

# Nonlinear computing with switching map systems <sup>1</sup>

東京大学 広域科学専攻 佐藤 譲 <sup>2</sup>

Dynamical systems based computational theory is studied. We demonstrate the ability of simple switching map systems to solve non-trivial computational problems. There exists a switching map system with two types of baker's map to emulate any Turing machines. Here the baker's maps are corresponding to elementary process for computing ('effective procedure') such as left/right shift and read/write symbols in the view of Turing machines. Taking other nonlinear mappings as elementary process, the dynamical system can solve various problems in different ways than classical Turing machines. We call this kind of computing 'non-linear computing' and analyze its computational power on a point of view of dynamical systems.

## 1 Baker's map networks と Turing machines

近年古典的計算モデルである Turing machine と二次元区分線形写像 (Generalized shifts) とが等価であることが C. Moore により示されたが [5], この発展としてここでは切り替え写像系 (Switching Map System) を用いた構成的な計算モデルを考える. 切り替え写像系とは以下のようなものである [9].  $(f_1, \dots, f_M)$  を  $x \in \mathbf{R}^n$  の  $M$  個の写像の組とする.

$$x \rightarrow f_i(x) \rightarrow \dots \rightarrow f_i^{n_1}(x) \rightarrow f_j(f_i^{n_1}(x)) \rightarrow \dots \rightarrow f_j^{n_2}(f_i^{n_1}(x)) \rightarrow \dots$$

のように写像  $f_i$  を切替えながら初期値に適用していく. 新たに選択されたダイナミクスの初期状態は先行したダイナミクスの終状態で与える. とくにある時点で適用している写像  $f_i$ , その時点での  $x$  の属する区間  $X_I$  に対して

$$F_n(x) = (g(n), f_i(x)) \quad x \in X_I \quad (n = 1 \dots N, i = 1 \dots M)$$

の形で切替えのタイミングが全て与えられている切替え写像系を決定性切替え写像系とよぶ. ここで  $N$  は  $F$  の総数,  $g(n)$  は写像  $F_{g(n)}$  へのラベルである.

ここでは力学系のルールをプログラム, 軌道を計算の過程ととらえる. 軌道の構造は計算過程の構造であり, プログラムの機能に対応する. この枠組により実空間上の計算のダイナミクスを直接構成的に扱うことが可能となり, 測度や次元, アトラクタの位相構造といった力学系の特性量を計算論的に解釈することができるようになる.

<sup>1</sup>京大基研研究会「複雑系 6」

<sup>2</sup>E-mail: ysato@sacral.c.u-tokyo.ac.jp

とくに以下の2つの baker's map の組み  $f_i = B_i (i = 0, 1)$ , および  $x = \frac{1}{2}$  での相空間の分割による切り替え  $g(n)$  を選ぶと, この切り替え写像系  $S(B_0, B_1)$  は Turing Machine と等価な記号操作系とみなせる.<sup>3</sup>

$$B_0(x, y) = \begin{cases} (\frac{x}{2}, 2y) & y \in [0, \frac{1}{2}) \\ (\frac{x+1}{2}, 2y-1) & y \in [\frac{1}{2}, 1) \end{cases}$$

$$B_1(x, y) = \begin{cases} (2x, \frac{y+1}{2}) & x \in [0, \frac{1}{2}) \\ (2x-1, \frac{y}{2}) & x \in [\frac{1}{2}, 1) \end{cases}$$

$$g(n) = \begin{cases} n_0 & x \in [0, \frac{1}{2}) \\ n_1 & x \in [\frac{1}{2}, 1) \end{cases} \quad (n_0, n_1 \text{ は } 1 \dots N \text{ のうちいずれか})$$

ここで系の構成要素を区分別形写像だけに限らず, 例えば Henon map のような連続な実写像を用いることを許せば, この拡張された切り替え写像系は, 実数性によって自明に古典的 Turing Machine の能力を上回る計算能力を持つことになる [8]. この場合は実数計算論 (例えば [1][7]) に則して議論するのが妥当である. また入出力を有理数, 写像を有理関数に限った場合を考えると, 2数の積を2数の和と同じ手間で計算できることなどより, この力学系は NP 完全問題を多項式時間で解くことができることがわかる. 実際にはこのとき適用される Henon map の操作を  $n$ -bit の精度で実現する<sup>4</sup> のに必要な baker's map の数が  $n$  に対して指数的に増大することが予想される [2]. また静的な出力だけでなく, 系の周期解やカオス解等も計算結果と見なすことにより停止問題 (halting problem) の概念が動的構造を含むものに拡張される [3].

