ま　え　が　き

本書では，オートマトン理論の基礎と，その種々の応用について解説する。有限オートマトンは，有限個の状態間の状態遷移を表現するシンプルなモデルであるが，その簡潔さゆえに，情報系の学問分野において最も重要な計算モデルとなっており，さまざまな分野で幅広く利用されている。情報系の学問を学ぶ学生諸君は，この有限オートマトンの基礎と応用について，深く理解しておく必要がある。

従来から，有限オートマトンに基づくアプリケーションが多数開発されてきた。その中でも，特に，プログラミング言語のコンパイラが構成できたことは，コンピュータサイエンス分野における非常に大きなブレイクスルーである。コンパイラとは，C や Java といった人間にとって理解しやすい高級言語で書かれたプログラムを，コンピュータが理解できるバイナリ形式のコードに翻訳するプログラムである。このような翻訳を行う際には，字句解析や構文解析という処理が必要となるが，その処理を，有限オートマトンや，プッシュダウンオートマトンというオートマトンモデルを用いて行うということで，コンパイラの実現が可能となった。コンパイラ構成論において開発されたこのような技術は，現在では，機械翻訳や情報検索にも応用されている。

また，これまでに，上記のような翻訳系ソフトの他にも，画像圧縮，暗号，機械学習などのさまざまな分野において，オートマトンに基づく数多くのアプリケーションが開発されている。本書では，有限オートマトンに基づいて，種々のアプリケーリョンを開発していく分野のことを「応用オートマトン工学」と呼び，その概要について解説していく。

本書の構成は，以下のとおりである。まず，1 章ではオートマトン・言語理論の入門として，特に有限オートマトンや正則文法，正則言語に関する内容に

絞って解説し，つづく 2 章では，オートマトン理論の代表的な応用分野であるコンパイラ構成論について概観する。さらに，3 章，4 章では，オートマトン理論の新たな応用分野を紹介する。まず，3 章では，計算論的学習理論の応用として，ジュウシマツの歌の学習（文法規則の抽出）について説明し，4 章では，量子セルオートマトンの考え方を応用した画像圧縮の研究を紹介する。

本書の学習の進め方としては，まず，1 章と 2 章は，情報科学・工学の重要な基礎となる標準的内容なので，熟読し，章末の練習問題なども解いて，理解を深めていただきたい。一方，3 章と 4 章は，オートマトンを用いた文法抽出，画像圧縮に関するアプリケーションの構築事例をトピックス的に紹介したものなので，どのような考え方で，オートマトンを実際のアプリケーションに応用していくのか，その基本的な考え方について学習していただきたい。

本書の執筆は，1 章を若月が，2 章を後藤が，それぞれ担当し，それ以外の章は，西野が執筆した。

なお，1 章の執筆にあたっては富田悦次氏（電気通信大学名誉教授）にご協力いただき，富田悦次・横森　貴 共著「オートマトン・言語理論」（森北出版）の内容を大いに参考とさせていただいた。また，3 章の執筆にあたっては柿下容弓氏（株式会社 日立製作所）に，4 章の執筆にあたっては大畑和樹氏（電気通信大学大学院修士 2 年生）に，たいへんお世話になった。さらに，コロナ社には，遅れがちな原稿執筆を辛抱強くサポートしていただいた。上記の皆様に，この場をお借りして，深く感謝申し上げたい。

2011 年 12 月

西野　哲朗
若月　光夫
後藤　隆彰

目　　次

1. オートマトン理論

2. コンパイラ

3. 文法学習

4. 画像圧縮

1章 オートマトン理論

1.1 オートマトンと形式言語

1.1.1 オートマトン

コンピュータや電卓，電子辞書などのデジタルシステムは基本的に，論理演算を行う論理素子を組み合わせた組合せ回路や順序回路といったデジタル回路によって構成されており，キーボードやマウスなどから**入力**（input）があると，システム内部でそれに対応した適切な処理を自動的に行い，その結果を画面などに**出力**（output）する。**図 1.1** には，例として，コンピュータのキーボードから英字キーを打って入力すると，それに対応した英文字がディスプレイ画面に表示されることを模式的に書いてある。

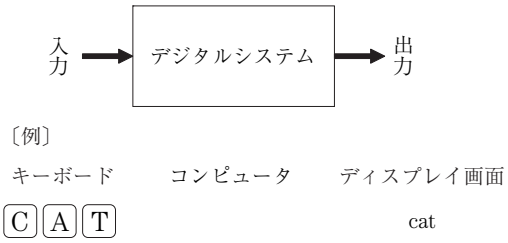

図 1.1 デジタルシステムとその例

このようなデジタルシステムを極力簡単化した数理的なモデルが，**オートマトン**（automaton，複数形は automata）である。オートマトンには，離散化された各時点において入力があり，その入力には通常，英小文字 $(a, b, c, \ldots)$ や数

