数理解析研究所講究録
第1599巻 2008年 9-16

DFAの極限学習における必要例数の解析

林 賢史
東京工業大学 数理計算科学専攻
e-mail: hayashi3@is.titech.ac.jp

1 はじめに

学習とは,「与えられた個々の事実から一般的な法則を導き出す」ことを一つの目標とする(以下個々の事実を例とよぶ).

この学習の枠組みの一つである極限学習とは,「一般的な例からの学習」が無限に続く過程であることに着目し,学習のもつ数学的な性質を明らかにするために,Gold[Gold67] によって提案された学習の枠組みである.

本研究では,学習対象をDFAとしたDFAの極限学習を考え,DFAの極限学習に必要な例の個数について解析した(以下では学習の対象となったDFAを目標DFAとよぶ).

DFAを学習対象とするので,文字列とその文字列に対する目標DFAでの受理非受理の判定結果との対を例として学習する.

本研究では,DFAの極限学習アルゴリズムの入力に,どのような例の集合を含んでいれば,目標オートマトンを必ず出力するかを考える.以下では,その集合を十分例集合とよぶ.

与えられた目標DFAと,極限学習アルゴリズムの組に対し,最小な十分例集合を考えることによって,このアルゴリズムが目標DFAを学習するために必要な例数の下界を与える.このため最小十分例集合の大きさを極限学習アルゴリズムの性能を表す一つの指標とすることができる.

本研究の結果として,次のことを示した.任意のDFAの極限学習アルゴリズムに対し,ある状態数 n 以下の目標DFAが存在し,次の式を満たす.

$$\left|\begin{array}{l}\text{極限学習アルゴリズムと}\\ \text{目標DFAの組に対する十分例集合}\end{array}\right| = \Omega(n\log n)$$

このことはDFAを極限学習することの難さのひとつの下界を示したことになる.

2 準備と本研究の結果

機械論的学習理論の研究の一つの柱である,Goldによって提案された,極限学習において,DFAを学習目標にした,DFAの極限学習について定義する.

まず,簡単のため,列 ρ に対する,部分有限列 $\rho[k](k=1,2,\ldots)$ を次のように定義する.

$$\rho[k] = \rho(1),\rho(2),\cdots,\rho(k)$$
(ただし,$\rho(i)$ はρの i 番目の要素)

さらに列 ρ,ρ' に対し,記号 $\in,\subset,\preceq$ を次のように定義する.

$a \in \rho \Leftrightarrow \rho$の要素に a が存在する.
$\rho' \subset \rho \Leftrightarrow \rho'$の任意の要素が$\rho$の要素となっている.
$\rho' \preceq \rho \Leftrightarrow$ ある k が存在し$\rho' = \rho[k]$ となる.

次にDFAの機械学習の定義で用いる,正規言語 L に対する例 e と完全提示 σ とDFAの学習機械 M を先に定義する.

定義 1. 対 $e=(w,l) \in \Sigma^* \times \{0,1\}$ を考える.言語 L に対し,以下の条件を満たすとき,対 e が L の例であるとする.

$$l = L(w)$$

ただし,言語 L を $\Sigma^* \to \{0,1\}$ の関数とも考え,

$$L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$$

とする.さらに例 $e=(w,L(w))$ が $L(w)=1$ のとき,e を正例とよび,$L(w)=0$ のとき,e を負例とよぶ.

無限列 $\sigma=(w_1.l_1)(w_2,l_2)(w_3,l_3)\cdots$ を考える.ただし,$w_i \in \Sigma^*(i=1,2,\ldots), l_i \in \{0,1\}(i=1,2,\ldots)$ とする.言語 L に対し,無限列 σ が以下の条件を満たすとき,σ は完全提示であるとする.
(1) $\{w_1,w_2,\cdots\} = \Sigma^*$

(2) すべての $i \geq 1$ に対し，$l_i = L(w_i)$

無限列 σ を入力とし，DFA の無限列 $\gamma = g_1.g_2,\ldots$ を仮説とし出力する機械を DFA に対する学習機械 M とする．ただし，$\sigma(k)$ を最後入力としたとき，学習機械 M は $\gamma(k)$ を出力する．