## 2 Examples

以下に Chomsky 階層に対応した言語の受理系を  $S(B_0, B_1)$  で構成した例を与える.

### Parity checker (RG):

Parity check は入力列に現れる 1 の数の偶奇性の判定をする問題である. この言語は正規言語 (regular language) であり図 1 左のような FSA (finite state automaton) で受理できる. 一般に有限オートマトンは右方向にのみ動く Turing machine と等価なので, 切り替え写像系  $S(B_0^{-1}, B_1)$  で構成できる Parity checker は  $x$  座標の 2 進展開中に 1 が偶数個現れると (0,0) に, 奇数個現れると (0,1) の不安定固定点に漸近する. この力学系の単位時間あたりの右シフト数は  $N_{RG} = 1$ , 平均 Lyapunov 指数は  $\lambda_{RG} = \log 2$  である.

### Dyck language recognizer (CFG):

Dyck language は入力列に現れる '(' と ')' の整合性を判定する問題である. この言語は文脈自由言語 (context free language) であり図 1 右のような PDA (pushdown automaton) で受理できる. PDA は初期テープの左側無限列に記憶領域 (スタック) をとり, 入力列を右側無限列にとれば切り替え写像系  $S(B_0, B_1)$  で構成できる. Dyck language recognizer は  $x$  座標の 2 進展開中の 0 と 1 が

<sup>3</sup> Appendix 参照.

<sup>4</sup> 有理パラメーターの Henon map は計算可能な関数なので  $S(B_0, B_1)$  で表現できる.

上の意味で整合性がとれているとき  $(0,0)$  に、そうでないとき  $(0,1)$  の不安定固定点に漸近する。スタック部分の操作は右シフトと左シフトが互いに打ち消し合うので、入力長さを  $n$ 、入力列中の 0 の数を  $m$  とおくとシフト数は  $N_{CFG} \sim \frac{m}{n}$ 、平均 Lyapunov 指数は  $\frac{m}{n} \log 2 \leq \lambda_{CFG} < \log 2 = \lambda_{RG}$  となる。

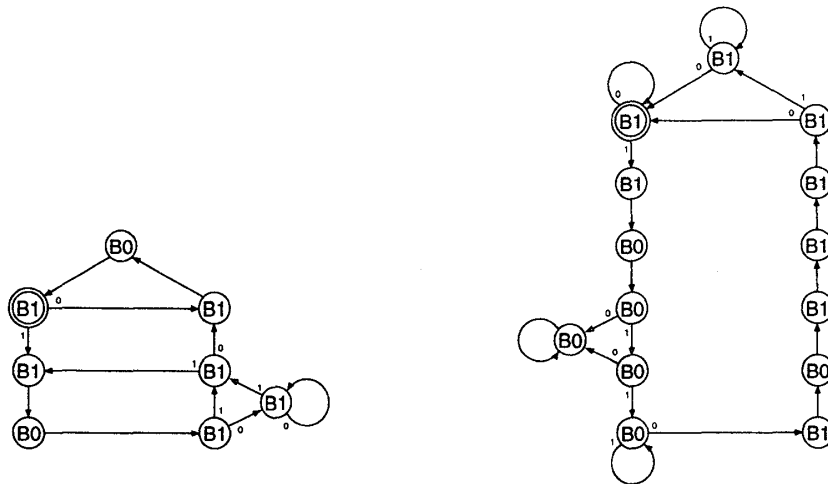


図 1: Parity checker (左図) と Dyck language recognizer (右図). 初期状態は二重円, 切り替え判定は枝に 0,1 で与えてある。

