■6 群（コンピュータ‐基礎理論とハードウェア）- 2 編（計算論とオートマトン）

2 章 有限オートマトンと正則表現

（執筆者：岩本宙造）［2010 年 3 月 受領］

■概要■

本章では，順序機械，決定性・非決定性の有限オートマトン，正則表現と正則言語，量子オートマトンについて概覧する．まず，順序回路を，入力の列に対して出力の列がきまる文字列関数と考える．その順序回路を，状態集合，遷移関数，出力関数で定式化して順序回路を定義し，順序回路で実現できる関数などについて概説する．次に，順序機械から有限オートマトン研究への成り立ちについて述べ，決定性と非決定性の有限オートマトンを定義する．また，Nerode の定理や，決定性と非決定性の能力の同等性，木オートマトンなどについて述べる．さらに，記号列の集合を簡潔に表現するための手法である正則表現について，有限オートマトンとの関係や，所属問題，正則言語について概説する．最後に，量子計算機の単純なモデルである量子有限オートマトンについて，定義や言語クラス，計算資源について説明する．

【本章の構成】

本章は，順序機械（2-1 節），決定性有限オートマトン（2-2 節），非決定性有限オートマトン（2-3 節），正則表現と正則言語（2-4 節），量子オートマトン（2-5 節）の 5 節からなる．

■6 群 - 2 編 - 2 章

2-1 順序機械

（執筆者：河原康雄・溝口佳寛）[2009 年 1 月 受領]

2-1-1 順序回路の例

図 2･1 は順序回路の基本要素の一つである RS 型フリップフロップ回路の入出力の例である. 順番に入力された信号に従って出力が順番に変化する. 出力はオン（1）かオフ（0）であり, 出力集合は $Y = \{0, 1\}$ と考えられる．入力は S（セット）と R（リセット）の 2 本あるが, これらを並べて (SR) で 2 進数で考えて入力集合は，$X = \{0 = (00), 1 = (01), 2 = (10)\}$ の三つ元の集合と考える．この順序回路は状態集合を $Q = \{a, b\}$ とし, 下表で定義される状態遷移関数 $\delta : Q \times X \to Q$, 出力関数 $\beta : Q \to Y$ を用いて，順序機械としてモデル化される．状態 q のとき, 入力 x があうと次の状態 $\delta(q, x)$ に遷移し, $\beta(\delta(q, x))$ が出力される．例では，最初の状態を a として, 入力 “0201021” に対する出力が “0110010” となっている．

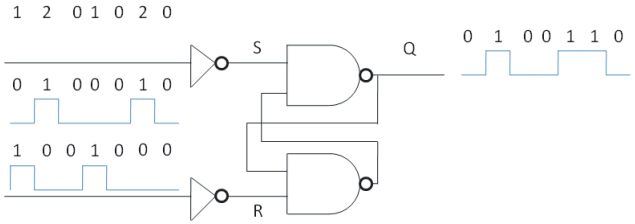


図 2･1

この順序機械を**状態遷移図**（state transition diagram）で表すと図 2･2（右）のようになる．状態を頂点，遷移を辺で表し，辺の上に入力文字，状態の下に出力文字を書く．入力文字に対し，辺をたどることで状態の変化や出力文字を検証できる．

q	x	$\delta(q, x)$
a	0	a
a	1	a
a	2	b
b	0	b
b	1	a
b	2	b

q	$\beta(q)$
a	0
b	1

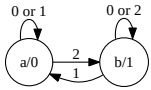


図 2･2

順序回路とは，入力の列に対して出力の列が定まる文字列関数と考えることができる．すなわち，X^* から Y^* への関数 $f : X^* \to Y^*$ である．ただし，ここで，X^* は空列（ここでは ε で表す）を含む X 上の文字列全体の集合である．順序回路を状態集合，遷移関数，出力関数で定式化し順序機械を定義し, どのような文字列関数が実現可能なのか，また，より少ない状態数での実現が可能かを問題として考える．

2-1-2 準備 -集合と合同関係-

集合 X から集合 Y への関数 $F : X \to Y$ は，$F(x) = m(e(x))$ $(x \in X)$ となる全射 $e : X \to Z$ と単射 $m : Z \to Y$ の合成で表すことができる．このとき，集合 Z は同型を除いて唯一に定まり，$Z \cong F(X) = \{F(x) \mid x \in X\}$ であり，また，$Z \cong X/\sim = \{[x] \mid x \in X\}$ でもある．ただし，X 上の合同関係 $\sim$ は，$[x \sim x' \text{ iff } F(x) = F(x')]$ で定め，$[x]$ は x を含む同値類 $\{x' \in X \mid x \sim x'\}$ を表す．

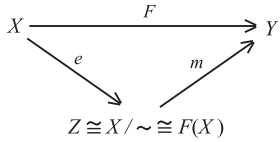


図 2･3

（補足） $e : X \to Z$ が全射とは任意の元 $z \in Z$ に対して $e(x) = z$ となる元 $x \in X$ が存在することである．また，$m : Z \to Y$ が単射とは任意の元 $z_1, z_2 \in Z$ に対して，$z_1 \neq z_2$ ならば $m(z_1) \neq m(z_2)$ であることである．

2-1-3 順序機械

順序機械（sequential machine）とは，状態集合 Q，入力記号の有限集合 X，出力記号の有限集合 Y，状態遷移関数 $\delta : Q \times X \to Q$，出力関数 $\beta : Q \to Y$，初期状態 $q_0 \in Q$ の六つ組 $M = (Q, X, Y, \delta, \beta, q_0)$ のことである．状態集合が有限集合のとき**有限順序機械**という．

（補足） 本定義は **Moore 型**の順序機械と呼ばれる．出力関数を β の代わりに $\lambda : Q \times X \to Y$ で与える定義を **Mealy 型**の順序機械という．この二つは等価なモデルであり，Mealy 型の順序機械は初期状態に対する出力が存在しないが，それ以外は入力と出力の関係が同値であるように相互に変換可能である．また，初期状態を定義には入れず，五つ組で順序機械を定義することもある．

状態遷移関数 $\delta : Q \times X \to Q$ は，$\delta^*(q, \varepsilon) = q$，$\delta^*(q, xw) = \delta^*(\delta(q, x), w)$ $(q \in Q,\ x \in X,\ w \in X^*)$ と定義し，自然に $\delta^* : Q \times X^* \to Q$ に拡張できる．また，一般に関数 $f : X^* \to Y$ は，$f_*(\varepsilon) = f(\varepsilon)$，$f_*(wx) = f_*(w)f(wx)$ $(x \in X, w \in X^*)$ と定義し，自然に $f_* : X^* \to Y^*$ に拡張できる．順序機械 $M = (Q, X, Y, \delta, \beta, q_0)$ に対して，$f_M : X^* \to Y$ を $f_M(w) = \beta(\delta^*(q_0, w))$ $(w \in X^*)$ とし，その拡張 $f_{M*} : X^* \to Y^*$ を考えると，これが入力と出力の間の文字列関数となっている．

関数 $t : X^* \to Y^*$ に対して，順序機械 M が存在して，$t = f_{M*}$ となるとき，関数 t は**順序機械で実現可能**という．関数 $t : X^* \to Y^*$ が順序機械で実現可能であるための必要条件は t が，ある関数 $f : X^* \to Y$ の拡張 $t = f_*$ であることである．このことは，任意の $w \in X^*$，$x \in X$ に対して，$t(wx) = t(w)y$ を満たす $y \in Y$ が存在することと同値であることが示される．すなわち $t(wx)$ の前半の出力は前半の入力 w だけに依存し，その後の x には影響されないことである．この条件を満たす関数 $t : X^* \to Y^*$ を**順序関数**という．

