903年12月28日 - 1957年2月8日)

20世紀の天才を挙げるとすれば必ず名前が登場

His name always comes up if you mention a genius of the 20th century

ハンガリー生まれの天才数学者

- 量子力学 (Quantum mechanics)
- ゲーム理論の創始者 (Game theory)
- ノイマン型コンピュータ (Neumann computer)



1946年春 セルオートマトンモデルを提案

In 1946, Neumann proposed the concept of cellular automata

セルオートマトンの歴史 (2)

History of C.A. (2)

ジョン・ホートン・コンウェイ

(John Horton Conway, 1937年12月26日~)

イギリス生まれの天才数学者

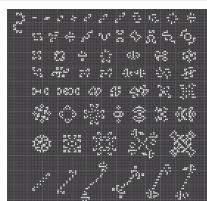
※散らかっているのが大好き (らしい)

He is a genius mathematician born
in the UK and likes messy place...



ライフゲームの提案

He proposed Life GAME.



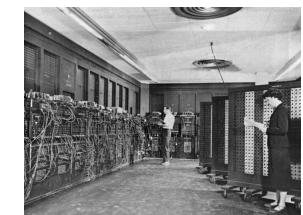
なぜノイマンは自己複製という機能に着目?

Why did Neumann focus on the function of self-replication?

当時 (1946年頃) のコンピュータの様子



<http://ja.wikipedia.org/wiki/真空管>



出典: <http://ja.wikipedia.org/wiki/ENIAC>

-コンピュータの部分的な故障に翻弄

There were the partial failures of the computer, frequently.

-自己修復できるコンピュータの切望

They really hope a computer that can self-repair if possible realization.

ライフゲームみてみましょう

セルオートマトンの歴史 (3)

History of C.A. (3)

スティーブン・ウォルフラム

(Stephen Wolfram, 1959年8月29日～)

神童と呼ばれ、14歳で素粒子論の論文を執筆

He is called a child prodigy, and writing papers of particle physics at the age of 14



- ・マッカーサー財団の天才助成金給付研究員の第1期生
- ・Wolfram Research社の創業者
- ・Mathematica の開発者

Founder of Wolfram Research and developer of Mathematica

1次元セルオートマトンモデルの提案

He proposed 1D-cellular automata model

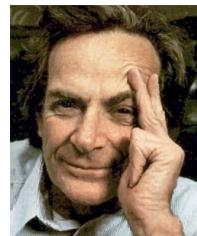
セルオートマトンの周辺

Another stream of C.A.

リチャード・ファインマン (物理学者)

(Richard Phillips Feynman,
1918年5月11日 - 1988年2月15日)

量子電磁力学の発展に寄与



※1 朝永振一郎、シェヴィンガーとともにノーベル賞受賞

「ご冗談でしょう、ファインマンさん」シリーズが有名
(時間のある人、読んでみるとよいでしょう)

http://en.wikipedia.org/wiki/Richard_Feynman

セルオートマトンの歴史の流れ

多くの天才たちを魅了

C.A. attracts a lot of geniuses



- ・セル空間という抽象的な空間を設定
(均等な格子ではない) 2次元セルオートマトンに近い
※挙動が複雑でよくわからない

Actually, the behavior of C.A. is too complex, nobody understand it.

- ・セル空間を均等な格子として定式化

2次元セルオートマトン（ライフゲーム）が誕生

※それでも、挙動が複雑でよくわからない



Still, it is too complex...



- ・次元を落として、1次元セルオートマトンへ

※ようやく、セルオートマトンのふるまいを系統的に
議論できるようになってきた

Wolfram reduced the dimension of C.A., that is, 1D C.A. and
it is easy for us to understand the behavior.

セルオートマトンの周辺

Another stream of C.A.

セルオートマトンの周辺

コネクションマシン (connection machine)
(超並列スーパーコンピュータ) の設計に関与



<http://ja.wikipedia.org/wiki/コネクションマシン>

セルオートマトンのプログラムを実装

本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの数学的定義

Mathematical Aspects of cellular automata

セルオートマトンとは？(1)

Mathematical Aspects of cellular automata (1)

独立変数と従属変数がすべて離散

従属変数の値域が有限集合となっている時間発展系

In C.A., all independent and dependent variables are discrete, and the range of dependent variables are finite sets. C.A. is the time evolution system.

