

# 形式言語を認識するセルオートマトンの漸次学習

河島 輝<sup>†</sup> 中村 克彦<sup>‡</sup>

<sup>†</sup>東京電機大学大学院理工学研究科 <sup>‡</sup>東京電機大学理工学部

## 1 まえがき

セルオートマトン (CA : Cellular Automata) は多数のセルと呼ばれる同一の有限状態機械を規則的に配置し相互接続したシステムである。あるセルの状態はそのセルと相互接続している近傍セルの状態によって決定し、すべてのセルは状態が同期して推移する。セルオートマトンは並列システムの理論モデルとして扱われる。一般に人間の言語認識も実時間であると考えられるので、CA での実時間言語認識は重要である。1次元1方向性セルオートマトン (OCA : One-way Cellular Automata) は各セルの左隣の1セルを近傍セルとするセルオートマトンである。

われわれはこれまで文脈自由言語 (CFG : Context Free Grammar) の文法推論のための Synapse システムを開発してきた [1]。OCA の局所関数はチョムスキー標準形の文脈自由文法の規則と類似の規則で表せるため、OCA の学習に CFG の学習の方式を応用できる。本研究の目的は形式言語の正例と負例の集合から OCA の規則集合を学習するシステムを構築することである。

## 2 1次元1方向性セルオートマトン

OCA は  $S = (K, \#, f, A)$  で表される。  $K$  は状態の有限集合、  $\# \in K$  は境界状態、  $f : K \times K \rightarrow K - \{\#\}$  は局所関数、  $A \subseteq K$  は受理状態の集合である。

OCA の概要を図1に示す。OCA はあるセルとその左隣のセルの状態によって次の状態が決定する。

われわれは局所関数  $f$  を規則  $lc \rightarrow f(l, c)$ ,  $l, c \in (K - \{\#\})^+$  の集合によって表す。状態 (configuration) は記号列  $\#w, w \in (K - \{\#\})^+$  で表される。状態の推移を表す全域関数  $F$  は次のように定義される。

$$F(\#a_1 \cdots a_n) = \#b_1 b_2 \cdots b_n$$

ただし  $b_i = f(a_{i-1}, a_i)$ ,  $a_0 = \#$  である。さらに、 $F$

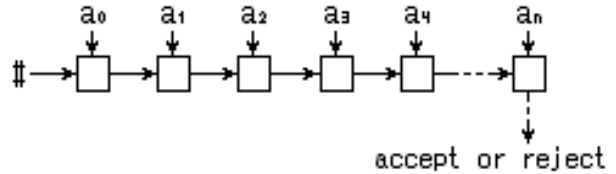


図1: 1次元1方向性セルオートマトンの概要

の拡張  $F^n (n \geq 0)$  が  $F^0(c) = c$ ,  $F^n(c) = F(F^{n-1}(c))$  によって定義できる。

あるOCA  $S$  が言語  $L \subseteq \Sigma^*$  を時間  $t$  で受理するとは、すべての  $w \in \Sigma^*, |w| \geq 2$  に対して、右側のセル画  $t$  ステップ後に受理状態になることである。

$$w \in L \Leftrightarrow F^t(\#w) \in \{\#\}K^+A.$$

$t = |w| - 1$  のとき  $S$  は  $L$  を実時間認識、また定数  $c \geq 1$  に対して  $t \leq c \cdot (|w| - 1)$  のとき、線形時間認識できるという。

ある記号列  $w \in \Sigma^*, |w| - 1 = t, P \in K$  のとき、 $F^t(\#w) = \#K^+P$  ならば  $w \Rightarrow P$  と書く。これを状態の推移系列 (Space-Time transition diagram) と呼ぶ。

OCA で受理できる言語を OCAL (One-way Cellular Automata Language) と呼ぶ。OCAL とほかの言語のクラスは以下のような関係がある。

1. OCAL のクラスは文脈自由文法のクラスを含んでいる
2. 実時間認識可能な OCAL のクラスは文脈自由でない言語を含んでいる。例えば  $\{a^n b^n c^n \mid n \geq 1\}$  である [2, 3]。

## 3 OCA の学習システム

OCA を学習するために本システムでは状態の推移系列を用いて受理、規則の生成を行っている。受理は各正例  $w$  に対する状態の推移系列を調べ、 $w \Rightarrow P, P \in A$  ならば受理が決定できる。受理できなかった場合、状態の推移系列を用いて正例を探索し必要な規則を探索する。

各セルの状態は次のように与える [4]。

Incremental Learning of Cellular Automata Recognizing Formal Languages  
 Kawashima AKIRA<sup>†</sup>, Katsuhiko NAKAMURA<sup>‡</sup>  
<sup>†‡</sup>Tokyo Denki University, School of Science and Engineering  
 350-0394 Ishizaka Hatoyama-cho Hiki-gun Saitama-ken Japan  
<sup>†</sup>highjumn@naklab.k.dendai.ac.jp <sup>‡</sup>nakamura@k.dendai.ac.jp

1. 状態は  $\{1, 2, \dots, M\}$  の部分集合  $K$  とする.  $|K| \geq 2$  のとき多重状態と呼び,  $|k| = 1$  のとき, 単一状態と呼ぶ.
2.  $\{2, 3, \dots, m\}$  を入力記号の集合とする.
3. 1 を受理状態とする.