次に任意の順序関数が順序機械で実現可能であること，すなわち，関数 $f : X^* \to Y$ が与えられたとき，$f = f_M$ となる順序機械 M が存在することを 2 通りの方法で順序機械を構成して示す．一つ目は最も形式的な順序機械，入力文字列全体を状態とする順序機械，

$M_I = (X^*, X, Y, \delta_I, f, \varepsilon)$ である．ここで，$\delta_I(w, x) = wx$ $(w \in X^*,\ x \in X)$ とする．二つ目は最も抽象的な順序機械, 入出力関数全体を状態とする順序機械, $M_T = (Y^{X^*}, X, Y, \delta_T, \beta_T, f)$ である．ここで，Y^{X^*} は X^* から Y への関数全体 $\{f \mid f : X^* \to Y\}$ を表し，$\delta_T(f, x) : X^* \to Y$ は $\delta_T(f, x)(w) = f(xw)$ $(x \in X,\ w \in X^*)$ で定義し，$\beta_T(f) = f(\varepsilon)$ $(f \in Y^{X^*})$ とする．このとき，$f = f_{M_I} = f_{M_T}$ となるが，M_I も M_T も有限順序機械ではないことに注意する．

では，有限順序機械で実現可能な関数 $f : X^* \to Y$ とは，どのような関数であろうか？

答えは M_T のなかにある．順序機械 M_T の無限にある状態のすべては必要ない．初期状態 f からすべての入力 w に対して δ_T で状態遷移をした結果が有限の範囲にあればよい．すなわち，$Z = \{\delta_T^*(f, w) \in Y^{X^*} \mid w \in X^*\}$ が有限集合であればよい．この条件「Z が有限集合であること」が有限順序機械で実現可能な条件である．$F(w) = \delta_T^*(f, w)$ $(w \in X^*)$ とすると，$F : X^* \to Y^{X^*}$ であり，$Z = F(X^*)$ である．本章 2-1-2 の全射・単射分解の性質を使えば，$Z = X^*/\sim$ であり，同値関係 $(\sim)$ は，$[w \sim w' \text{ iff } \delta_T^*(f, w) = \delta_T^*(f, w')]$，言い換えると任意の $z \in X^*$ に対して $\delta_T^*(f, w)(z) = \delta_T^*(f, w')(z)$，すなわち，$f(wz) = f(w'z)$ である．この同値関係による同値類が有限個であれば，有限順序機械で実現可能である．更に，M_T の状態を Z に制限した順序機械が f を実現する状態数最小の順序機械であることが分かる．

最初に与えられる $f : X^* \to Y$ が順序機械 $M = (Q, X, Y, \delta, \beta, q_0)$ から定まる $f_M : X^* \to Y$ のとき，$F(w) = \delta_T^*(f_M, w)$ で定まる $F : X^* \to Y^{X^*}$ は，$f_e : X^* \to Q$ と $f_m : Q \to Y^{X^*}$ の合成に分解できて，$F(W) = f_m(f_e(w))$ となる．ここで，$f_e(w) = \delta^*(q_0, w)$ であり，$f_m(q)(w) = \beta(\delta^*(q, w))$ である $(w \in X^*,\ q \in Q)$．f_e が全射のとき M を**可到達**（reachable），f_m が単射のとき M を**可観測**（obserbable），または，**既約**（reduced）という．そして，可到達，かつ，可観測のとき，順序機械 M は f_M を実現する状態数最小（minimal）な実現であることが分かる．

$f_m : Q \to Y^{X^*}$ が単射でないときは，更に，f_m を全射・単射分解することで状態数最小の順序機械を構成することが可能である．このとき，Q 上の同値関係 $[q \sim q' \text{ iff } f_m(q) = f_m(q')]$ は，任意の $w \in X^*$ に対して，$f_m(q)(w) = f_m(q')(w)$ であること，すなわち，$\beta(\delta^*(q, w)) = \beta(\delta^*(q', w))$ であることである．Q が有限集合（サイズを n とする）の場合には，すべての $w \in X^*$ について調べなくても，長さ n 以下の w について $\beta(\delta^*(q, w)) = \beta(\delta^*(q', w))$ であれば，$q \sim q'$ であることが示され，与えられた有限順序機械から状態数最小の順序機械をアルゴリズムに従って具体的に構成することができる．

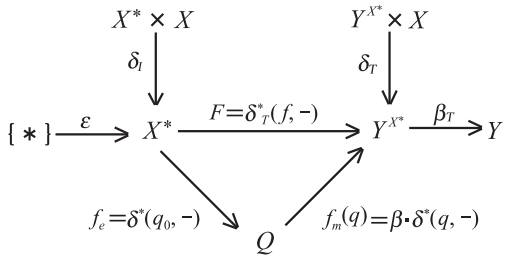


図 2･4

■参考文献

1) M.A. Arbib and E.G. Manes, “Machines in a cagegory, an expository introduction,” SIAM Review, vol.16, pp.163-192, 1974.
2) T.L. Booth, “Sequential Machines and Automata Theory,” John Wiley & Sons, 1967.
3) “Sequential Machines: Selected Papers,” ed. by E.F. Moore, Addison-Wesley 1964.
4) 高原康彦, 飯島淳一, “システム理論,” 共立出版, 1990.
5) 藤野精一 編集, “計算数学ハンドブック,” 朝倉書店, 1977.

■6 群 - 2 編 - 2 章

2-2 決定性有限オートマトン

（執筆者：河原康雄・溝口佳寛）[2009 年 1 月 受領]

2-2-1 有限オートマトン

オートマトンの研究が知られるようになったのは，情報理論の創始者としてで著名な C.E. Shanon と人工知能研究で著名な J. McCarthy の編集により 1956 年に発行された論文集 Automata Studies[2)]からである．当初は，順序回路の抽象化により状態遷移と入出力の関係についての様々な考察がなされていた．その状態集合のなかに特別な状態（受理状態）の集合を導入することにより，認識機械としての考察が始まり，言語との重要な関係が導かれ，有限オートマトンと言語の理論体系が構築された．この有限オートマトンの理論の最初の論文が，M.O. Rabin と D. Scott の 1959 年の論文[1, 3)]である．彼らは本成果により，1976 年に ACM チューリング賞を受賞している．

2-2-2 アルファベット, 語, 言語

空でない有限集合を**アルファベット**と呼び，その要素を文字，または，記号と呼ぶ．文字を重複を許して有限個並べた文字列を語，または，テープと呼び，語 w に含まれる文字の個数を w の長さと呼び，$|w|$ で表す．文字を 0 個並べた語を空語と呼び，ε で表す，アルファベット Σ 上の空語を含む語全体の集合を Σ^* で表す．二つの語をつないで書き並べると新しい長い文字列ができる．この操作を連接と呼び，語 w_1 と語 w_2 の連接を $w_1 \cdot w_2$ で表す（まぎらわしくない場合は，単に w_1w_2 と書く）．任意の語 w に対して，$\varepsilon w = w\varepsilon = w$ であり，Σ^* は連接を積，ε を単位元とする自由半群（モノイド）となる，

Σ^* の部分集合を（アルファベット Σ 上の）形式言語，または，単に言語という．語の連接は言語の連接に拡張される．すなわち，言語 L_1，L_2 の連接は，$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1,\ w_2 \in L_2\}$ である．言語 L に対して，$L^0 = \{\varepsilon\}$，$L^n = L^{n-1} \cdot L$ $(n \geq 1)$ とし，$L^* = \cup_{n=0}^{\infty} L^n$ とする．言語 L^* を言語 L の Kleene 閉包，または，単に閉包という．アルファベット Σ を言語と考えた際の閉包は語全体の集合 Σ^* になり記号に混乱はない．