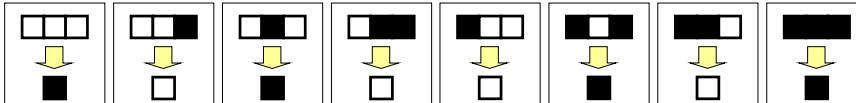
1次元セルオートマトンの一般式

$$a_{t+1}^i = F(a_t^{i-r}, a_t^{i-r+1}, \dots, a_t^i, \dots, a_t^{i+r-1}, a_t^{i+r})$$

The general expression of 1D cellular automata.

セルオートマトンとは？(2)

Mathematical Aspects of cellular automata (2)



$$a_{t+1}^i = F(a_t^{i-1}, a_t^i, a_t^{i+1})$$

tは時刻 (t = time)
iは位置 (i = position)

aは白黒の状態をとることができる
Each cell = white/black = 2 states

定義域側の組み合わせは 8 個
Domain = $2 \times 2 \times 2 = 2^3 = 8$ states

値域側の組み合わせは 2 個
Range = 2 states

ゆえに、構成可能なルールの個数は
 $2^{2^3} = 256$

The number of configurable rules

復習が必要と感じた人には・・・

応用事例とイラストでわかる離散数学 (共立出版)



こちらの第4章 写像
を読んでみましょう

延原 肇 (著)
2,400円

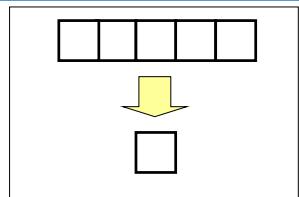
セルオートマトンとは？(3)

Mathematical Aspects of cellular automata (3)

もちろん、もっと複雑化することも可能

※セル数の増加・状態数の増加

$$a_{t+1}^i = F(a_t^{i-2}, a_t^{i-1}, a_t^i, a_t^{i+1}, a_t^{i+2})$$



定義域は2の5乗で、
32通りの組み合わせ
値域側の組み合わせは 2個

構成可能なルールの個数は 4,294,967,296個

ちょっと演習

配布した資料を利用し
1次元セルオートマトンを自分で計算

Exercises: Calculate your own one-dimensional cellular automata by using the distributed paper

くどいようですが、復習が必要と感じた人には・・・

応用事例とイラストでわかる離散数学（共立出版）

延原 肇 (著)

2,400円



セルオートマトンとは？(4)

Mathematical Aspects of cellular automata (4)

2状態、3入力、1出力セルオートマトン

Cellular Automata: 2 cell states, 3 inputs, and 1 output.

$$a_{t+1}^i = F(a_t^{i-1}, a_t^i, a_t^{i+1})$$

構成可能なルールの個数は

The number of configurable rules

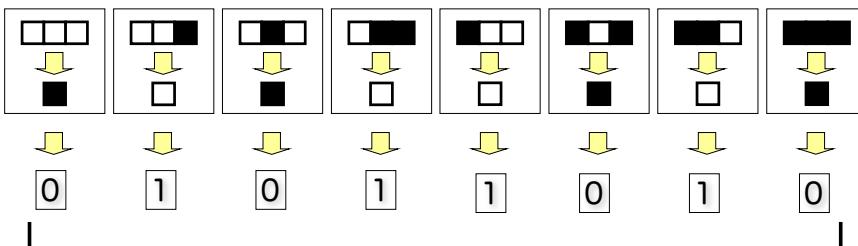
$$2^{2^3} = 256$$



これくらいならば系統的に
調べることができるだろう！
We can investigate the behavior of C.A., systematically

ウォルフラムの調査 (1)

Wolfram research (1)