以上の定義を用いて，DFA の極限学習を定義する．

定義 2. 正規言語 L を識別する DFA A_* を目標 DFA とする．学習機械 M が L に対する任意の完全提示 σ を入力とし，M の出力 γ が次の式を満たすとき，M は A_* を極限学習すると定める．

$$\lim_{k\to\infty} \gamma(k) = A_*$$

さらに，学習機械 M が極限学習するとき，M を極限学習アルゴリズムとよび，保守的な極限学習アルゴリズムであるとは，M が極限学習するさい，出力 γ において以下の制限があったときをいう．

$$\sigma(k) \text{ が} \gamma(k-1) \text{ に無矛盾} \Rightarrow \gamma(k) = \gamma(k-1)$$

次に本研究で用いる十分例集合を定義する．
その前に十分例集合の定義で用いる記号 ALG,REG_n を定義する．

ALG = {DFA の保守的な極限学習アルゴリズムの集合 }
REG_n = { 状態数 n 以下の DFA で識別できる言語の集合 }

目標 DFA A_* を識別する言語 L と極限学習アルゴリズム M_* に対する例のある集合を極限学習アルゴリズム M_* に入力として与えれば，必ず目標 DFA A_* を出力するとき，ある集合を言語 L と M_* に対する十分例集合とよぶ．

以下では十分例集合の集合 $S_{M,L}$，十分例集合 $s_{M,L}$,s_{M,REG_n},s_{*,REG_n} を形式的に定義する．

定義 3. $M \in ALG, L \in REG$ に対して，十分例集合の集合 $S_{M,L}$，十分例集合 $s_{M,L}$,s_{M,REG_n},s_{*,REG_n} を以下のように定義する．

M の入力の列が集合 $S_{M,L}$ の要素 s を含んでいたならば必ず M は DFA A s.t. $\mathcal{L}(A) = L$ を出力する．つまり，形式的には以下のようになる．

$$S_{M,L} = \left\{ s_+ \cup s_- \;\middle|\; \begin{array}{l} s_+ \subset \forall s'_+ \subset 2^{L\times\{1\}}, \\ s_- \subset \forall s'_- \subset 2^{\Sigma^*-L\times\{0\}}, \\ \forall\varphi \text{ は } s'_+, s'_- \text{を含む有限列} \end{array} \left[\mathcal{L}(M(\varphi)) = L \right] \right\}$$

定義した $S_{M,L}$ において一番小さい要素を $s_{M,L}$ とする．つまり

$$s_{M,L} = \arg \min_{s\in S_{M,L}} |s|$$

となる．状態数 n 以下 で表すことができる正規言語 REG_n の中で，$s_{M,L}$ の値が最大となる L に対する十分例集合を s_{M,REG_n} とする．つまり，

$$s_{M,REG_n} = \arg \max_{L\in REG_n} |s_{M,L}|$$

となる．ALG のなかで $|s_{M,REG_n}|$ が最小となる M に対する，十分例集合を s_{*,REG_n} とする．つまり，

$$s_{*,REG_n} = \arg \min_{M\in ALG} |s_{M,REG_n}|$$

となる．

以上の定義より，任意のアルゴリズムに対し，正規言語のある言語 L で，その十分例集合の大きさは $|s_{*,REG_n}|$ 以上となる．

以上の定義を用い，本研究の結果として，以下の定理示した．

定理 1.

$$|s_{*,REG_n}| = \Omega(n\log n)$$

つまりこの定理により，任意のアルゴリズムに対し，正規言語のある言語 L で，その十分例集合の大きさは下界 $\Omega(n\log n)$ をもつ，ことを意味する．
次節でこの定理 1 の証明を記す．

3 定理 1 の証明

定理 1 の証明の方針は以下のようになる．

1. 正規言語の部分集合で，ある特徴をもった言語集合 $\mathcal{Y}$ を定義する．
2. $\mathcal{Y}$ がある特徴をもっているので，どのような $M \in ALG$ に対しても $\mathcal{Y}$ の要素に対する十分例集合が満たさなくてはならない条件が存在する．その条件を元に，$\mathcal{Y}$ の要素に対し，大きさが k 以下の十分例集合になりえる集合の数を考える．
3. $|\mathcal{Y}| \geq$ 十分例集合になりえる集合の種類
であることから，$\mathcal{Y}$ に含まれる，ある言語に対する，十分例集合の大きさの下界を示す．$\mathcal{Y}$ は正規言語の部分集合であるので，これは求める下界となる．