字（$0, 1, 2, \ldots$）などの記号が使われる。これらの抽象化された個々の入力は，**入力記号**（input symbol）と呼ばれる。個々のオートマトンにおいては，取り扱える入力記号の種類を有限個に限定し，これを入力記号の有限集合（例えば，$\Sigma = \{a, b\}$ (a と b の 2 記号からなる集合) など）として明示する。また，入力記号がつづいて入力されたときに与えられる系列を**入力記号列**（input string），あるいは**入力系列**（input sequence）と呼ぶ。オートマトンの出力についても同様に扱う。特定のオートマトンを定めるには，入力記号および出力記号の有限集合をそれぞれ明示する他，内部の記憶機構や，入力を読み込んだときの動作の仕方に関する規則を決める必要がある。

入力記号が入力されるとそれに対応した出力記号列がただちに出力されるような部類のオートマトンは，**変換器**（transducer）と呼ばれる。代表的な古典暗号の一つである換字式暗号のうち，ビジネル暗号（Vigenere cipher）[9)†]と呼ばれる暗号化の鍵を周期的に使用するような暗号は，**図 1.2** のような，内部に記憶機構として鍵の長さと同じ個数の**状態**（state）をもつ，**順序機械**（sequential machine）と呼ばれる変換器として表現できる。この例では，暗号化鍵である

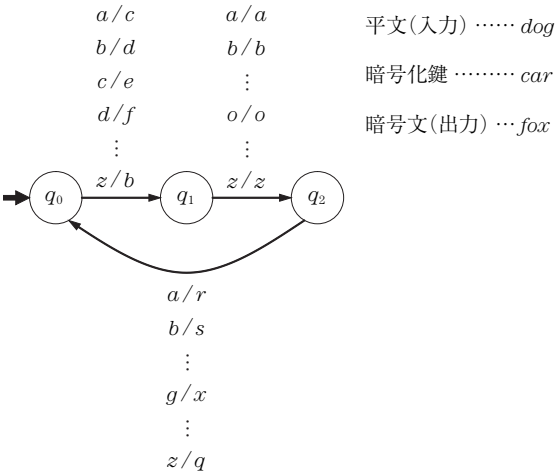

図 1.2　順序機械の一例

† 肩付番号は，巻末の引用・参考文献の番号を表す。

“*car*” に対応した順序機械があらかじめ与えられ，暗号化する前の平文として “*dog*” が入力されると，暗号化鍵に従って決められた，状態遷移に関する規則（**状態遷移関数**（state transition function）という）および出力に関する規則（**出力関数**（output function）という）が適用され，暗号文として “fox” が出力される。ここで，図中の q_0 のように，入力の読込み開始時点に設定される状態を，**初期状態**（initial state）と呼ぶ。

順序機械としてモデル化できるシステムの記憶量は，有限個の内部状態数に応じた有限の範囲内に限られるため，ある種の順序機械は有限オートマトンとも呼ばれる。さらに複雑な記憶機構を有するオートマトンとしては，**プッシュダウンオートマトン**（pushdown automaton）や，**線形拘束オートマトン**（linear bounded automaton），**チューリング機械**（Turing machine）などがある。どんなに高度なコンピュータであろうとも，そのようなコンピュータによって計算できることはチューリング機械によっても必ず計算できるため，チューリング機械はコンピュータの基本的機能をモデル化していると考えることができる。

1.1.2 形 式 言 語

われわれが日常使用している日本語や英語などは**自然言語**（natural language）と呼ばれ，Pascal や C，Java など，コンピュータに命令を与えるプログラムを記述するための言語は**プログラミング言語**（programming language）と呼ばれる。これらの言語を抽象化したものが**形式言語**（formal language）である。

A から Z までの英字 26 文字を英語のアルファベットというが，一般に，**記号**（symbol）あるいは**文字**（letter）の空でない有限集合を**アルファベット**（alphabet）という。アルファベット Σ 中の記号を重複を許して有限個並べたものを，Σ 上の**記号列**（または**文字列**）（string）あるいは**語**（word）という。一般的に，Σ 中の記号を n 個並べて得られる記号列は，$a_1a_2\cdots a_n$ のように書く。ここで，$i = 1, 2, \ldots, n$ に対して $a_i \in \Sigma$ である。記号列 w について $w = a_1a_2\cdots a_n$ と表されるとき，w を構成している記号の数 n を w の**長さ**（length）といい，$|w|$ で表す。例えば，$w = abcc$ は $\Sigma = \{a, b, c\}$ 上の記号列であり，$|w| = 4$