2進数から10進数に変換
The conversion to decimal from binary

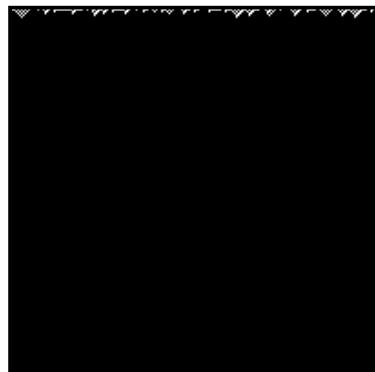
この場合はルール90番目

This case = the rule 90th

クラス1

Class 1

世代を経るうちに、全体がONまたはOFFになる場合

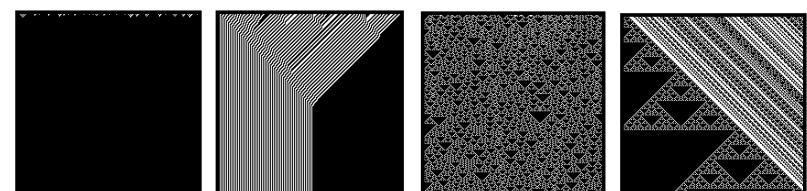


The whole of cells = ON or OFF

ウォルフラムの調査 (2)

Wolfram research (2)

256個のルールを
4つのカテゴリに分類できることを発見
He found 256 rules can be classified into four categories



クラス1
Class 1

クラス2
Class 2

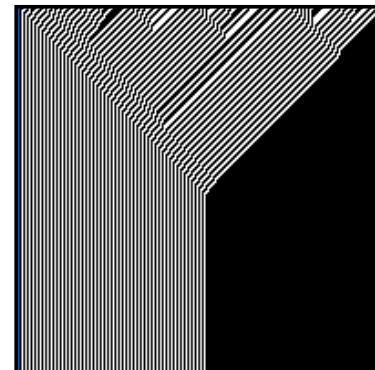
クラス3
Class 3

クラス4
Class 4

クラス2

Class 2

最初はバラバラな動きをするが
ある安定した模様に落ち着く場合

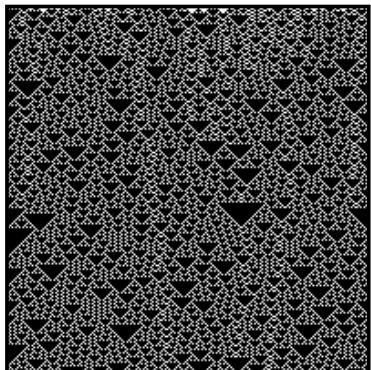


The behavior will be converged to some patterns.

クラス3

Class 3

いつまでもON/OFFがランダムな模様となって出現する場合。（カオス）

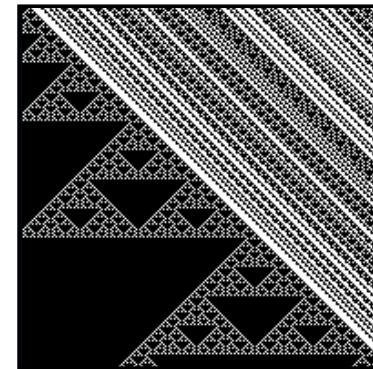


The behavior will be Chaos.

クラス4

Class 4

ランダムな動きをしたかと思えば、何世代か後に
突然意味ありげな模様を生成したりする場合



We recognize some Cosmos in the patterns.