2-2-3 決定性有限オートマトン

決定性有限オートマトン（derministic finite automaton）とは，状態の有限集合 Q，アルファベット Σ，状態遷移関数 $\delta : Q \times \Sigma \to Q$，初期状態 $q_0 \in Q$，最終状態 $F \subset Q$ の五つ組 $M = (Q, \Sigma, \delta, q_0, F)$ のことである．

有限オートマトンは図 2･5 のように表され，入力語がテープの上に並べられ，最初に M が初期状態 q_0 で左端の記号を読む．状態 q で入力記号 s のとき，状態を $\delta(q, s)$ に遷移させ，テープのヘッドを 1 文字右に移動する．この操作を繰り返し，入力語を読み終わったときの状態が最終状態かどうかを出力する．

状態遷移関数 $\delta : Q \times \Sigma \to Q$ は，$\delta^*(q, \varepsilon) = q$，$\delta^*(q, aw) = \delta^*(\delta(q, a), w)$ $(a \in \Sigma, w \in \Sigma^*)$ と定義し，自然に $\delta^* : Q \times \Sigma^* \to Q$ に拡張できる．初期状態 q_0 から語 w を読み終わったときの状態が最終状態 F に含まれるとは，$\delta^*(q_0, w) \in F$ で表される．このとき，M は語 w を受理するという．M で受理される語全体の集合を $L(M)$ と書き，M で受理される言語という．すなわち，$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$ が有限オートマトン M で受理される言語である．

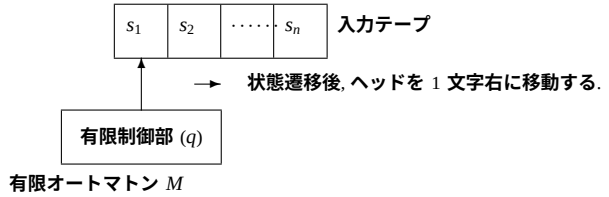


図 2･5

例 1: 有限オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ を $Q = \{q_0, q_1, q_2\}$, $F = \{q_1\}$, $q_0 \in Q$ とし, $\delta : Q \times \Sigma \to Q$ を以下で定義する. $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_2$, $\delta(q_1, a) = q_2$, $\delta(q_1, b) = q_0$, $\delta(q_2, a) = q_2$, $\delta(q_2, b) = q_2$.

例えば, 語 $w = aba$ は, $\delta^*(q_0, aba) = \delta^*(\delta(q_0, a), ba) = \delta^*(q_1, ba) = \delta^*(\delta(q_1, b), a) = \delta^*(q_0, a) = \delta^*(\delta(q_0, a), \varepsilon) = \delta^*(q_1, \varepsilon) = q_1 \in F$ であり, M によって受理される.

M は図 2･6 のような状態遷移図でも表される. 頂点のラベルが状態であり, 入力文字列に従って辺をたどり次の遷移状態のある頂点に移る. 最終状態が二重丸で表されており, 語を読み終わったときに最終状態かどうかを判定できる.

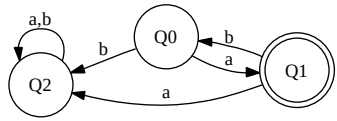


図 2･6

2-2-4 Nerode の定理

Σ^* 上の同値関係 $\sim$ が任意の語 $w_1, w_2 \in \Sigma^*$ に対して, 「$w_1 \sim w_2 \Rightarrow$ 任意の $z \in \Sigma^*$ に対して $w_1 z \sim w_2 z$」を満たすとき, **右不変**という. 同値関係 $\sim$ による同値類全体の集合 $\{[x] \mid x \in \Sigma^*\}$ が有限集合のとき, 同値関係 $\sim$ を**有限指数**という. ここで, $[x]$ は語 x を含む同値類 $\{x' \mid x \sim x'\}$ を表わす. 言語 $L \subset \Sigma^*$ に対して, Σ^* 上の関係を「$w_1 \sim_L w_2$ iff 任意の $z \in \Sigma^*$ に対して $w_1 \in L \Leftrightarrow w_2 \in L$」で定めると, $\sim_L$ は右不変な合同関係になる.

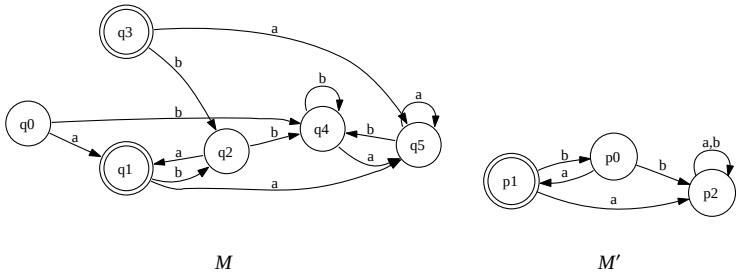
定理:（Nerode の定理） アルファベット Σ 上の言語 $L \subset \Sigma^*$ について次の命題 (1), (2), (3) は同値である. **(1)** L を受理する有限オートマトン M が存在する. **(2)** L は Σ^* 上の有限指数をもつ右不変な合同関係による同値類のいくつかの和集合である. **(3)** L から導かれる右不変な合同関係 $\sim_L$ は有限指数である.

上記の (3) を満たす言語 L に対して, $Q = \{[w] \mid w \in \Sigma^*\}$, $\delta([w], a) = [wa]$, $q_0 = [\varepsilon]$, $F = \{[w] \mid w \in\}$ で定めると, $M = (Q, \Sigma, \delta, q_0, F)$ は決定性有限オートマトンになる, 更に, L を受理する状態数最小の決定性有限オートマトンであり同型を除いて一意に定まる.

本章 2-1 の順序機械 $M = (Q, X, Y, \delta, \beta, q_0)$ で出力集合 Y が 2 点集合，例えば，$Y = \{0, 1\}$ のとき，出力関数 $\beta : Q \to \{0, 1\}$ と Q の部分集合 $F = \{q \in Q | \beta(q) = 1\}$ とは 1 対 1 に対応する．すなわち，出力集合が 2 点集合の有限順序機械を決定性有限オートマトンと考えることができる．更に，関数 $f : X^* \to Y$ は X^* の部分集合に対応する．部分集合 L に対応する $f_{:}X^* \to \{0, 1\}$ は $f(w) = 1$ $(w \in L)$, $f(w) = 0$ $(w \notin L)$ である．本章 2-1 において最小状態数の順序機械を構成する際に用いた同値関係 $w \sim w'$ は任意の $z \in X^*$ に対して $f(wz) = f(wz')$ であることであり，これは言語 L から導かれる関係 $\sim_L$ のことである．

順序機械 M で定まる順序関数 $f_M : X^* \to Y$ に対応する部分集合が M の受理言語 $L(M)$ である．与えられた順序機械から状態数最小の順序機械を構成する際は Σ を同値類に分けるのではなく，M の状態集合 Q を同値類に分ける．その際の同値関係 $q \sim q'$ は任意の $z \in X^*$ に対して $\beta(\delta^*(q, z)) = \beta(\delta^*(q', z))$ であり，すなわち，$\delta^*(q, z) \in F \Leftrightarrow \delta^*(q', z) \in F$ である．状態集合 Q がサイズ n の有限集合の場合，任意の長さの語 $z \in X^*$ に対して同値性を確認しなくても長さ n 以下の語 z に対してすべての $\delta^*(q, z)$ を計算することにより同値類，すなわち，状態数最小の有限オートマトンを構成できる．