よって，始めに正規言語の部分集合 $\mathcal{Y}$ を定義する．以下この節では，M_* を

$$s_{*,REG_n} = s_{M*,REG_n}$$

となるアルゴリズムとする．

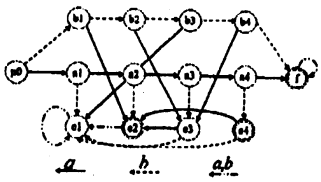

図 1: Λ_v の例

上記（図1）のようなオートマトンの集合を考え，そのオートマトン集合で表現できる言語の集合を $\mathcal{Y}$ とする．

次にこのオートマトンの集合を定義するために必要になる，文字列の集合 E，ならびに l_E 次元のベクトル $row(q_{ck})$,$v_a(row(q_{ck}))$,$v_b(row(q_{ck}))$ を以下で定義する．

その前に文字列の集合 S に対して，関数 N_S : $\{1,2,\ldots,|S|\} \to S$ を次のように定義する．

$$N_S(a) = \left(\begin{array}{l} \text{集合 } S \text{ の要素を長さ優先辞書式順序で} \\ \text{小さい順に並べたときの } a \text{ 番目の文字列} \end{array} \right)$$

定義 4. $w \in \Sigma^*$ に対し，文字列の集合 $E(l_E = |E|$ とする)，ならびに l_E 次元のベクトル $row(q_{ck})$,$v_a(row(q_{ck}))$, $v_b(row(q_{ck}))$(ただし，$k \in \{1,2,\ldots,m\}^m (m = 2^{l_E})$) を以下のように定義する(直感な理解を深めるためには，図2参照)．

- $E = \{\lambda, a, b, aa, ab, \cdots, w_*|w_*$ と長さ優先辞書式順序で w_* より前の文字列 $\}$
- $row(q_{ck}) = (b(k, N_E(1)), b(k, N_E(2)), b(k, N_E(3)), b(k, N_E(4)), \cdots, b(k, N_E(l_E))$

  ただし，

$$b(k, N_E(i)) = \begin{cases} 1 \le i \le \lfloor \log k \rfloor \\ \quad k-1 \text{ の 2 進数表記で下から } i \text{ ビット目の値} \\ \text{その他のとき} \\ \quad 0 \end{cases}$$

- $v_a(row(q_{ck})) = (b(k, N_{E_a}(1)), b(k, N_{E_a}(2)), \cdots, b(k, N_{E_a}(|E_a|)), 0, 0, \cdots, 0)$

  ただし，

$$E_a = \{w | w \in E[w \text{ の接頭語が } a]\}$$

- $v_b(row(q_{ck})) = (b(k, N_{E_b}(1)), b(k, N_{E_b}(2)), \cdots, b(k, N_{E_b}(|E_b|)), 0, 0, \cdots, 0)$

  ただし，

$$E_b = \{w | w \in E[w \text{ の接頭語が } b]\}$$

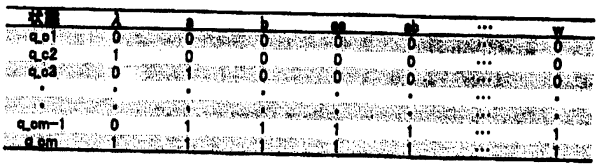

図 2:

例を用い l_E 次元ベクトル $row(q_{ck})$ を説明すると，

$row(q_{c3}) = (b(3, N_E(1)), b(3, N_E(2)), b(3, N_E(3)), \cdots, b(3, N_E(l_E)))$

は以下のようになる．

q_{c3} は $3-1=2$ を二進数表記すると 10 となるので $b(2, N_E(1)) = b(2, \lambda) = 0$, $b(2, N_E(2)) = b(2, a) = 1$ それ以外は 0 となる．

さらに例を用い l_E 次元ベクトル $v_a(row(q_{ck}))$ を説明すると，$v_a(row(q_{cm}))$ は以下のようになる．

$$\begin{aligned} v_a(row(q_{cm})) = & (b(m, N_{E_b}(1)), b(m, N_{E_b}(2)) \\ & , \cdots, b(m, N_{E_b}(|E_b|)) \\ & , 0, 0, \cdots, 0) \\ = & (b(m, b), b(m, ba), b(m, bb) \\ & , \cdots, b(m, bw') \\ & , 0, 0, \cdots, 0) \\ = & (1, 1, \cdots, 1, 0, 0, \cdots, 0) \end{aligned}$$

以上の定義を用いてある DFA を定義する．

任意の $v = (\alpha_1, \alpha_2, \cdots, \alpha_m)$,$\alpha_i \in \{1, 2, \ldots, m\}$ $(i = 1, 2, \ldots, m)$ に対し，DFA A_v を $A_v = (Q, \Sigma, \delta^v, p_0, F)$ とし，遷移関数 δ^v 以外は v によらず等しくなるようにする．

