■6 群（コンピュータ‐基礎理論とハードウェア）－ 計算論とオートマトン

# 1 章 オートマトンと言語

（執筆者：岩本宙造）[2010 年 3 月 受領]

**■概要■**

本章では，オートマトンと形式言語・形式文法について概覧する．まず，有限オートマトンとは何かについて，自動販売機を例にとって説明する．次に，有限オートマトンを，文字列の書かれたテープ，文字を読み取るヘッド，状態を保持する制御部からなる機械として定義する．更に，有限オートマトンに，テープ上の文字列の書き換えや，ヘッドの動く方向，ヘッド数，補助記憶装置などを付加することで，多様なオートマトンが定義できることを述べる．また，それらのオートマトンの能力の関係を紹介する．

一方，自然言語の文法を定性的に研究するための数学モデルとして形式言語の理論がある．まず，形式文法とは何かについて概説し，Chomsky による句構造文法の分類である 0〜3 型文法や，2 型文法をベースにした様々な変種の文法を紹介する．その後，句構造文法以外の独自の研究動機をもつ形式文法の幾つかを紹介する．最後に，語を生成する文法と，語を認識するオートマトンの関係を述べる．

**【本章の構成】**

本章は，オートマトン（1-1 節），形式言語と文法（1-2 節）の 2 節からなる．

■計算論とオートマトン－1 章

# 1-1 オートマトン

（執筆者：守屋悦朗）［2009 年 1 月 受領］

オートマトンは,「自動的に動くもの」を意味するギリシャ語の automaton を語源とし，英語やドイツ語でもロボットや自動販売機の類の「自動機械」を意味する．自動販売機は外部からの刺激（硬貨の投入）によって「状態」（それまでに何円投入されたか）が変化する．初期状態（投入金額 0 円）から始めて，硬貨の投入に従って次々に状態を変え，目的の状態（商品の販売価）に到達すると商品を放出する．オートマトンへの入力である硬貨の種類は有限個に限られており，また，途中で取りうる状態の種類も有限個である．図 1･1 は，投入された入力に従って状態が遷移する可能性を→で表し，初期状態を start →○ で，目的が達成される状態（受理状態と呼ぶ）を◎で表している．このような図を遷移図と呼び，一般に，このような遷移図で表される系を有限オートマトン（FA）という．

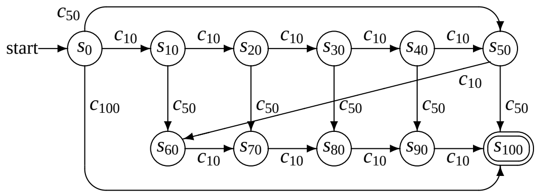



**図 1･1　自動販売機を表す有限オートマトンの遷移図**

図 1･1 の FA において，投入硬貨 $x$ 円を記号 $c_x$ で表し，投入された金額の合計（オートマトンの内部状態）$y$ 円を記号 $s_y$ で表し，この FA に名前 $A$ を付けて $A = (S, \Sigma, \delta, s_{int}, F)$ と表す．$S = \{s_0, s_{10}, \cdots, s_{90}, s_{100}\}$ は取りうる状態の集合，$\Sigma = \{c_{10}, c_{50}, c_{100}\}$ は可能な入力の集合，$F = \{s_{100}\}$ は受理状態の集合（この例では受理状態は一つ)，$\delta$ は現在の状態 $s$ とそのとき入力される記号 $c$ に対しオートマトンが次に取る状態 $\delta(s, c)$ を与える遷移関数 $\delta : Q \times \Sigma \to Q$ (例えば $\delta(s_{20}, c_{50}) = s_{70}$)，$s_{init} = s_0$ は $A$ の初期状態である．

また，FA は，入力される文字の列が書かれた 1 本のテープをもち，テープ上の文字を読むためのヘッドを備え，状態を保持する制御部をもつ機械であると考える（図 1･2)．左図から動作を始め，読んだ入力文字とそのときの状態に従ってヘッド位置を変えながら遷移し，全部の文字を読みきれば動作を終了する（右図)．動作終了時の状態が $F$ の元であるか否かによって，FA $A$ は入力文字列が求めるかたちのものであるか否かを認識（受理）する．