例 2: 図 2･7 の有限オートマトン M に対して，その状態を最小化したものが M' である．ここで, $p_0 = \{q_0, q_2\}$, $p_1 = \{q_1, q_3\}$, $p_2 = \{q_4, q_5\}$ である．



M M'

図 2･7

■参考文献

1) M.O. Rabin and D. Scott, “Finite Automata and Their Decision Problems,” IBM Journal, vol.3, pp.114-125 1959.
2) “Automata Studies,” ed. by C.E. Shannon and J. Mac Carthy, Princeton University Press, 1956.
3) “Sequential Machines; Selected Papers,” ed. by E.F. Moore, Addison-Wesley, 1964.
4) 野崎明弘, 高橋正子, 町田元, 山崎秀記共訳, “オートマトン 言語理論 計算論 I (第 2 版),” サイエンス社, 2003;（原著: J.E. Hopcroft and J.D. Ullman, “Formal Languages and Their Relation to Automata, 2nd. ed.,” Addison-Wesley, 2001）
5) 野崎明弘, 町田元, 山崎秀記, 横森貴共訳, “計算論とオートマトン理論,” サイエンス社, 1988;（原著: A. Salomaa, “Computation and Automata,” Cambridge University Press, 1985）
6) 有川節夫, 宮野悟, “オートマトンと計算可能性,” 培風館, 1986.

■6 群 - 2 編 - 2 章

2-3 非決定性有限オートマトン

（執筆者：河原康雄・溝口佳寛）[2009 年 1 月 受領]

2-3-1 非決定性有限オートマトン

非決定性有限オートマトン（nondeterministic finite automaton）とは，状態の有限集合 Q，アルファベット Σ，状態遷移関数 $\delta : 2^Q \times \Sigma \to Q$，初期状態集合 $Q_0 \subset Q$，最終状態 $F \subset Q$ の五つ組 $M = (Q, \Sigma, \delta, Q_0, F)$ のことである．ここで，2^Q は Q の部分集合全体の集合であり，$2^Q = \{S \mid S \subset Q\}$ である．

状態遷移関数 $\delta : Q \times \Sigma \to 2^Q$ は $\delta^*(Q, \varepsilon) = Q$，$\delta^*(Q, aw) = \delta^*(\cup_{q \in Q} \delta(q, a), w)$ $(a \in \Sigma, w \in \Sigma^*)$ で定義し，$\delta^* : 2^Q \times \Sigma^* \to 2^Q$ に自然に拡張できる．語 w は，$\delta^*(Q_0, w) \cap F \neq \phi$ のとき，M に受理されるといい，M によって受理される語全体の集合（受理言語）を $L(M)$ で表す．すなわち，$L(M) = \{w \in \Sigma^* | \delta^*(Q_0, x) \cap F \neq \phi\}$ である．

例 1: 非決定性有限オートマトン $M = (K, \Sigma, \delta, Q_0, F)$ を $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$, $Q_0 = \{q_0\}$ とし，$\delta : Q \times \Sigma \to 2^Q$ を以下で定義する $[\delta(q_0, a) = \{q_0\}$, $\delta(q_0, b) = \{q_0, q_1\}$, $\delta(q_1, a) = \phi$, $\delta(q_1, b) = \{q_2\}$, $\delta(q_2, a) = \phi$, $\delta(q_2, b) = \phi]$. 例えば, 語 $w = abb$ は，$\delta^*(Q_0, abb) = \delta^*(\{q_0\}, abb) = \delta^*(\delta(q_0, a), bb) = \delta^*(\{q_0\}, bb) = \delta^*(\delta(q_0, b), b) = \delta^*(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\}$, したがって，$\delta^*(Q_0, abb) \cap F = \{q_2\} \neq \phi$ より，語 w は M で受理される．すなわち, $w \in L(M)$ である．

非決定性有限オートマトン M の状態遷移図は以下の図 2･8 ように表される．辺が出ていない頂点があること，同じラベルをもった 2 本以上の辺があることが決定性有限オートマトンの場合とは異なる．

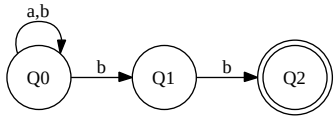


図 2･8

次の定理は，決定性有限オートマトンと非決定性有限オートマトンでは言語受容機械としては能力の差はないことを示している．

定理: 有限オートマトンについて次の (1)，(2) が成り立つ．**(1)** 決定性有限オートマトン $M_d = (Q, \Sigma, \delta_d, q_0, F)$ に対して，$M_n = (Q, \Sigma, \delta_n, \{q_0\}, F)$ によって定められる非決定性有限オートマトンの受理する語の集合は M_d のそれに等しい．すなわち，$L(M_d) = L(M_n)$ である．ここで, $\delta_n : Q \times \Sigma \to 2^Q$ は $\delta_n(q, a) = \{\delta_d(q, a)\}$ $(q \in Q, a \in \Sigma)$ とする．**(2)** 非決定性有限オートマトン $M_n = (Q, \Sigma, \delta_n, Q_0, F_n)$ に対して，$M_d = (2^Q, \Sigma, \delta_d, Q_0, F_d)$ によって定められる決定性有限オートマトンの受理する語の集合は M_n のそれに等しい．すなわち，$L(M_n) = L(M_d)$ である．ここで，$\delta_d : 2^Q \times \Sigma \to 2^Q$ は $\delta_d(S, a) = \delta_n^*(S, a)$ $(S \subset Q, a \in \Sigma)$ とし, $F_d = \{T \mid T \subset Q, T \cap F \neq \phi\}$ とする．

例 2: 例 1 の非決定性有限オートマトンと同じ言語を受理する決定性有限オートマトン $M_d = \{2^Q, \Sigma, \delta_d, \{q_0\}, F_d\}$ は次のようになる．$2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$, $F_d = \{\{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$, $\delta_d : 2^Q \times \Sigma \rightarrow 2^Q$ は，次のようになる $[\delta(\phi, a) = \phi, \delta(\phi, b) = \phi, \delta(\{q_0\}, a) = \{q_0\}, \delta(\{q_0\}, b) = \{q_0, q_1\}, \delta(\{q_1\}, a) = \phi, \delta(\{q_1\}, b) = \{q_2\}, \delta(\{q_2\}, a) = \phi, \delta(\{q_2\}, b) = \phi, \delta(\{q_0, q_1\}, a) = \{q_0\}, \delta(\{q_0, q_1\}, b) = \{q_0, q_1, q_2\}, \delta(\{q_0, q_2\}, a) = \{q_0\}, \delta(\{q_0, q_2\}, b) = \{q_0, q_1\}, \delta(\{q_1, q_2\}, a) = \phi, \delta(\{q_1, q_2\}, b) = \{q_2\}, \delta(\{q_0, q_1, q_2\}, a) = \{q_0\}, \delta(\{q_0, q_1, q_2\}, b) = \{q_0, q_1, q_2\}]$.