ちょっとビデオ鑑賞

ウォルフラム登場

今回は時間がないので、飛ばします

本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの数学的定義

Mathematical Aspects of cellular automata

セルオートマトンとカオスの縁

cellular automata and chaos

λ (ラムダ) パラメータ

Lambda parameter

クリストファー・ラングトンが提案

This parameter is proposed by Christopher Langton



λパラメータ

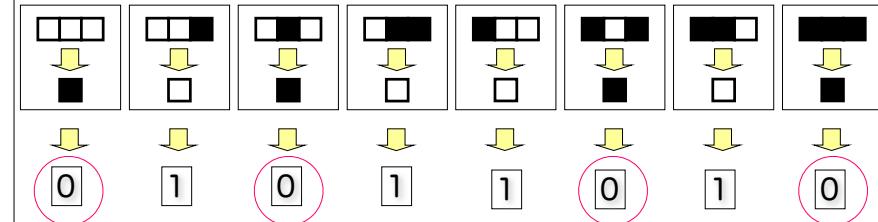
$$\lambda = \frac{K^N - D}{K^N}$$

K: セルのとりうる状態数 The number of cell states

N: 近傍のセル数 The number of neighborhood

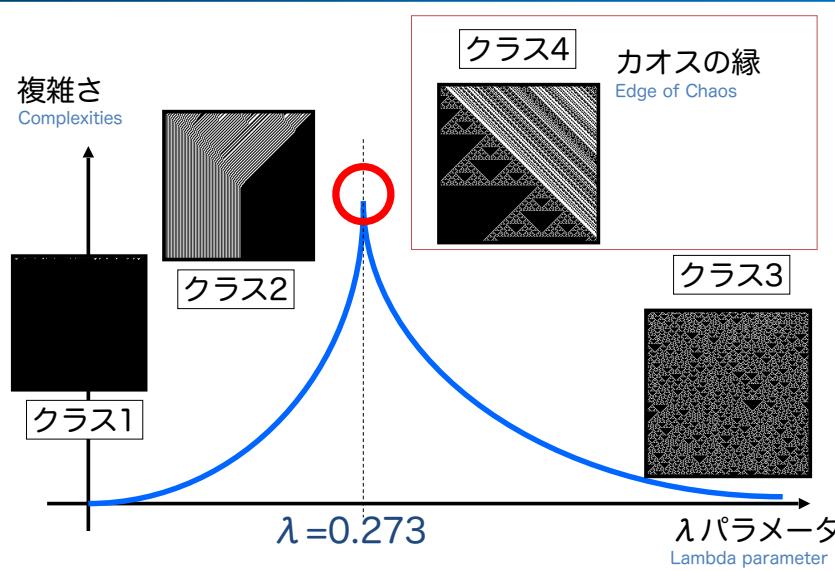
D: 次の状態で0(死)になるルール数
The number of died cells in next step.

λパラメータの計算例

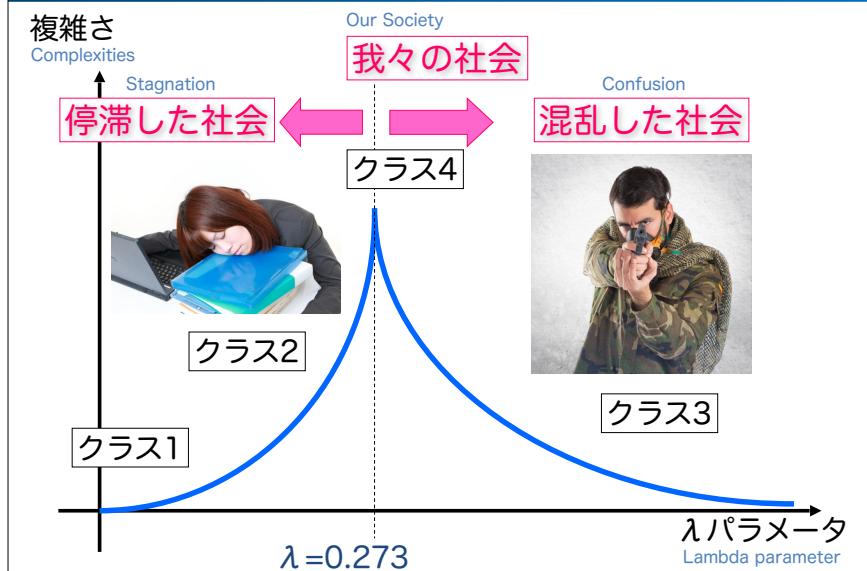


$$\lambda = \frac{2^3 - 4}{2^3} = \frac{8 - 4}{8} = 0.5$$

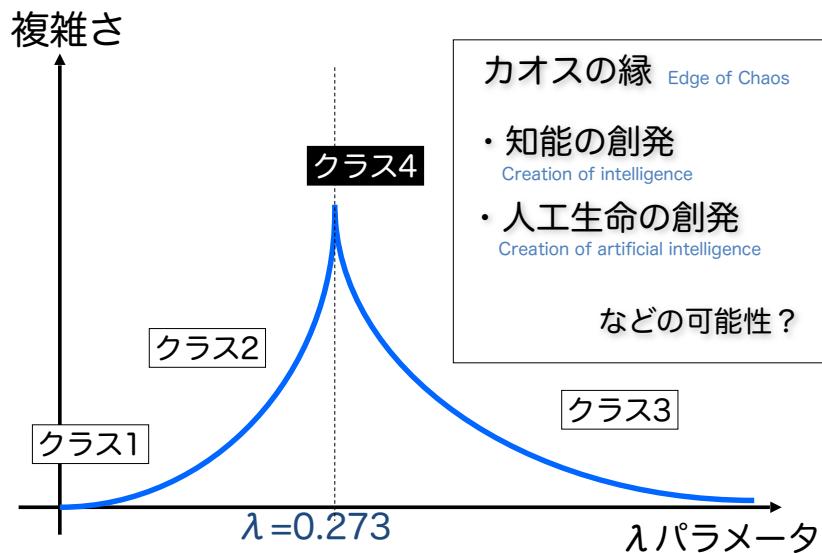
λパラメータとカオスの縁 (1)