さらに，任意の $w \in \Sigma^*$ に対して，

$$[b(q_{ck}, w) = 1 \Rightarrow \delta^v(q_{ck}, w) \in F] \wedge \left[\begin{array}{l} b(q_{ck}, w) = 0 \vee b(q_{ck}, w) \text{ が定義されていない} \\ \quad \Rightarrow \delta^v(q_{ck}, w) \notin F \end{array} \right]$$

が成り立つように A_v を次のように定義する．定義した A_v が上式をみたすことは後に証明する．

定義 5. 任意の $v = (\alpha_1, \alpha_2, \cdots, \alpha_m), \alpha_i \in \{1, 2, \ldots, m\} (i = 1, 2, \ldots, m)$ に対し，DFA $A_v = (Q, \Sigma, \delta^v, p_0, F)$ を以下のように定める．ただし，$m = 2^{l_E}$ とする．

- 状態 $Q = \{p_0, q_f, q_{a1}, q_{a2}, \cdots, q_{am}, q_{b1}, q_{b2}, \cdots, q_{bm}, q_{c1}, q_{c2}, \cdots, q_{cm}\}$
- アルファベット $\Sigma = \{a, b\}$
- 遷移関数 δ^v を以下のように定める．
  まず，状態 $p_0, q_f, q_{ak} (k = 1, 2, \ldots, m), q_{bk} (k = 1, 2, \ldots, m)$ に対する遷移を定める．

  p_0に対する遷移．
  $\delta^v(p_0, a) = q_{a1}$
  $\delta^v(p_0, b) = q_{b1}$

  $q_{ak} (k = 1, 2, \ldots, m)$に対する遷移．
  $$\delta^v(q_{ak}, a) = \begin{cases} q_{a(k+1)} & (k = 1, 2, \ldots, m-1) \\ q_f & (k = m) \end{cases}$$
  $\delta^v(q_{ak}, b) = q_{ck}$

  $q_{bk} (k = 1, 2, \ldots, m)$ に対する遷移．
  $$\delta^v(q_{bk}, b) = \begin{cases} q_{b(k+1)} & (k = 1, 2, \ldots, m-1) \\ q_f & (k = m) \end{cases}$$
  $\delta^v(q_{bk}, a) = q_{c\alpha_k}$，ただし$\alpha_k$は v の k 番目の成分
  (注：この部分のみがvごとに異なる.)
  q_fに対する遷移．
  $\delta^v(q_f, a) = q_f$
  $\delta^v(q_f, b) = q_f$

  次に，$q_{ck} (k = 1, 2, \ldots, m)$ に対する遷移を定める．
  (i) $\delta^v(q_{ck}, a)$ を $\delta^v(q_{ck}, a) = q_{ck'}$ と定める．
  ただし，$q_{ck'}$ は次の式を満たす
  $$v_a(row(q_{ck})) = row(q_{ck'}).$$
  (ii) $\delta^v(q_{ck}, b)$ を $\delta^v(q_{ck}, b) = q_{ck'}$ と定める．
  ただし，$q_{ck'}$ は次の式を満たす
  $$v_b(row(q_{ck})) = row(q_{ck'}).$$
- p_0 は開始状態
- $F = \{q_f\} \cup \{q_{ck} | k$ が偶数 $\}$

このDFA A_v によって定まる言語を要素にもつ言語集合 $\mathcal{Y}$ を定義する．

定義 6. 言語集合 $\mathcal{Y}$，文字列の集合 $A_{\mathcal{Y}}$，十分例集合の部分集合 $s^{A_{\mathcal{Y}}}_{M,L}$ を以下のように定義する．

$$\mathcal{Y} = \{L(A_v) | v \in \{1, 2, \ldots, m\}^m\}$$

定理1の証明で用いる文字列の集合 $A_{\mathcal{Y}}$ を以下のように定義する．

$$A_{\mathcal{Y}} = \{u_{A_v}(q_{bk}) a \cdot e | k \in \{1, 2, \ldots, m\}, e \in E, v \in \{1, 2, \ldots, m\}^m\}$$