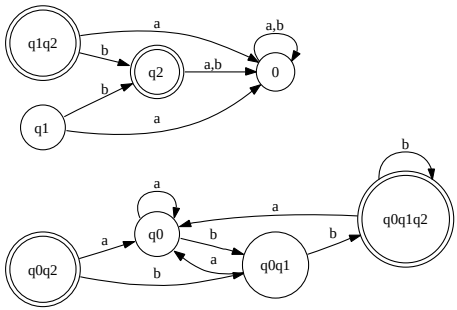


図 2･9

2-3-2 階層つきアルファベット, Σ-代数, 木

アルファベット Σ と写像 $\sigma : \Sigma \rightarrow N$（ただし，$N = \{0, 1, ..., \}$）の対 $< \Sigma, \sigma >$ を**階層つきアルファベット**という．$a \in \Sigma$ に対して，$\sigma(a)$ を a の階層という．以降，混乱がない限り，$< \Sigma, \sigma >$ を Σ と略記する．Q を空でない集合とし，$a \in \Sigma$ に対して，δ_a を Q 上の $\sigma(a)$ 項演算，すなわち，$\delta_a : Q^{\sigma(a)} \rightarrow Q$ とする．ただし，$\sigma(a) = 0$ のときは，$\delta_a \in Q$ とする．Q と δ の対 $< Q, \delta >$ を **Σ-代数**という．$n \in N$ に対し，$\Sigma_n = \{a \in \Sigma | \sigma(a) = n\}$ とし，$T_\Sigma = \Sigma_0 \cup \bigcup \{(T_\Sigma)^n a \,|\, a \in \Sigma_n, n \geq 1\}$ を満たす Σ^* の部分集合 T_Σ が唯一定まる．T_Σ の元 t を $< \Sigma, \sigma >$ 上の**木**と呼ぶ．t は各節点が Σ の元をラベルにもつ木であり，ラベル $a \in \Sigma_n$ をもつ節点は n 個の子をもち，$a \in \Sigma_0$ の節点が葉となる．$a \in \Sigma_0$ に対して，$\delta_{Ia} = a \in \Sigma^*$, $a \in \Sigma_n$ $(n \geq 1)$ を $\delta_{Ia}(t_1, \cdots, t_n) = t_1 \cdots t_n a$ $(t_1, \cdots, t_n \in T_\Sigma)$ とすると $< T_\Sigma, \delta_I >$ は, Σ-代数となり，**自由 Σ-代数**と呼ばれる．任意の Σ-代数 $< Q, \delta >$ に対して，$h_{<Q,\delta>} : T_\Sigma \rightarrow Q$ を $a \in \Sigma_0$ に対して，$h_{<Q,\delta>}(a) = \delta_a \in Q$, $a \in \Sigma_n$ $(n \geq 1)$ を $h_{<Q,\delta>}(t_1 \cdots t_n a) = \delta_a(h_{<Q,\delta>}(t_1), \cdots, h_{<Q,\delta>}(t_n))$ $(t_1, \cdots, t_n \in T_\Sigma)$ とすると，$< T_\Sigma, \delta_I >$ から $< Q, \delta >$ への構造を保存する唯一の写像 $h_{<Q,\delta>}$ が定まる．

2-3-3 木オートマトン

木オートマトン（tree automaton）は，J.W. Thatcher と J.B. Wright（1968）により提案され，W.S. Brainerd（1969）によって名づけられたものである．通常の有限オートマトンの入力は語（記号列）であり，文字 a が状態の遷移，すなわち，Q から Q への単項の演算を定め

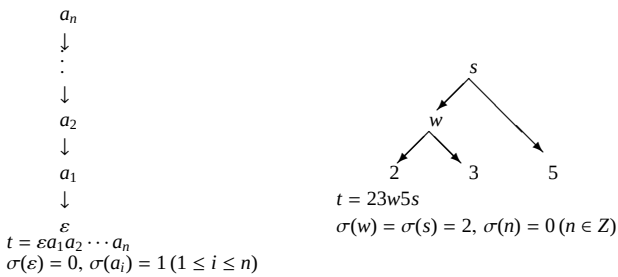


図 2·10

ると考えられる．木オートマトンでは，入力は木とする．ラベル f をもつ木の節点が n 個の子をもつとき，f は Q^n から Q への多項演算を定めると考える．語は子を一つもつ木と考えられるので，木オートマトンは通常のオートマトンの一般化である．

定義: 階層つきアルファベット $<\Sigma, \sigma>$ 上の**決定性木オートマトン**とは，状態集合と状態遷移関数を定める Σ-代数 $<Q, \delta>$ と最終状態 $F \subset Q$ の対 $M = <<Q, \delta>, F>$ のことである．木 $t \in T_\Sigma$ は, $h_{<Q,\delta>}(t) \in F$ のとき，かつ，そのときに限り M で受理されるといい，受理される木全体を $T(M) = \{t \in T_\Sigma \mid h_{<Q,\delta>}(t) \in F\}$ で表す．

例 1: 文字を階層 1 のアルファベット，空文字列 ε を階層 0 のアルファベットとすると文字列を木と考えることができる．$M_A = (Q, \Sigma_A, \delta_A, q_0, F)$ を決定性有限オートマトンとする．$\Sigma = \{\varepsilon\} \cup \Sigma_A$, $\sigma(\varepsilon) = 0$, $\sigma(x) = 1$ $(x \in \Sigma_A)$ とし，階層つきアルファベット $<\Sigma, \sigma>$ を定める．$\delta_\varepsilon = q_0 \in Q$, $\delta_x(q) = \delta_A(q, x)$ $(x \in \Sigma_A,\ q \in Q)$ とすると，$<Q, \delta>$ は Σ-代数となる．語 $w = a_1 a_2 \cdots a_n \in \Sigma_A$ に対して，木 $t = \varepsilon a_1 a_2 \cdots a_n \in T_\Sigma$ を対応させると，$\delta^*(q_0, w) = h_{<Q,\delta>}(t)$ であり，決定性有限オートマトン M_A の受理集合と木オートマトン $<<Q, \delta>, F>$ の受理集合が対応する．

例 2: n を自然数とするとき，$Z_n = \{0, 1, \cdots, n-1\}$ とし, $x, y \in Z_n$ に対して，$x + y \in Z_n$ を x と y の和を n で割った余り，$x * y \in Z_n$ を x と y の積を n で割った余りとする．$\Sigma = Z \cup \{w, s\}$, $\sigma(n) = 0$ $(n \in Z)$, $\sigma(w) = 2$, $\sigma(s) = 2$ とし，階層つきアルファベット $<\Sigma, \sigma>$ を定める．$\delta_n = n \in Z$ $(n \in Z)$, $\delta_w : Z^2 \times Z$ を $\delta_w(x, y) = x + y$, $\delta_s : Z^2 \times Z$ を $\delta_s(x, y) = x * y$ で Σ-代数 $<Z_n, \delta>$ を定める．T_Σ は和と積の数式全体を表し，$t \in T_\Sigma$ に対して，$h_{<Z_n,\delta>}(t)$ は数式を計算した値となる．例えば，$h_{<Z_n,\delta>}(23w5s) = 30$ である．

通常の有限オートマトンと同様に非決定性の木オートマトンも定義でき，更に，類似の結果が木オートマトンにも自然に導かれる．

定理: 次の命題 (1), (2), (3), (4) が成り立つ．**(1)** 非決定性木オートマトンにより受理される集合は，決定性木オートマトンでも受理される．**(2)** 木オートマトンで受理される集合，

$S \subset T_\Sigma$, $T \subset T_\Sigma$ に対して，$S \cup T$, $S \cap T$, $T_\Sigma - T$ も受理される．**(3)** 木オートマトン M に対し，$T(M) = \phi$ か否かは決定可能である．**(4)** 二つの木オートマトン M_1, M_2 に対して，$T(M_1) = T(M_2)$ か否かは決定可能である．

■参考文献

1) M.A. Arbib and E.G. Manes, “Arrows, structures, and functors,” Academic Press, 1975.
2) Comon et al., “Tree Automata Techniques and Applications,” Available on http://www.grappa.univ-lille3.fr/tata, 2007.
3) H. Ehrig, K.D. Kiermeir, H.J. Kreowski, and W. K”uhnel, “Universal Theory of Automata,” B.G.Teubner, 1974.
4) F. Gécseg and M. Steinby, “Tree Automata,” Budapest, Akademiai Kiado, 1984.
5) W.C. Rounds, “Mappings and grammars on trees,” Mathematical Systems Theory, vol.4, pp.257-287, 1970.
6) J.W. Thatcher and J.B. Wright, “Generalized finite automata theory with an application to a decision problem of second order logic,” Mathematical Systems Theory, vol.2, pp.57-81, 1968.
7) 小林孝次郎, 高橋正子, “オートマトンの理論,” 共立出版, 1983.
8) 藤野精一 編集, “計算数学ハンドブック,” 朝倉書店, 1977.