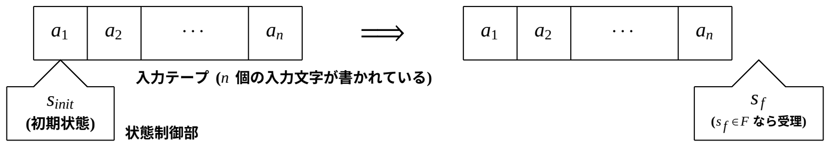



**図 1･2　入力テープをもつ有限オートマトン（左：動作開始時，右：動作終了時）**

FA は，テープ上に書かれた文字を書き換えることはできないし，最初に文字列が書かれた範囲を超えてテープを使うこともできない．これらの制約をなくし，更にヘッドも左右に自

由に移動できるようにしたオートマトンをチューリング機械（TM）という．

FA に様々な機能を付加することにより多様なオートマトンが定義できる．

**（1）読み取り専用テープ vs. 読み書き可能テープ**

FA のようにテープ上のデータを書き換えることができないもの（入力専用テープ）に対し，書き換えを許したテープをもつもの．

**（2）1 方向と 2 方向**

読み書きヘッドがテープ上を片方向にだけしか移動できないものに対し，両方向に自由に移動できるもの．

**（3）ヘッド数**

テープごとに読み書きヘッドが $k$ 個あるもの（$k$h で表す（1h は省略））．

**（4）補助記憶装置**

FA のように補助記憶装置がないものに対し，読み取り専用の入力テープ以外に様々な補助記憶装置（作業用テープ）を付加したものがある．例えば，プッシュダウンメモリ（PD），内部が読める PD であるスタックメモリ（S），S が入れ子になった各種のメモリ（nested stack: NS, nonerasing stack: NES, checking stack: CS, など），1 記号だけを記憶できる PD であるカウンター（C），読み書きに制限のないメモリ（TM），など．以下，FA に X タイプの記憶装置（作業テープ）を $k$ 個付加したものを +$k$X で表す．

**（5）決定性・非決定性・交代性・確率性**

FA のように，そのときの状態とそのとき入力された文字に対して次の状態（と文字の書き換えやヘッドの移動）が一意的であるものを決定性（D で表す）であるといい，複数の遷移の仕方を可能としたものを非決定性（N で表す）であるという．非決定性だけでなく更に並列計算機能を加えたもの（A で表す）を交代性という．交代性オートマトンでは指定された状態に入ると，その時点で複数の遷移をした後，それぞれの遷移先で計算が並列に進められる．一方，複数ある遷移可能性のうちのどれが選ばれるかが確率的に決められるものを確率オートマトンという．D, N, A のどれであるかは，テープヘッドの移動方向（片方向か両方向か）と組み合わせて 2D, 1N などのように表す．

**（6）メモリ量などの制限**

入力文字列の長さが $n$ のとき，作業テープ上で使えるメモリ量（文字列の長さ）や時間量（遷移回数）を $O(f(n))$ に制限する（$\mathrm{s}(f(n)), \mathrm{t}(f(n))$ で表す）．

**（7）出力機能**

出力用のテープをもち，1 ステップの動作のたびに有限長の文字列を出力テープ上に書き出すことができるもの．出力機能をもつ FA を順序機械ともいう．

**（8）無限長の語の認識**

オートマトンへの入力は長さが有限の文字列（語）であるが，これを無限長の語（$\omega$ 語）にまで拡張したものを $\omega$ オートマトンという．

**（9）文字列以外の認識**

通常の文字列（=1 次元の文字列）を一般化した $n$ 次元の文字列，ラベル付きの木，グラフなど，文字列以外の対象を認識するもの．

**（10）オートマトンのネットワーク**

同種のオートマトンを多数ネットワーク上に配置して並列に動作させるモデルのうち，$n$

次元空間の格子点上に配置した FA が近傍の FA の状態に依存して自分の状態を変えていくものをセルオートマトンという．

以上の機能を付加したオートマトンの能力の一例を以下に示す（1/2, D/N/A などはそれぞれの場合を対応させて読むこと．| はどちらでも等しいことを，多h は多ヘッドを表す）．