λパラメータとカオスの縁 (2)



λパラメータとカオスの縁 (3)



多次元セルオートマトン

Multi-dimensional cellular automata

計算機上に生命を創ることはできるか？

人工生命への挑戦？

Can you create a life on a computer?



2次元セルオートマトン (ライフゲーム)

2D-cellular automata (Life game)

「生命」の定義は？

The definition of life?



本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの数学的定義

Mathematical Aspects of cellular automata

セルオートマトンとカオスの縁

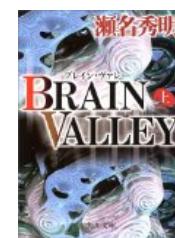
cellular automata and chaos

多次元セルオートマトン

Multi-dimensional cellular automata

人工生命に関する

Artificial Life



「ブレイン・ヴァレー」

著者: 濑名秀明

出版社: 角川書店 (文庫)

発売日: 2000/12



もし皆さんが上図の模様をもつ貝を観測したら、何を思う？

If you observe the shellfish has a pattern in the figure above, then… what do you think?

Wikipedia より引用

人工生命に関連して

Artificial Life



このような模様の形成に関する分野

拡散反応系

Reaction-diffusion system

Wikipedia より引用

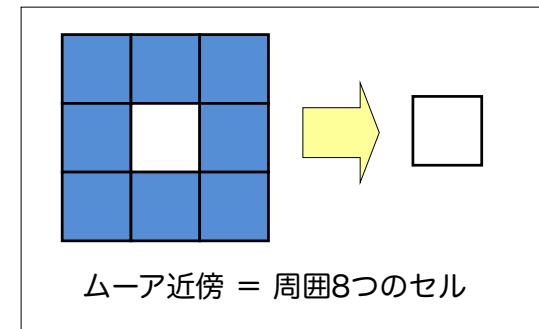
ライフゲーム (1)

ライフゲーム (1)

Life game (1)

1次元セルオートマトン同様、遷移ルールを定義

We need to prepare the state-transition rules like 1D C.A.



$$2^8 = 256$$

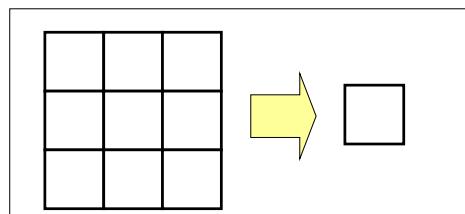
※但し、1次元の場合と異なり、自分自身は定義域に含まない

ライフゲーム (2)

Life game (2)

256個のルールを定義するのは面倒

It is actually quite hard to define the 256 rules.



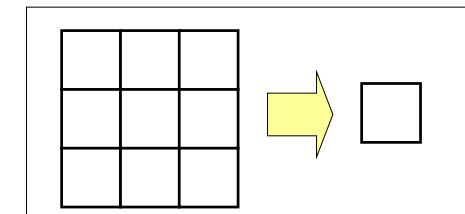
近傍セルの合計数で次の状態を決定

- 1) 注目セルが1の場合、近傍セルのうち、2~3個が1ならば、次も1とする
- 2) 注目セルが1であり、近傍セルのうち、4個以上が1ならば、あるいは1個以下となれば0とする
- 3) 注目セルが0であり、近傍セルのうち、1のものがちょうど3個になれば、1とする

※この場合、自分自身を定義域に含んでいる

How to make life game

It is actually quite hard to define the 256 rules.