■6 群 - 2 編 - 2 章

2-4 正則表現と正則言語

（執筆者：山本博章）[2008 年 12 月 受領]

正則表現（正規表現: regular expression）は，記号列の集合を簡潔に記述するための手法で，プログラミング言語，パターン照合などコンピュータサイエンスの多くの分野への応用をもつ．本節では，正則表現，正則言語，正則文法，及びそれらと有限オートマトンとの関連について述べる．

2-4-1 正則表現の定義

今，Σ をアルファベットとする．そのとき，Σ 上の正則表現は以下のように，三つの演算，和集合，連接，閉包（Kleene 閉包）を用いて，再帰的に定義される．

1. $\emptyset$，ϵ は正則表現である．これらはそれぞれ，空集合 ($\emptyset$)，集合 $\{\epsilon\}$ を表す．ここで，ϵ は空記号列を表す．

2. 任意の $a \in \Sigma$ に対し，**a** は正則表現である．これは，集合 $\{a\}$ を表す．通常，混乱の恐れがなければ，**a** と a は区別しない．

3. r_1 及び r_2 は正則表現で，それぞれ集合 R_1 及び R_2 を表すとする．そのとき，$(r_1 + r_2)$，$(r_1 r_2)$，(r_1^*) も正則表現である．ここで，$r_1 + r_2$ は集合 $R_1 \cup R_2$（和集合），$r_1 r_2$ は集合 $R_1 R_2$（連接），r_1^* は集合 R_1^*（閉包）を表す．

正則表現に現れる演算に，強い方から，閉包，連接，和集合という順で優先順位を導入すると，括弧を省略することができる．正則表現 r が表す言語は $L(r)$ と表される．一般に，正則表現が表す言語は正則言語と呼ばれる．正則表現に共通集合演算，補集合演算を導入して拡張したものを拡張正則表現と呼ぶ．正則言語は共通集合演算，補集合演算に関して閉じているため，拡張正則表現が表す言語もまた正則言語となる．

2-4-2 正則表現から有限オートマトンへの変換

正則表現と有限オートマトンの間には等価な関係がある．すなわち，以下の定理が成り立つ．

定理 2–4.1 r を Σ 上の正則表現とする．そのとき，$L(r)$ を受理する有限オートマトンを構成することができる．逆に，言語 L が有限オートマトンによって受理されるなら，L を正則表現で表すことができる．

応用の観点から，正則表現から有限オートマトンを作成するためのアルゴリズムが活発に研究されている．目標は，できるだけ高速に，かつサイズの小さいオートマトンを作ることである．正則表現から得られる有限オートマトンとして最もよく知られたものは Thompson automaton（TA）と呼ばれるものである．このオートマトンは ϵ 動作をもつ非決定性有限オートマトンで，正則表現の再帰的な定義に従って構成することができる．各演算に対する TA の構成法を図 2･11 に与える．図 2･11 において，M_1, M_2 はそれぞれ正則表現 r_1, r_2 に対するオートマトンを表す．そのとき，(a) は和集合，(b) は連接，(c) は閉包に対する構成法を示

している[2]．構成法から明らかなように，TA はたかだか $2m$ 個の状態と高々 $4m$ 個の状態遷移をもつ．ここで，m は正則表現の長さを表す．

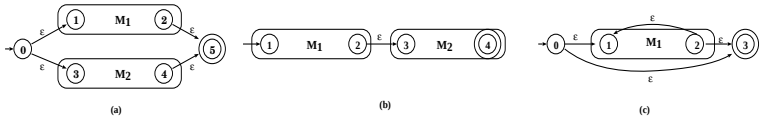


図 2·11　正則表現から有限オートマトンへの変換

もう一つ以前から知られている有限オートマトンに position automaton（PA, これは Glushkov automaton とも呼ばれる）と呼ばれるものがある．このオートマトンは ϵ 動作のない非決定性有限オートマトンであり，$s+1$ 個の状態をもつ．ここで，s は正則表現に出現する Σ の記号の数である．状態数を見ると TA より少ないが，状態遷移数が最悪 s^2+s になる．正則表現 r に対し，PA $M=(Q,\Sigma,\delta,q_0,F)$ は次のように定義される．まず，正則表現 r に出現する記号 a に対し，その記号が i 番目の位置に出現するならば，a に番号 i を割り振り，a_i とする．正則表現 r に出現するすべての記号に番号を割り振った結果を $\overline{r}$ と定義し，I をこのような位置を示す番号の集合とする．例えば，$r=(ab+a)b^*$ ならば，$\overline{r}=(a_1b_2+a_3)b_4^*$ となり，$I=\{1,2,3,4\}$ となる．このとき，**First**，**Last**，**Follow**(i) の 3 種類の I の部分集合をそれぞれ以下のように定義する．ここで，α，β は，$\overline{r}$ に出現する番号つきの記号からなる記号列を表す．

- **First** $=\{i \mid a_i\alpha \in L(\overline{r})\}$,
- **Last** $=\{i \mid \alpha a_i \in L(\overline{r})\}$,
- すべての $i \in I$ に対し，**Follow**$(i)=\{j \mid \alpha a_i a_j \beta \in L(\overline{r})\}$.

これらの定義のもと，PA M の状態の集合 Q を $Q=I\cup\{q_0\}$ と定義し，最終状態の集合 F を $F=$ **Last** と定義する．状態遷移関数 δ は次のように定義される．初期状態 q_0 に対しては，$i \in$ **First** でかつ i 番目の記号が a のとき，かつそのときに限り $i \in \delta(q_0,a)$ と定義する．同様に，任意の $i \in I$ に対し，$j \in$ **Follow**(i) でかつ j 番目の記号が a のとき，かつそのときに限り $j \in \delta(i,a)$ と定義する．このとき，正則表現から PA を生成するための $O(s^2)$ 時間アルゴリズムが知られている．後の例で示すように，PA は TA から直接構成することもできる．

最近，よりサイズの小さなオートマトンとして，follow automaton（FA）と呼ばれるモデルが提案された[3]．これは，PA の状態間に次の等価性を定義することによって得られる．$M=(Q,\Sigma,\delta,q_0,F)$ を PA とする．そのとき，任意の二つの状態 $q,p \in Q$ が等価であるとは，$q,p \in F$ かまたは $q,p \notin F$ であり，かつすべての記号 $a \in \Sigma$ に対し，$\delta(q,a)=\delta(p,a)$ が成り立つことである．等価な状態はまとめることができ，その結果得られる有限オートマトンを FA と呼ぶ．明らかに，FA の状態数はたかだか $s+1$ で，状態遷移数はたかだか s^2+s である．FA についても $O(s^2)$ 時間で正則表現から構成できる．

TA，PA 及び FA の三つのオートマトンの例を示す．正則表現 $b((aa)^*+(bb)^*)^*b$ を考える．そのとき，図 2·12 は対応する Thompson automaton を示す．このオートマトンから状態

0, 1, 6, 8, 12, 14, 19 だけを取り出して作ったものが図 2･13 の position automaton である．更に，PA において，状態 1,8,14 が等価となり，これらの状態まとめたものが図 2･13 の follow automaton である．