1. 1D-FA = 1N-FA = 2D-FA = 2N-FA = 1A-FA = 1|2 D|N・s(0) -TM.
2. 1D・$k$h-FA $\subsetneq$ 1N・$k$h-FA $\subsetneq$ 1A・$k$h-FA, 1 D/N・$k$h-FA $\subsetneq$ 2 D/N・$k$h-FA, ($k \geq 2$).
3. 2D・多h-FA=2D・s(log$n$)-TM ⊆ 2N・多h-FA=2N・s(log$n$)-TM ⊆ 2A・多h-FA=$\cup_k$2D・t($n^k$)-TM.
4. 1/2 D/N/A・$k$h-FA $\subsetneq$ 1/2 D/N/A・(k+1)h-FA.
5. D・+1PD $\subsetneq$ 1N・+1PD $\subsetneq$ 2N・+1PD $\subsetneq$ 1|2 A・+1PD = $\cup_c$ 2D・t($c^n$)-TM.
6. 2D・$k$h・+1NES = 2D・s($n^k \log n$)-TM $\subsetneq$ 2N・$k$h・+1NES = 2N・s($n^{2k}$)-TM.
7. 2D・+1S = $\cup_c$ 2D・t($n^{cn}$)-TM $\subsetneq$ 2N・+1S = $\cup_c$ 2D・t($2^{cn^2}$)-TM $\subsetneq$ 2A・+1S = $\cup_c$ 2D・t($2^{2^{cn}}$)-TM.
8. 1/2 D/N・+1C $\subsetneq$ 1/2 D/N・+2C, 2D・+2C = 1|2 D|N-TM.

■計算論とオートマトン - 1 章

## 1-2 形式言語と文法

（執筆者：守屋悦朗）[2009 年 1 月 受領]

1950 年代の中頃，N. Chomsky は自然言語の文法を定性的に研究するための数学モデルとして**形式文法・形式言語**の理論を提唱した．まず，記号や文字を要素とする有限集合を**アルファベット**という．Σ をアルファベットとするとき，Σ の元を重複を許して有限個並べた記号列（文字列）$a_1a_2\cdots a_n$ （$a_1, a_2, \ldots, a_n$ それぞれは Σ の要素) を Σ 上の**語**という．語 $w = a_1a_2\cdots a_n$ を構成している記号の個数 $n$ ($n \geq 0$) をこの語の長さという．Σ 上の語の全体を $\Sigma^*$ で表す（すなわち，$\Sigma^*$ は Σ の文字からなる長さが有限の語の集合である)．$\Sigma^*$ の部分集合を Σ 上の**言語**という．例えば，アルファベット $\{a, b, c\}$ 上の言語 $L$ を

$$L := \{\, xcy \mid x, y \in \{a, b\}^*,\ x \text{ の長さは偶数},\ y \text{ の長さは奇数} \,\}$$

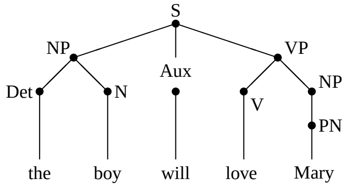
と定義する（$xcy$ は語 $x, c, y$ をこの順に並べてつなげた語を表す）と，$L = \{ca, cb, aaca, aacb, abca, abcb, baca, bacb, bbca, bbcb, \cdots\}$ である．

形式言語理論では，言語の文法的構造を記述するための数学モデルとして形式文法を考える．形式文法においては，文が有す構造的規則性（統語構造とか構文という）を，「文」，「名詞節」，「名詞句」，「名詞」，「固有名詞」，$\cdots$ といった文法機能の間の関係として記述する．例えば名詞句（NP）を「冠詞（Det）＋名詞（N)」，「冠詞＋形容詞句（AP）＋名詞」，「固有名詞（PN)」のいずれかであると考える場合には，このことを三つの関係

$$\mathrm{NP} \to \mathrm{Det\ N},\quad \mathrm{NP} \to \mathrm{Det\ AP\ N},\quad \mathrm{NP} \to \mathrm{PN}$$

により表す．→ は左辺の文法的機能が右辺のように分解されることを表す．当然，Det → an, Det → the, $\cdots$, N → boy, N → dog, $\cdots$ といった関係も存在する．ここに登場したそれぞれの文法機能を一つの文字で表し（それらの文字の集合を $V$ とする)，それぞれの単語も上述のように一つの文字で表す（それらの文字の集合を Σ とする)．このとき，上述の文法機能間の関係は $(V \cup \Sigma)^*$ 上の関係 $\alpha \to \beta$ の集合（$P$ とする）と考えることができる．「文」という文法機能を文字 $S$ で表せば，$G = (V, \Sigma, P, S)$ によって言語の文法的構造を規定できる（句構造文法という)，このように形式的に定義される文法を一般に**形式文法**という．「文（S)」から始めて，$P$ により定められた関係に従って順次，文法機能を細分化していく（このプロセスは，関係 $(\alpha \to \beta) \in P$ の左辺の文字列 $\alpha$ を右辺の文字列 $\beta$ で書き換えていくことに等しいので，$P$ の元のことを書き換え規則ともいう）と，この言語において文法的に正しい具体的な文が生成される（図 1･3 参照．この例では，$V$ や Σ の要素を 1 文字ではなく上述のような文字列で表している．$P$ は省略した)．こういう文字列の書き換え系は既に 1900 年代に研究されていて，Chomsky の句構造文法はこれをベースにしている．