である。特に，$n = 0$ の場合，すなわち長さが 0 の記号列を**空記号列**（empty string）と呼び，ε（epsilon）で表す。

二つの記号列 $x = a_1a_2\cdots a_m$，$y = b_1b_2\cdots b_n$ $(m, n \geqq 0)$ に対して，x の後に y をつないで得られる記号列 $a_1a_2\cdots a_mb_1b_2\cdots b_n$ を，x と y の**連接**（concatenation）といい，$x \cdot y$ あるいは単に xy と書く。例えば，$\Sigma = \{a, b\}$ で，$x = ab$, $y = bab$ としたとき，$xy = ab \cdot bab = abbab$ である。特に，任意の記号列 x に対して，$x\varepsilon = \varepsilon x = x$ である。同じ記号 a を n 個並べて得られる記号列を a^n，また，同じ記号列 x を n 回連接して得られる記号列を x^n と表すこともある。

記号列 w について $w = xyz$ と表されるとき，x を w の**接頭辞**（prefix），z を w の**接尾辞**（suffix）という。また，x, y, z はそれぞれ，w の**部分記号列**（substring）であるという。特に，$yz \neq \varepsilon$ のとき，x は $w\,(= xyz)$ の**真の**（proper）接頭辞，$xy \neq \varepsilon$ のとき，z は w の真の接尾辞，などという。

空記号列 ε を含む Σ 上の記号列全体からなる無限集合を Σ^* で表し，空記号列 ε 以外の Σ 上の記号列からなる無限集合を Σ^+ で表す。したがって，$\Sigma^* = \{\varepsilon\} \cup \Sigma^+$ である。例えば，$\Sigma = \{a, b\}$ であるとき

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, \ldots\}$$

である。Σ^* のある部分集合 L を，Σ 上の**言語**（language）という。

二つの言語 $L_1, L_2 \subseteq \Sigma^*$ に対して，L_1 中の記号列と L_2 中の記号列とをこの順に連接して得られる記号列全体の集合，すなわち

$$L_1L_2 = \{xy \mid x \in L_1 \text{ かつ } y \in L_2\} \tag{1.1}$$

を言語 L_1 と L_2 の**連接**と呼び，$L_1 \cdot L_2$ あるいは単に L_1L_2 と書く。例えば，$\Sigma = \{a, b\}$ で，$L_1 = \{\varepsilon, a, aa\}$, $L_2 = \{ba, baba\}$ としたとき，$L_1L_2 = \{ba, baba, aba, ababa, aaba, aababa\}$ である。特に，空記号列 ε のみからなる集合 $\{\varepsilon\}$ に関しては，任意の言語 L に対して，$L\{\varepsilon\} = \{\varepsilon\}L = L$ である。また，空集合 $\emptyset$ に関しては，$L\emptyset = \emptyset L = \emptyset$ と定義する。

ある言語 $L \subseteq \Sigma^*$ に対して，L を n 回続けて連接した集合を L^n で表す。すなわち

1. $L^0 = \{\varepsilon\}$
2. $n \geqq 1$ である任意の n に対して，$L^n = LL^{n-1}$

と帰納的に定義する。また

$$L^* = \bigcup_{n=0}^{\infty} L^n = \{\varepsilon\} \cup L \cup L^2 \cup \cdots \tag{1.2}$$

と定義し，これを L の**クリーネ閉包**（Kleene closure），あるいは**スター閉包**（star closure），あるいは単に**閉包**（closure）という。すなわち，L^* は L 中の任意個数の記号列を任意回数，任意の順序で並べて得られる記号列全体からなる無限集合である。例えば，$L = \{a, ba\}$ であるとき

$$\begin{aligned}
L^2 &= \{aa, aba, baa, baba\},\\
L^3 &= \{aaa, aaba, abaa, ababa, baaa, baaba, babaa, bababa\},\\
&\vdots\\
L^* &= \{\varepsilon, a, ba, aa, aba, baa, baba, aaa, aaba, abaa, ababa,\\
&\qquad baaa, baaba, babaa, bababa, \ldots\}
\end{aligned}$$

である。特に，空記号列 ε のみからなる集合 $\{\varepsilon\}$ に関しては，$\{\varepsilon\}^n = \{\varepsilon\} (n \geqq 1)$ より，$\{\varepsilon\}^* = \{\varepsilon\}$ である。また，空集合 $\emptyset$ に関しては，$\emptyset^n = \emptyset\,(n \geqq 1)$ より，$\emptyset^* = \{\varepsilon\}$ である。