### Primarity checker (CSG):

Primarity check は入力列に現れる 1 の数の素数性の判定をする問題である。この言語は文脈依存言語 (regular language) であり図 3 のような LBA (linear bounded automaton) で受理できる。LBA は初期テープの左側に左端記号をおき、入力列+右端記号を右側無限列にとれば切り替え写像系  $S(B_0, B_1)$  で構成できる。  $x$  座標の 2 進展開が  $110^{2i}11$  で与えられる初期値に対して、Primarity checker は  $i$  が素数の場合  $(0,0)$  に、合成数の場合は  $4p+2$  周期軌道に収束する (図 2)。ここで  $p$  は  $i$  を割りきる最小の素因数である。

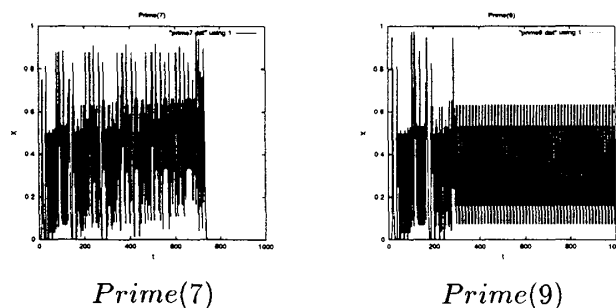


図 2: Primarity checker の計算プロセス: 7 を入力した場合 (左図,  $(0,0)$  に収束して受理状態に入る) と 9 を入力した場合 (右図, 周期 14 の不安定周期解)。素数は無限に存在するのでこの力学系は可算無限個の不安定周期解をもち、また非可算無限個の不当な入力に対して非周期解をもつ。