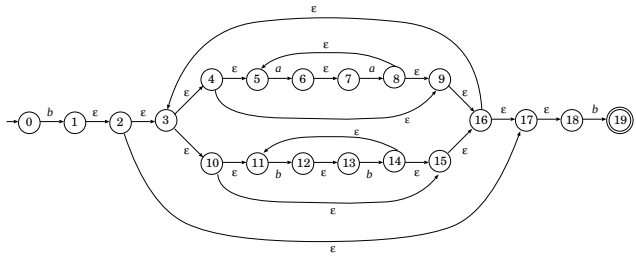


図 2･12　Thompson automaton

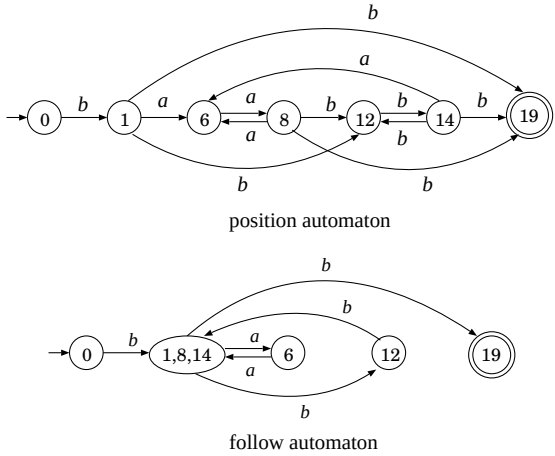


図 2･13　position automaton と follow automaton

2-4-3 正則表現の所属問題

正則表現の所属問題とは，任意に与えられた正則表現 r と記号列 x に対し，$x \in L(r)$ かどうかを判定する問題である．この問題はパターン照合問題を解くために利用することができるため，効率的なアルゴリズムの開発が活発に行われてきた．既に見たように，正則表現は効率的に非決定性有限オートマトンに変換できる．従って，この問題を解くためのアルゴリズムは，通常，正則表現を有限オートマトンに変換し，得られたオートマトンを模倣するように設計される．今，正則表現 r の長さを m，記号列 x の長さを n とする．そのとき，所属問題を解くアルゴリズムについて以下の結果が成り立つ．

1. TA を用いるならば，所属問題を $O(mn)$ 時間及び $O(m)$ 領域で解くことができる．

2. PA，FA を用いるならば，所属問題を $O(s^2n)$ 時間，$O(s^2)$ 領域で解くことができる．

上記は最悪の場合であり，実際の実行時間は得られたオートマトンのサイズに依存する．従って，PA や FA ではかなりサイズの小さいオートマトンになることもあり，そのときは高速に動作する．なお，現在知られているアルゴリズムのなかで最悪の実行時間が最も良いものは，Myers[4]によって与えられた $O(mn/\log n)$ 時間アルゴリズムである．

決定性有限オートマトンを利用したアルゴリズムに関しても研究されている．決定性有限オートマトンは，次の遷移が一意に決まるため，オートマトンが与えられれば，$O(n)$ 時間で所属問題を解くことができる．しかし，決定性有限オートマトンにおける最も大きな問題は，その状態数が正則表現の長さに対し指数的に増大することである．そのため，できるだけコンパクトな決定性有限オートマトンを生成する手法に関する研究が行われている．例えば，Navarro[5]らは PA からコンパクトな決定性有限オートマトンの表現方法を与えた．彼らは，PA の特別な性質，すなわち，任意の状態に対し，その状態に入る遷移はすべて同じ記号によるという性質，を利用した．山本[6]は，dual position automaton （dual-PA）を用いた決定性有限オートマトンのコンパクト表現と所属問題のためのアルゴリズムを与えた．Dual-PA は，PA とは逆に，任意の状態に対し，その状態から出ていく遷移がすべて同じ記号による．このオートマトンもまた TA から直接構成することができる．

2-4-4 正則文法と正則言語

形式文法 G は，$G = (V, T, P, S)$ の 4 項組で定義される．ここで，V は非終端記号の集合，T は終端記号の集合で $V \cap T = \emptyset$，P は書き換え規則（生成規則）の集合，$S \in V$ は開始記号である．正則文法とは，書換え規則がすべて，$A \to aB$ または $A \to a$ のかたちをした形式文法である．ここで，$A, B \in V$，$a \in T$ である．この定義だけでは，空記号列 ϵ を生成できないため，開始記号についてのみ $S \to \epsilon$ を許す．形式文法が生成する言語 $L(G)$ は，開始記号 S から出発し，書換え規則を任意の回数適用することによって得られる終端記号列の集合である．すなわち，$L(G) = \{x \in T^* \mid S \overset{*}{\to} x\}$ と定義される．このとき，有限オートマトンと等価な関係として，以下の結果が成り立つ．

定理 2–4.2 任意の正則文法 G に対し，$L(G)$ を受理する非決定性有限オートマトンを構成することができる．逆に，任意の決定性有限オートマトン M に対し，M が受理する言語を生成する正則文法を構成することができる．

このように，正則文法が生成する言語もまた正則言語である．正則文法の書き換え規則を少し拡張し，すべての規則が，$A \to wB$ または $A \to w$ のかたちをした文法を考える．ここで，$A, B \in V$，$w \in T^*$ である．このような文法は右線形文法（right-linear grammar）と呼ばれる．同様に，すべての規則が，$A \to Bw$ または $A \to w$ のかたちをした文法は，左線形文法（left-linear grammar）と呼ばれる．そのとき，以下の定理が成り立つ．この定理から，右線形文法や左線形文法まで拡張したものを正則文法と定義する場合もある[1]．

定理 2–4.3 右線形文法や左線形文法によって生成される言語は正則言語である．

■参考文献

1) D. Du and K. Ko, “Problem Solving in Automata, Languages, and Complexity,” John Wiley & Sons, Inc., 2001.
2) J.E. Hopcroft and J.D. Ullman, “Introduction to Automata theory, Languages, and Computation,” Addison Wesley, Reading Mass, 1979.
3) L. Ilie and S. Yu, “Follow automata,” Information and Computation, vol.186, pp.140-162, 2003.
4) G. Myers, “A Four Russians Algorithm for Regular Expression Pattern Matching,” J. Assoc. Comput. Mach., vol.39, no.4, pp.430-448, 1992.
5) G. Navarro and M. Raffinot, “New Techniques for Regular Expression Searching,” Algorithmica, vol.41, pp.89-116, 2004.
6) H. Yamamoto, “Regular Expression Matching Algorithms using Dual Position Automata,” Proc. of 18th International Workshop on Combinatorial Algorithms, College Publications, pp.212-225, 2007.