OCA を  $S = (K, \#, f, A)$  に対して, 入力記号列を  $w \in \Sigma^+, \Sigma \subseteq K$ , 文字列の最後の一文字を取り除いた部分記号列を  $w_L$ , 最初の一文字を取り除いた部分記号列を  $w_R$ ,  $P, Q, R \in K$ , 状態の推移系列は  $w \Rightarrow P$  とする.

### 3.1 OCA による記号列の受理

本来 OCA は入力によって与えた初期状態を推移させて受理状態になるかによって受理を判定する. しかし本システムではトップダウン構文解析のように受理状態から推移規則を逆に使って状態の推移系列を調べている.

トップダウン方式で受理状態から状態を分岐させていくと同時に, 入力記号列  $w$  も  $w_L$  と  $w_R$  に分岐させ, 状態と対応させながら再帰的に降りていく.

1.  $w \Rightarrow P$  であれば終了である.
2. 規則集合に  $QR \rightarrow P$  が含まれている時,  $w_L \Rightarrow Q$ ,  $w_R \Rightarrow R$  でならば,  $w \Rightarrow P$  である.
3. 規則集合に  $QR \rightarrow P$  が含まれている時,  $w_L$  から  $Q$ ,  $w_R$  から  $R$  へそれぞれ推移できるかを再帰的に調べる. それぞれ  $w_L \Rightarrow Q$ ,  $w_R \Rightarrow R$  であった時,  $w \Rightarrow P$  である.

### 3.2 局所関数の導出

推移規則の導出もトップダウン的に動作し, 受理状態から記号列へ再帰的に降りていきながら, 足りない規則を生成する. 以下の規則を非決定的に実行し規則の生成を行なう.

1.  $w \Rightarrow P$  であれば終了である.
2. 規則集合に  $QR \rightarrow P$  が含まれている時,  $Q$  から  $w_L$ ,  $R$  から  $w_R$  をそれぞれ再帰的に規則の生成を行う. 規則の生成に成功したら  $w \Rightarrow P$  である.
3. 規則集合に  $QR \rightarrow P$  を新しく加え,  $Q$  から  $w_L$ ,  $R$  から  $w_R$  をそれぞれ再帰的に局所関数の導出を行う. 規則の生成に成功したら  $w \Rightarrow P$  である.

## 4 実行結果

システムの実験結果を表 1 に示す. ここで GR は OCA の学習が完了するまでに生成した規則の数, 時間の単位は秒である. (a) と (b) の言語は文脈自由言語,

表 1: 形式言語を受理する OCA の学習結果

言語	規則数	状態数	GR	時間
(a) $\{a^n b^n \mid n \geq 1\}$	5	4	7	0.14
(b) 括弧言語	13	4	66	0.53
(c) $\{a^n b^n c^n \mid n \geq 1\}$	14	6	6663	90

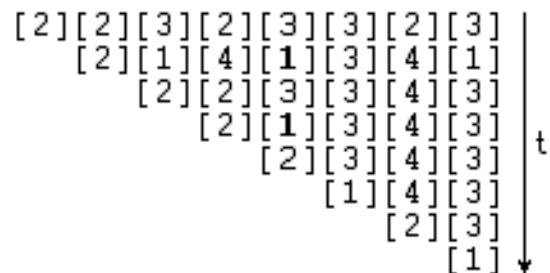


図 2: 学習した規則による括弧言語の推移図 (入力は aababbab)

(c) の言語は文脈自由でない言語である. (c) の言語が (a) と (b) に比べて時間, GR とともに大きくなっており, 状態数が大きいと規則の探索に多くのコストがかかることが分かる. 学習した推移関数を用いた (b) の言語の推移図を図 2 に示す. 1 に推移しているので受理に成功していることがわかる.

## 5 むすび

OCA の推移関数の導出法について述べた. まだ導出できている言語が少ない状態なのでもっと多くの言語を導出できるようにしていきたい.

## 参考文献

- [1] K. Nakamura, Incremental learning of context free grammars by bridging rule generation and search for semi-optimum rule sets. *LNCS*, 4201:72, 2006.
- [2] C. Dyer, One-way bounded cellular automata, *Inform. and Control* 44 54-69, 1980.
- [3] A.R. Smith III, Real-time language recognition by one-dimensional cellular automata, *Jour. Comput. and System Sci.* 6 233-253, 1972.
- [4] K. Nakamura and K. Imada, Incremental Learning of Cellular Automata for Parallel Recognition of Formal Languages, *LNCS* 6332 Springer-Verlag 117-131, 2010.