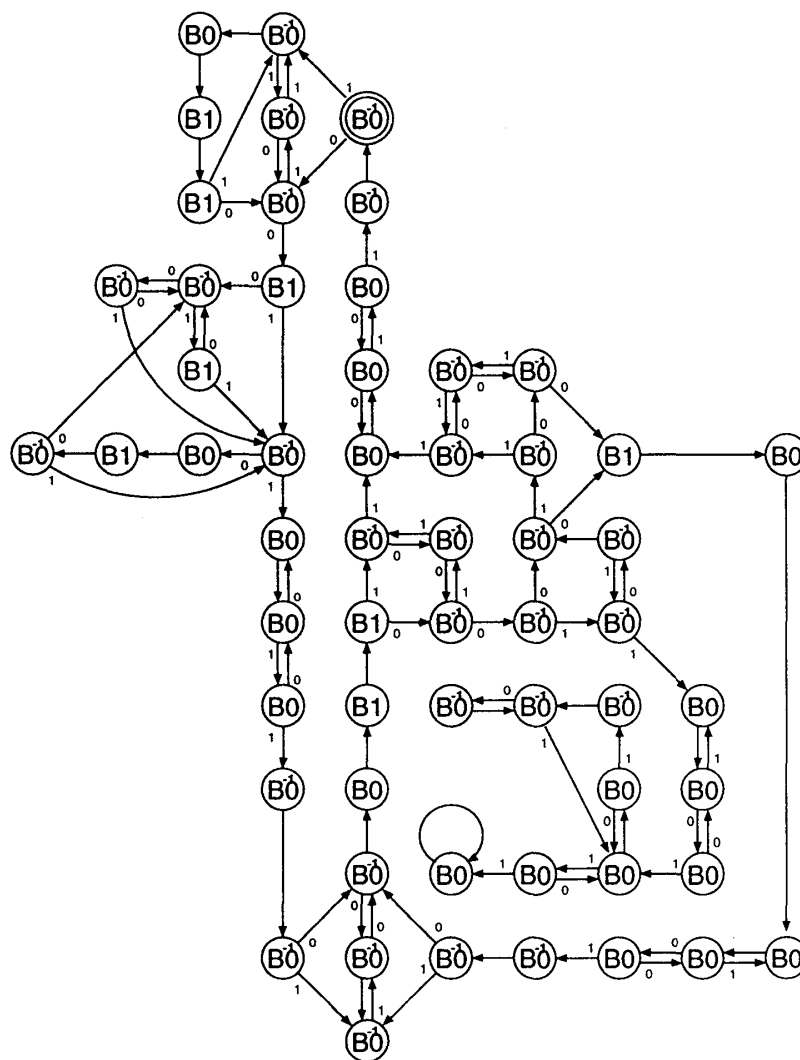


図 3: Primarity checker

**Minsky's universal Turing machine (TM):**

万能 Turing machine も切り替え写像系  $S(B_0, B_1)$  で構成できる. 図 4は M. Minsky によって与えられたサイズの小さい万能 Turing 機械である. この力学系は計算可能な任意の 2 次元離散力学系の時系列を模倣できる. 各々の切り替え写像系の構造は初期値の一部に埋め込まれることになる.