また，言語 $L \subseteq \Sigma^*$ に対して

$$L^+ = \bigcup_{n=1}^{\infty} L^n = L \cup L^2 \cup L^3 \cup \cdots \tag{1.3}$$

と定義し，これを L の**正閉包**（positive closure）と呼ぶ。したがって，$L^* = \{\varepsilon\} \cup L^+$ である。

連接や（正）閉包を求めることは，言語に対する一種の演算とみなすことができる。また，言語は記号列の集合であるから，言語 L_1, L_2 に対する和集合

$(L_1 \cup L_2)$，共通集合（または積集合）$(L_1 \cap L_2)$，差集合 $(L_1 - L_2)$ といった各集合演算も，言語に対する演算と考えることができる。

Σ 上の言語 L は一般に無限集合であり，これを有限の記述で Σ^* 中から規定する必要がある。そのための一つの方法として，**形式文法**（formal grammar）と呼ばれる文法が用いられる。指定した形式文法の規則に従って Σ 上の記号列が生成されるとき，その記号列をその文法によって生成される**文**（sentence）と呼ぶ。ちょうど言語 L 中の文だけを生成する文法を指定することによって，言語 L が規定される。言語 L を規定するためのもう一つの方法として，**オートマトン**による方法がある。変換器は入力記号が読み込まれる度にそれに対応した出力記号列が出力されるが，入力記号列を読み尽くした後にのみ，“1”（受理，yes）か “0”（非受理，no）いずれかの応答を出力する**認識機械**（recognizer）として，オートマトンを定めることもできる。L 中の記号列に対してだけ，それを入力して読み尽くした後に出力 “1” の応答を出すオートマトンを指定することで，言語 L を規定できる。以上のように，オートマトンと形式文法とは言語を介して密接な関係がある。なお，Σ 上の言語をオートマトンによって考える場合，Σ は入力記号の有限集合であり，文法によって考える場合，Σ は終端記号の有限集合と呼ばれるものである。

形式言語は，その構造に従って，正則言語，文脈自由言語，文脈依存言語，句構造言語と階層的に分類される。ここで，前者の言語（集合）は後者の言語の特別な場合となっていることが知られている。

1.2　有限オートマトン

有限オートマトン（finite automaton，**FA**），あるいは**有限状態オートマトン**（finite state automaton，**FSA**）は，内部の記憶機構として有限個の状態をもつ，認識機械とみなされるオートマトンである。有限オートマトンは形式的に，つぎに示すような 5 項組 $(Q, \Sigma, \delta, q_0, F)$ によって定義される。

1.　Q は状態の有限集合

索　　引

―― 著 者 略 歴 ――

西野　哲朗（にしの　てつろう）
1982 年　早稲田大学理工学部数学科卒業
1984 年　早稲田大学大学院博士前期課程修了（数学専攻）
1984 年　日本アイ・ビー・エム株式会社勤務
1987 年　東京電機大学助手
1991 年　理学博士（早稲田大学）
1992 年　北陸先端科学技術大学院大学助教授
1994 年　電気通信大学助教授
2006 年　電気通信大学教授
　　現在に至る

若月　光夫（わかつき　みつお）
1988 年　電気通信大学電気通信学部通信工学科卒業
1990 年　電気通信大学大学院博士前期課程修了
　　（電子情報学専攻）
1993 年　電気通信大学大学院博士後期課程修了
　　（電子情報学専攻）
　　博士（工学）
1993 年　電気通信大学助手
2007 年　電気通信大学助教
　　現在に至る

後藤　隆彰（ごとう　たかあき）
2001 年　東洋大学工学部情報工学科卒業
2003 年　東洋大学大学院博士前期課程修了（情報工学専攻）
2003 年　パシフィックシステム株式会社勤務
2008 年　電気通信大学 e ラーニング推進センター技術補佐員
2009 年　東洋大学大学院博士後期課程修了（情報工学専攻）
　　博士（工学）
2009 年　電気通信大学産学官連携センター特任助教
　　現在に至る

応用オートマトン工学

Applied Automata Engineering

2012 年 2 月 18 日　初版第 1 刷発行　★

検印省略

著　者　西野哲朗
　　若月光夫
　　後藤隆彰
発行者　株式会社　コロナ社
　　代表者　牛来真也
印刷所　三美印刷株式会社

112–0011　東京都文京区千石 4–46–10
発行所　株式会社　コロナ社
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　http://www.coronasha.co.jp

ISBN 978–4–339–02459–3　（金）　（製本：愛千製本所）

Printed in Japan