■6 群 - 2 編 - 2 章

2-5 量子オートマトン

（執筆者：中西正樹）[2008 年 12 月 受領]

量子オートマトンは量子計算機のシンプルなモデルであり，使用できる計算資源の違いにより，様々なものが提案されている．本節では，そのなかでも量子有限オートマトンに焦点を絞り，説明する．

2-5-1 量子有限オートマトンの定義

量子有限オートマトン[2, 3]は確率有限オートマトンを量子計算モデルに拡張したものであり，量子計算機のモデルのなかで最もシンプルなものの一つである．ヘッドが左右 2 方向に動くことができる 2 方向量子有限オートマトンの定義は以下のとおりである．2 方向量子有限オートマトンは次の 6 項組で定められる：$M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$. ここで，$Q, \Sigma, q_0, Q_{acc}, Q_{rej}$ はそれぞれ, 状態の有限集合, 有限の入力アルファベット, 初期状態, 受理状態の集合, 非受理状態の集合であり, $Q_{acc} \subset Q$, $Q_{rej} \subset Q$, $Q_{acc} \cap Q_{rej} = \emptyset$ である．確率有限オートマトンでは, 状態遷移に「確率」が割り当てられているが, 量子有限オートマトンでは, 状態遷移に量子計算の特徴である「確率振幅」が割り当てられており, 状態遷移関数は $\delta : Q \times \Sigma \cup \{\#, \$\} \times Q \times \{-1, 0, +1\} \longrightarrow \mathbb{C}$ として与えられる．ここで，# と $ は Σ に含まれない終端記号である．確率有限オートマトンと比べると，状態遷移関数の値域が非負実数から複素数に拡張されていることが分かる．

$\delta(q, a, q', D) = \alpha$ は状態 q で入力記号 a を読んだときに状態 q' に遷移して入力ヘッドを D (-1:左，0:移動せず，+1:右）の方向に移動する確率振幅が α であることを表す．2 方向量子有限オートマトンの時点表示は (q, k)（q : 状態，k : ヘッド位置）で表される．ここで， 各時点表示に対して，次のような列ベクトル $|q, k\rangle$ への対応づけを考える．

- $|q, k\rangle$ は $n|Q|$ 行 1 列の列ベクトルである（ただし，n は入力長）.
- (q, k) に対応する行が 1，それ以外の行は 0 である.

$\{|q, k\rangle \,|\, q \in Q, k \in \{1, \ldots, n\}\}$ で張られるヒルベルト空間の任意のノルム 1 の要素のことを時点表示の重ね合せと呼び，量子計算モデルはその特徴として，時点表示の重ね合せを保持することができる．また，状態遷移関数 δ は量子力学に起因する制約上，以下の性質を満たさなければならない．ここで，入力 $\boldsymbol{x}$ に対し，次のように状態遷移関数を行列 $U^{\boldsymbol{x}}$（時間発展行列と呼ばれる）を用いて表現することを考える．

$$U^{\boldsymbol{x}}(|q, k\rangle) = \Sigma_{q' \in Q, D \in \{-1,0,1\}} \delta(q, x(k), q', D) \left|q', k + D\right\rangle,$$

ただし， $x(k)$ は入力 $\boldsymbol{x}$ における k 番目の入力記号である．このとき，$U^{\boldsymbol{x}}$ はユニタリ行列でなければならない．つまり，$U^{\boldsymbol{x}} U^{\boldsymbol{x}\dagger} = U^{\boldsymbol{x}\dagger} U^{\boldsymbol{x}} = I$ である．ここで，$U^{\boldsymbol{x}\dagger}$ は $U^{\boldsymbol{x}}$ の共役転置行列である.

2 方向量子有限オートマトンの動作を説明する前に, いくつかの表記を定義しておく．$|\Psi_0\rangle = |q_0, 0\rangle$ とする．また, $\{|q, k\rangle \,|\, q \in Q, k \in \{1, \ldots, n\}\}$ で張られる空間を次のような $E_{acc}, E_{rej}, E_{non}$ の三つの部分空間に分割する：$E_{acc} = span\{|q, k\rangle \,|\, q \in Q_{acc}\}$, $E_{rej} = span\left\{|q, k\rangle \,|\, q \in Q_{rej}\right\}$,

$E_{non} = span\left\{|q,k\rangle | q \in Q \setminus (Q_{acc} \cup Q_{rej})\right\}$．更に，$\{E_{acc}, E_{rej}, E_{non}\}$ に対する射影測定による出力をそれぞれ「acc」，「rej」，「non」と定める．これらを基に，以下，2 方向量子有限オートマトンの動作を説明する．

1. $U^{\boldsymbol{x}}$ を $|\Psi_i\rangle$ に適用し，$|\Psi_{i+1}\rangle = U^{\boldsymbol{x}} |\Psi_i\rangle$ とする．
2. $|\Psi_{i+1}\rangle$ に対して，$\{E_{acc}, E_{rej}, E_{non}\}$ への射影測定を行う．$\left|\phi_j\right\rangle$ を $|\Psi_{i+1}\rangle$ の E_j への射影とする．ここで，j は「acc」，「rej」，「non」のいずれかである．量子力学に従うと，測定の結果，各出力 j を確率 $\left\|\left|\phi_j\right\rangle\right\|^2$ で得る．また，測定の結果が j であった場合，量子状態 $|\Psi_{i+1}\rangle$ は $\frac{1}{\||\phi_j\rangle\|}\left|\phi_j\right\rangle$ へと収縮する．

上記 1. 及び 2. を「acc」もしくは「rej」が出力されるまで繰り返す．

2-5-2 量子有限オートマトンが認識する言語のクラス

ヘッドの動く方向を 1 方向に限定したものを 1 方向量子有限オートマトンと呼ぶ．古典有限オートマトンでは 1 方向，2 方向，また，遷移の決定性，非決定性を問わず，認識できる言語のクラスは正則言語であることが知られている（更には，多項式時間 2 方向確率有限オートマトンが認識できる言語のクラスも正則言語である）．一方，量子有限オートマトンの場合は，1 方向モデルと 2 方向モデルで能力が大きく変わることが知られている[2)]．具体的には，1 方向モデルでは認識できる言語のクラスが正則言語よりも真に小さく，2 方向モデルでは真に大きい（図 2･14 参照）．1 方向モデルでは，量子計算モデルの方が対応する古典計算モデルよりも能力が弱くなるというのは興味深い結果である．これは，量子計算モデルは可逆でなければならない，つまり，上述のように状態遷移関数がユニタリ行列（定義から，必ず逆行列をもつ）で記述できなければならないという量子力学上の制約に起因するものである．

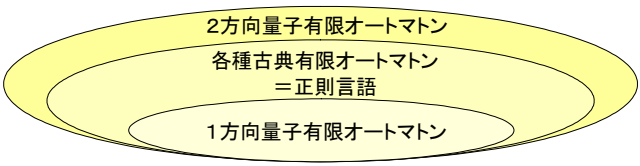


図 2･14　有限オートマトンが認識する言語のクラスの包含関係

2-5-3 量子有限オートマトンの実装に必要な計算資源

ところで，2 方向量子有限オートマトンの時点表示が (q,k) で与えられること，すなわち，時点表示にヘッド位置が含まれることを考えると，このモデルを実装するのに必要な量子ビット数は入力長に依存することになる．なお，1 方向量子有限オートマトンでは，常にヘッドが右に動くことから時点表示においてヘッド位置を省略することができるため，こちらは実装する際に必要な量子ビット数は入力長に依存しない．そこで，実装する際に必要な量子ビット数が入力長に依存しない 2 方向量子有限オートマトンとして，古典ヘッド，古典有限状態制御部及び量子有限状態制御部をもつ有限オートマトンが提案されている[1)]．このモデルで

は，正則言語のほかに，$\{0^n1^n|n \in \mathbb{N}\}$ といった言語や，回文（$\{w \in \{0,1\}^*|w = w^R\}$, w^R は w を反転した語）を認識できることが知られている．このことは，定数個の量子ビットを用いて実装できる計算モデルにおいても，量子計算モデルに優位性があることを示している．

■参考文献

1) A. Ambainis and J. Watrous, “Two-way finite automata with quantum and classical states,” Theoretical Comput. Sci., vol.287, no.1, pp.299-311, 2002.
2) A. Kondacs and J. Watrous, “On the power of quantum finite state automata,” Proc. of 38th Symp. on Foundations of Computer Science, pp.66-75, 1997.
3) C. Moore and J.P. Crutchfield, “Quantum automata and quantum grammars,” Theoretical Comput. Sci., vol.237, no.1-2, pp.275-306, 2000.