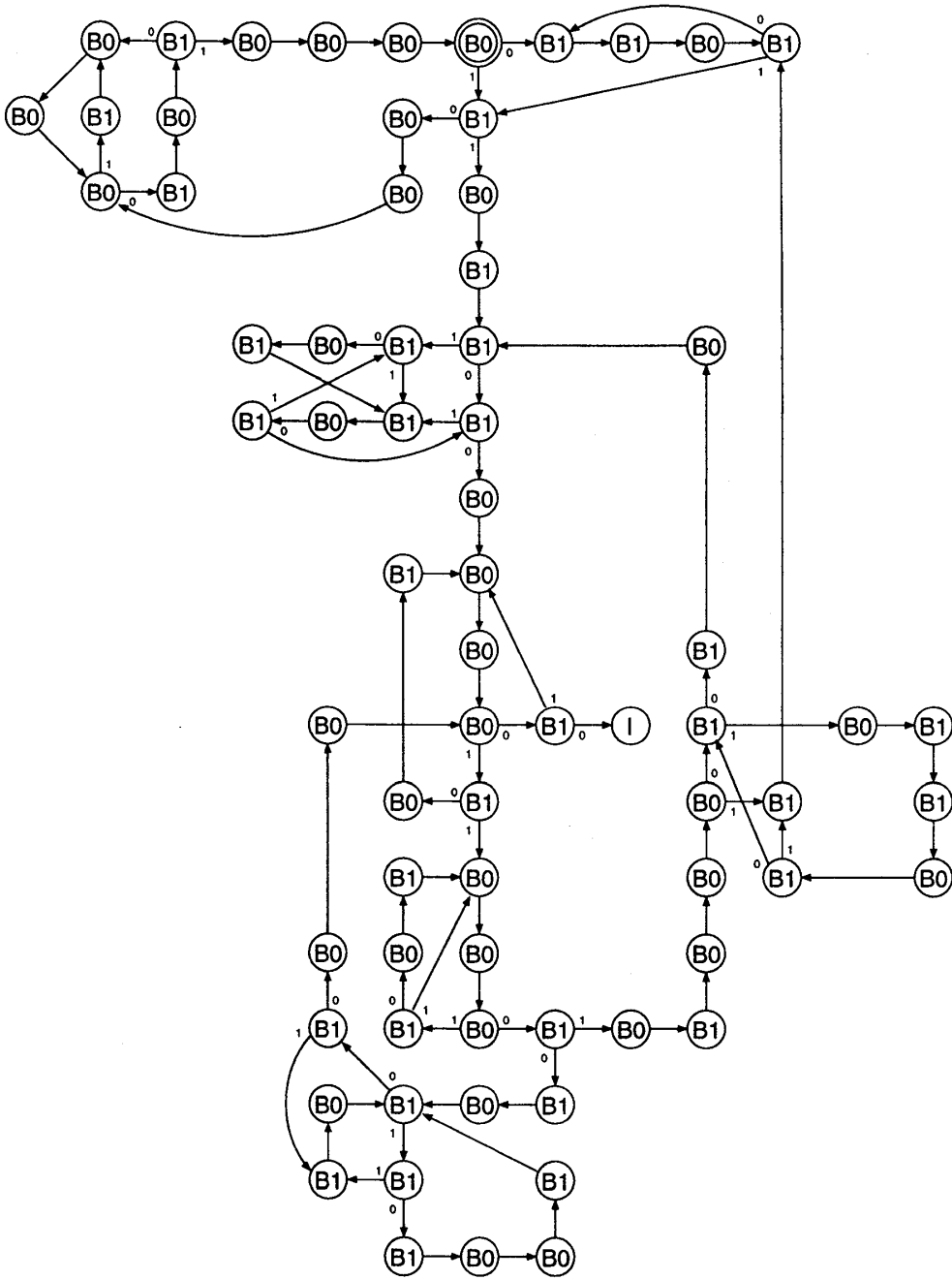


图 4: Minsky's UTM

### 3 Discussion : Non-linear computing

今後の課題と非線型計算論の可能性について議論する.

#### Non-linear procedure:

計算の素過程を baker's map 以外の写像にすると何が起こるだろうか? 2 節で扱った Parity checker に 2 次の非線型性を持つ写像を組み込んでみると baker's map を用いた場合と異なり, 計算過程において軌道の混合が見られ machine の内部状態の分離が困難になる (図 3). またプログラムの構造にも歪みが生じ文法の再構成が困難になっていることがわかる [6].

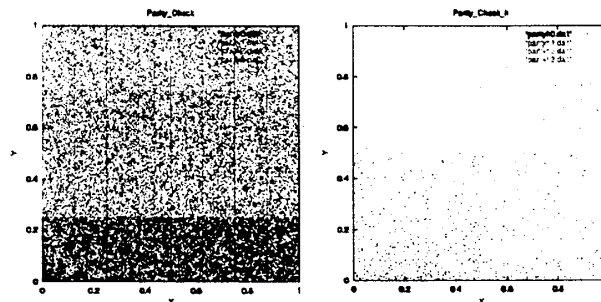


図 5: Primarity checker の計算プロセス: Parity checker の内部状態 (左図) と Henon map を書き換え操作に組み込んだ non-linear parity checker の軌道内部状態 (右図). 4 つの内部状態を色分けして示している. Non-linear parity checker には軌道の混合がみられる.

#### Non-determinicity:

初期入力 (TM 描像ではテープの初期 bit 配置) を誤差を含む  $\epsilon$  球で与えると系の発展は領域判定に伴う分割により非決定的になり軌道は非一様に分岐していく. このような計算過程は非決定性 Turing machine と対比でき, 動的計算階層の構築にも役立つ.

#### Parallel interference computation:

この系では並列計算は要素である写像の高次元化により実現される. とくにプロセス同士が干渉しあうような並列干渉計算のモデルを構成し, このような系の組織化 (学習) についての考察を将来的に試みたい.

#### Computational flow:

Generalized shift のような区分線形な離散力学系は懸垂によって高次元の滑らかな連続力学系に埋め込むことができる [5]. 発展として例えば 2 節で得た UTM を Poincare section に持つような flow を構成し, その力学系的性質を解析する予定である.<sup>5</sup>

非線型計算論の枠組みを考察したが, 現時点で有意義な結果が得られたとはいえない. 今後の展望としては軌道の混合や並列計算プロセスの干渉といった性質を積極的に生かした情報処理のメカニズムを検討し, 学習による組織化を念頭において, non-trivial な計算能力をもつ高次元の flow を構成, 解析する予定である.

<sup>5</sup>[5] ではそのような flow の存在が示されているが, 具体的に構成した例は無い.

## Appendix: $S(B_0, B_1)$ と Turing machine との対応

2 記号 Turing machine のテープを両側無限の 2 進記号列とみなし、ヘッドより右側の記号列を  $x$  座標の 2 進展開, 左側の記号列を  $y$  座標の 2 進展開とみてテープを点  $(x, y) \in [0, 1) \times [0, 1)$  に対応させる. このとき  $B_0$  と  $B_1$  はそれぞれ machine の左シフト, および右シフト+記号書き換えの操作に相当する (図 6, 7). 右シフトは  $B_0^{-1} = B_1 \circ B_0 \circ B_1$ , 左シフト+記号書き換えは  $B_1^{-1} = B_0 \circ B_0 \circ B_1$  で表され,  $x = \frac{1}{2}$  での相空間の分割による領域判定はヘッド下の記号  $\{0, 1\}$  の読み取りに対応する. 以上に halting 動作に相当する恒等写像  $I$  を含めて構成された切り替え写像系は Turing machine の記号操作を実行できる.