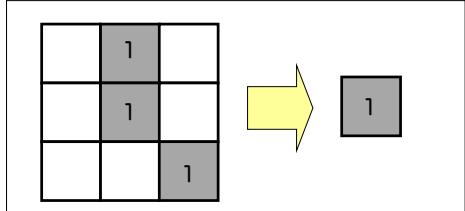


2D C.A. will decide the next state according to the number of neighborhood.

- 1) Center cell = 1 and 2~3 neighbor cells = 1, then next center cell = 1
- 2) Center cell = 1 and [more than 3 neighbor cells = 1 or less than 2 neighbor cells = 1], then next center cell = 0.
- 3) Center cell = 0 and 3 neighbor cells = 1, then next center cell = 1.

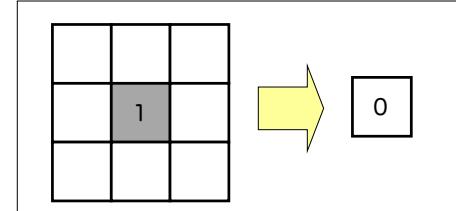
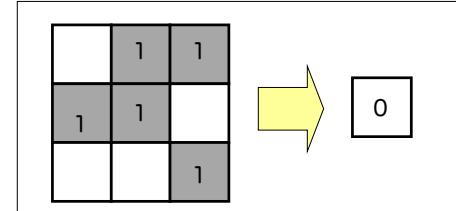
※Here, the center cell should be considered as one of the domain cells.

1) 注目セルが1の場合、近傍セルのうち、2~3個が1ならば、次も1とする



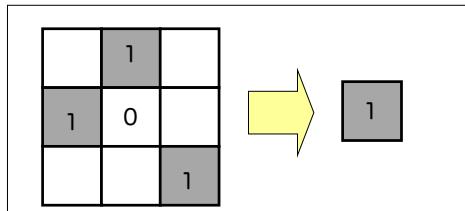
1) Center cell = 1 and 2~3 neighbor cells = 1, then next center cell = 1

2) 注目セルが1であり、近傍セルのうち、4個以上が1ならば、あるいは1個以下となれば0とする



2) Center cell = 1 and [more than 3 neighbor cells = 1 or less than 2 neighbor cells = 1, then next center cell = 0.

3) 注目セルが0であり、近傍セルのうち、1のものがちょうど3個になれば、1とする



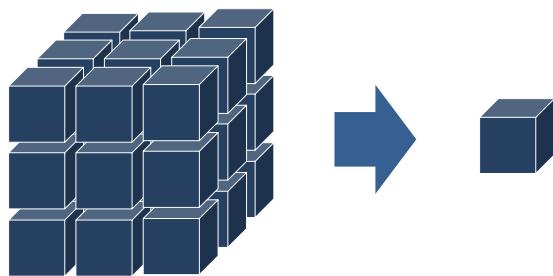
3) Center cell = 0 and 3 neighbor cells = 1, then next center cell = 1.

ちょっとビデオ鑑賞

ライフゲームについて

3次元オートマトンへの拡張

3D cellular automata



周囲26セルの関係から次のステップを
決定する遷移ルールを決めればよい

Q. 学問分野として成立するのか？

じゅうたいがく
渋滞学

traffic jams theory?

A. 我々の予想以上に奥が深い！



本日の構成

オートマトンとは？

What is automata?

3分でわかるセルオートマトン

3-min super-understandable-introduction

セルオートマトンの歴史

History of cellular automata

セルオートマトンの数学的定義

Mathematical Aspects of cellular automata

セルオートマトンとカオスの縁

cellular automata and chaos

多次元セルオートマトン

Multi-dimensional cellular automata

渋滞学との関連

Cellular automata and traffic jams

渋滞とは？ 渋滞学とは？

車の渋滞 congestion



人の行列 queue



動物の群れ group

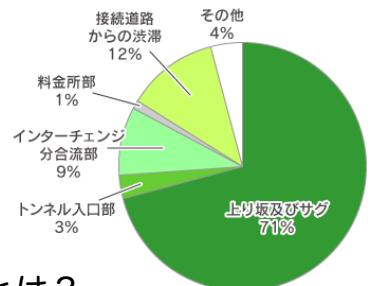
物流 logistics

これらを数理的に一般化し解析する学問領域
「渋滞学」

Disciplines to analyze such the phenomena generally and mathematically.