Chomsky は書き換え規則のかたちに従い，句構造文法を 0 型，1 型，2 型，3 型の 4 種類に分類した．特に，プログラミング言語の構文記述方法の一つである BNF（バッカス記法）と本質的に等しい 2 型句構造文法は文脈自由文法とも呼ばれ，その重要性ゆえによく研究されている．文脈自由文法をベースにして様々な変種の文法が導入されている（代表的なものをカッコ内に記した．文献を探しやすいように，名称は英語で記した．G=grammar)．




**文法** $G = (V, \Sigma, P, S)$

$V$ = {S, NP, Aux, VP, Det, N, V, PN,$\cdots$}

$\Sigma$ = {the, boy, will, love, Mary,$\cdots$}

$G$ **における文の生成**

$S \Longrightarrow$ NP Aux VP $\Longrightarrow$ Det N Aux VP

$\Longrightarrow$ the N Aux VP $\Longrightarrow$ the boy Aux VP

$\Longrightarrow \cdots \Longrightarrow$ the boy will love Mary

**図 1·3　左図：文の文法的構造（構文木），右図：文の生成**

(1) 書き換え規則の適用位置や適用順を制限した文法（matrix G, programmed G, G with control set）
(2) 複数の書き換え規則を同時に適用できるように一般化した文法（scattered context G）
(3) 語の構文を表す木に基づき, 意味論的な制約を加えることで生成を制御する文法（attribute G, tree adjoining G, lexical functional G）
(4) 文法機能を表す文字（上述の $V$ の元）に何らかの機能（例えば，記憶装置）を付加した文法（indexed G, G with memory）
(5) オートマトンと同様な交代性の概念を導入した文法（alternating G）

句構造文法以外にも，独自の研究動機をもつ興味深い形式文法が多数ある．

(6) 線状生物の細胞生長モデルとして導入された成長系（L sysytem という）は，書き換えが並列に進められるだけでなく，生成される言語は 1 次元の文字列ではなく 2 次元の図形である．このような，文字列以外の語を生成する文法には，（ラベル付き）グラフあるいはラベル付き木を生成する各種の graph G/tree G，2 次元の語（ラベル付きの単位正方形を縦横に並べたもの）を生成する 2-dimensional G/array G などがある．
(7) 文法機能を表す文字も使い（使わないものもある)，ある種の書き換え規則に従って文字列を書き換えていくことは句構造文法と類似しているが，研究動機と生成の方法が異なるものに head G, categorial G, contextual G などがある．
(8) DNA 鎖の仕組みや遺伝子の組み換え原理を生成文法の仕組みとして捉えた文法（Head system, splicing system, insertion/deletion system）

語を「生成する」文法と，語を「認識する」オートマトンは密接な関係がある．代表的なものを以下に記す（オートマトンの型をカッコ内に記した)．

1. 0 型句構造文法 = チューリング機械 (1|2 D|N|A-TM)
2. 1 型句構造文法 = 線形有界オートマトン (2N・s($n$) -TM)
3. 2 型句構造文法 = プッシュダウンオートマトン (1N・+1PD)
4. 3 型句構造文法 = 有限オートマトン (1|2 D|N|A-FA)
5. 各種メモリ付き文法 = 各種記憶装置付き FA (1N・+1S など)
6. インデックス文法 = スタックオートマトンの変種 (1N・+1NS)
7. 交代性 1|2 型句構造文法 = 交代性 PDA (1|2 A・+1PD = 1|2 A・s($n$) -TM)

**■参考文献**

1) J.E. Hopcroft and J.D. Ullman, “Introduction to Auotmata Thoery, Languages, and Computation,” Addison-Wesley, Reading, Mass., 1979.
2) K. Wagner and G. Wechsung, “Computational Complexity,” D. Reidel Pub. Co., 1986.
3) G. Rozenberg and A. Salomaa, “Handbook of Formal Languages,” vols.1-3, Springer, Berlin, 1997.
4) 守屋悦朗, “形式言語とオートマトン,” サイエンス社, 2001.