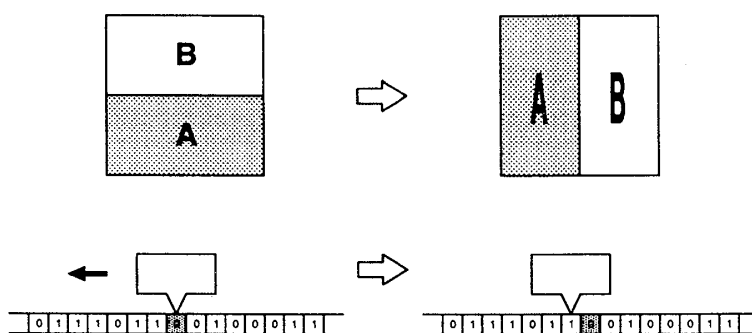


図 6:  $B_0$  と TM の左シフト

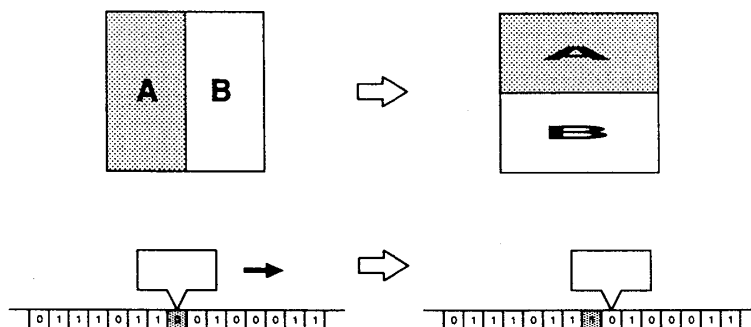


図 7:  $B_1$  と TM の右シフト+記号書き換え

## 謝辞

池上 高志 (東大教養), 泰地 真弘人 (統数研), 梅野 健 (CRL), 橋本 敬 (JAIST), 山口 明宏 (北大理), 山本 知幸 (北大理), 後藤 謙太郎 (北大理), 斎藤 朝輝 (RIKEN) 氏に感謝します.

## 参考文献

- [1] L. Blum, M. Shub, S. Smale, "On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines," *Bull. Amer. Math. Soc.*, **21**, pp. 1-46 (1990).
- [2] J. P. Crutchfield, K. Young, "Computation at the onset of chaos," *Complexity, Entropy and the Physics of information*, Addison Wesley, pp. 223-269 (1990).
- [3] P. Grim, "Philosophical computers," MIT press (1999).
- [4] M. Minsky, "Computation: Finite and Infinite Machines," Prentice-Hall (1967).
- [5] C. Moore, "Generalized shifts: Unpredictability and undecidability in dynamical systems," *Nonlinearity*, **4**, pp. 199-230 (1991).
- [6] J. B. Pollack, "The induction of dynamical recognizers," *Machine Learning*, **7**, pp. 227-252 (1991).
- [7] M. B. Pour-El, J. I. Richards. "Computability in Analysis and Physics," Springer-Verlag Berlin (1989).
- [8] H. T. Siegelmann, Eduardo D. Sontag, "Analog computation via neural networks," *Theoretical Computer Science*, **131**, pp. 331-360 (1994).
- [9] I. Tsuda, "Chaotic itinerancy as a dynamical basis of hermeneutics in brain and mind," *World Futures*, **32**, pp. 167 (1991).
- [10] A. M. Turing, "On computable numbers, with an application to the enentscheidungsproblem," *Proc. London Math. Soc.* **2-42**, pp. 230-265 (1936).