渋滞その1：高速道路

Example (1) : highway



サグ部とは？

ゆるやかなたるみと、それに伴う坂道



渋滞その2：人の行列

Example (2) : queue

知らない土地へ行って・・・

行列のないラーメン屋と行列のあるラーメン屋
どちらに入る？

You find a ramen shop with a queue and a ramen shop with no queue.
Which you will enter?



行列は、ラーメン屋にとって好都合？

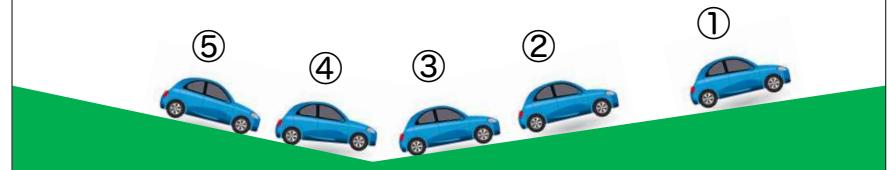
ちょっとビデオ鑑賞

渋滞その1：高速道路

Example (1) : highway

サグ部で渋滞が発生するメカニズム

- ①サグ部では、自分の車の速度の変化に気がつきにくい
- ②すぐ後ろの車が、安全のため軽くブレーキを踏む
- ③さらに、その後ろの車は、安全のため②よりも強くブレーキを踏む
- ④さらに、後の車は強くブレーキを踏む
- ⑤の状態では、完全に車が停止する。



渋滞その2：人の行列

Example (2) : queue

意図的に行列を作る (= 意図的に価値を上げる?)

If you can make a queue intentionally, your shop will be successful?



お店は大繁盛になるか？

- ・都内にあるカフェのパンケーキ (**時間待ち)
(こんなに売れているのに、なぜ小さな店構えのまま?)
- ・ゴールデンウィークの旅行先の温泉・旅館において
自分たちの他に客がいなかつたらどんな気持ちになる?

渋滞その2：人の行列

Example (2) : queue

Webショッピング等で目にする広告 Advertising of Web shopping

- 在庫希少につき早めの購入をお薦めします
- 現在、注文過多のため予約のみ受付中

Accepting only reserved for excessive order

これらは本当だろうか？

Is it real?

ちょっとビデオ鑑賞

回転寿司の映像

渋滞その2：人の行列

Example (2) : queue



数年前から・・・

「お座席番号＊＊列目より後方のお客様・・・」
というアナウンスが流れるようになった

渋滞その3：生物の群れ

Example (3) : group



渋滞その4：在庫の渋滞

Example (4) : Logistics

山のような在庫 = 物流における渋滞

Heavy stock = traffic jams in the logistics



在庫がなくなれば、健全な運営

あるバス停の風景

@bus stop



渋滞学の数理：概要

Mathematical models

高次 Complex

Queuing theory

待ち行列理論

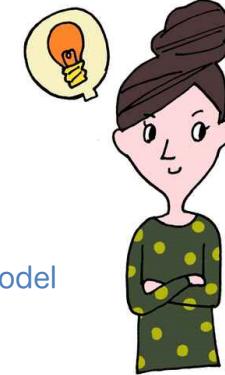
流体モデル

追従モデル

セルオートマトンモデル

Cellular automata model

低次 Simple



あるバス停の風景

@bus stop

疑問

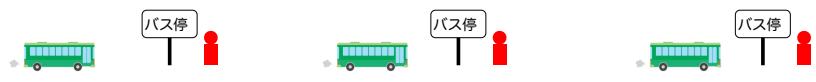
なぜか3台連なってくるバス



1台目はギュウギュウ、2台目は空いている
3台目にいたっては、誰も乗っていない・・・

不思議の解明

- ①3台は均等にスタート



- ②あるバス停で乗る人が急増（例えば学会・試験等が終わった直後）



- ~~③バスの間隔が不均等になりはじめる~~



なぞは解けたか？

なぜか3台連なってくるバス



1台目はギュウギュウ、2台目は空いている

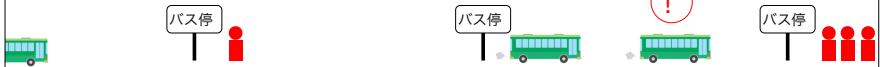
3台目にいたっては、誰も乗っていない・・・

このような状態を解消する方法は？

How to solve this problem?

不思議の解明

- ④間隔が長い分、待つ人が増える



このバス停に
待つ人が増える

- ⑤待つ人が多いバス停がネックになり、バスの間隔がさらに短くなる



- ⑥「！」のバスの直後のバスは、間隔が短いので待つ人も少ない
ゆえに、「！」のバスにどんどん追いつく



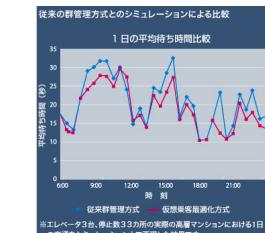
ダンゴ状態回避の方策例



「時間調整のため1分ほど停車します。」
"This train will stop about a minute because of the time adjustment"

山手線の現在の運行状況視覚化サイト

<http://portal.nifty.com/2008/02/01/c/index.htm>



エレベーターの群管理システム

仮想乗客最適化方式

将来発生する乗客を含めて、ビル全体の交通需要を予測。
平均待ち時間を最大10%※短縮できます。

本日の構成

オートマトンとは？

[What is automata?](#)

3分でわかるセルオートマトン

[3-min super-understandable-introduction](#)

セルオートマトンの歴史

[History of cellular automata](#)

セルオートマトンの数学的定義

[Mathematical Aspects of cellular automata](#)

セルオートマトンとカオスの縁

[cellular automata and chaos](#)

多次元セルオートマトン

[Multi-dimensional cellular automata](#)

渋滞学との関連

[Cellular automata and traffic jams](#)

今週の演習について

[About this weekly report](#)

演習 (exercises)

演習1. 1次元セルオートマトンの実装 (2点)

演習2. 1次元セルオートマトン（インタラクティブ）の実装 (2点)

演習3. 2次元セルオートマトンの実装 (3点)

演習4. セルオートマトンのハードウェア実装への挑戦 (5点)

選択課題 (どれも5点)

選択1. 拡散反応系の実装 (5点)

選択2. 高速道路シミュレータの実装 (5点)

選択3. バスシミュレータの実装 (5点)

演習4. セルオートマトンのハードウェア実装への挑戦 (5点)

授業中に紹介した歯車計算機のようなイメージで、例えば、セルオートマトンをハードウェアとして実装しようとしたら、皆さんは、どのようなハードウェアを設計しますか？自由に発想し、それをPDFファイルにして提出すること。（他のメンバとの協力はせずに、必ず1人でアイデアを出してください）。また、ハードウェアとしての実装が難しい場合には、どこが最大の阻害要因となっているのかを説明すること。

交通流シミュレータ (Nagare-san)

To implement the highway simulation



交通流シミュレータ: 「ながれさん」

Rule 184

アクセル



スペースあり



Case 1

ブレーキ



スペースなし



Case 2

交通流シミュレータ (Nagare-san)

To implement the highway simulation



交通流シミュレータ: 「ながれさん」

ルール184をベースにしているが
なめらかに動くように改良している

ルール184

状態t=0



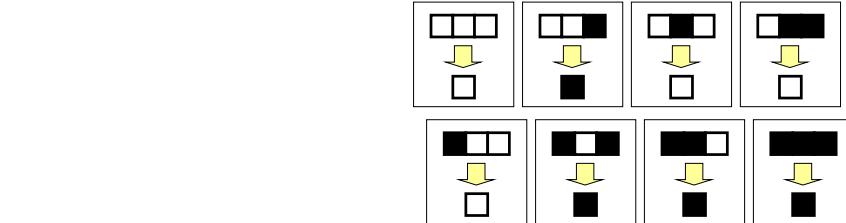
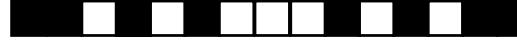
■ = 0

□ = 1

状態t=1

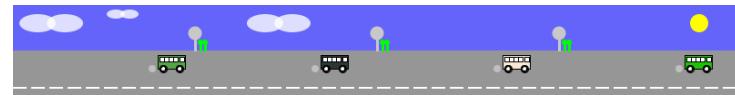


状態t=2



バス停シミュレータ (Ittetsu)

To implement the bus simulation



Nagare-san 同様 ルール184をベース