ただし，DFA A_v に対する，関数 $u_{A_v} : Q \to \Sigma^*$ は次のように定義する．

$u_{A_v}(q) = A_v$で開始状態p_0から状態qに遷移する最小な文字列

ただし，最小とは長さ優先辞書式順序で一番最初に現れること．

任意のDFA A_v において，任意の $k \in \{1, 2, \cdots, m\}$ で

$$u_{A_{v_1}}(q_{bk}) = u_{A_{v_2}}(q_{bk}) \tag{1}$$

となる．よって以下この章では $u(q)$ は $u_{A_v}(q)$ を意味する．

よって，$|\{u(q_{ck}) a | k \in \{1, 2, \ldots, m\}| = m$, $|E| = \log m$ より，$|A_{\mathcal{Y}}| = m \log m$ となる．

十分例集合 $s_{M,L}$ の部分集合で A_v の要素を用いているものの全体を，$s^{A_{\mathcal{Y}}}_{M,L}$ とする．形式的には次のように定義する．

$$s^{A_{\mathcal{Y}}}_{M,L} = \left\{ (w, L(w)) \;\middle|\; \begin{array}{l} (w, L(w)) \in s_{M,L} \\ \text{かつ } w \in A_{\mathcal{Y}} \end{array} \right\} \tag{2}$$

以上で $\mathcal{Y}$ を定義した．

$|\mathcal{Y}|$ を計算するため，$A_{v_1} \neq A_{v_2} \Rightarrow L(A_{v_1}) \neq L(A_{v_2})$ を示す．よってDFA A_v が最小状態DFAであることを，補題2で示す．その前に以下の，事実1，事実，を用い補題1を先に示す．

以下では簡単のため，文字列 $w_1, w_2 \in \Sigma^*$ に対する比較文 $<_{lex}$ 次式を満たすように定義する．

$w_1 <_{lex} w_2 \Leftrightarrow w_1$は長さ優先辞書式順序で$w_2$よりも前

事実 1.
任意の $x \in \Sigma, w \in \Sigma^*, k \in \{1,2,\ldots,|E_x|\}$ に対して,

$$xw \in E \Rightarrow [N_{E_x}(k) = xw \Leftrightarrow N_E(k) = w]$$

となる.

事実 2.
任意の $w \in E$, 任意の $q_{ck}, k \in \{1,2,\ldots,m\}$ に対し,

$$\exists w' \in \Sigma^*, \exists x \in \Sigma[w = xw'] \Rightarrow b(q_{ck}, w) = b(\delta^v(q_{ck}, x), w')$$

となる.

補題 1. 任意の $w \in \Sigma^*$, 任意の $k \in \{1,2,\ldots,m\}$ に対して

$$[b(q_{ck}, w) = 1 \Rightarrow \delta^v(q_{ck}, w) \in F] \wedge \left[\begin{array}{l}[b(q_{ck}, w) = 0 \vee b(q_{ck}, w) \text{が定義されていない}] \\ \Rightarrow \delta^v(q_{ck}, w) \notin F\end{array}\right]$$

が成り立つ.

[proof] $w = x_1 x_2 \cdots x_{|w|}, x_i \in \Sigma (i = 1,2,\ldots,|w|)$, $\delta^v(q_{ck}, x_1 x_2 \cdots x_i) = q_{ck_i} (i = 1,2,\ldots,|w|)$ とする. ただし, $k_i \in \{1,2,\ldots,m\}$ とする.

$w \in E$ であるなら, 事実 2. より

$$\begin{aligned} b(q_{ck}, w) &= b(q_{ck_1}, x_2 x_3 \cdots x_{|w|}) \\ &= b(q_{ck_2}, x_3 x_4 \cdots x_{|w|}) \\ &= \cdots \\ &= b(q_{ck_{|w|}}, \lambda) \end{aligned}$$

となる. A_v の定義より,

$$b(q_{ck_{|w|}}, \lambda) = 1 \Rightarrow q_{ck_{|w|}} \in F$$

$$b(q_{ck_{|w|}}, \lambda) = 0 \Rightarrow q_{ck_{|w|}} \notin F$$

となる. よって, $w \in E$ なら補題を満たす.

次に $w \notin E$ について考える.

まず, 空文字列でない, 任意の文字列 $w = xw', x \in \Sigma$, 任意の $k \in \{1,2,\ldots,m\}$ に対する状態 q_{ck} で,

$$w \notin E \wedge w' \in E \Rightarrow b(\delta^v(q_{ck}, x), w') = 0 \qquad (3)$$

となることを示す.

$b(\delta^v(q_{ck}, x), w') = 1$ となるためには, A_v の定義より,

$$\exists k[b(q_{ck}, N_{E_x}(k)) = 1 \wedge N_E(k) = w']$$

とならなくてはならない. さらに, 事実 1. より, $N_E(k) = w' \Rightarrow [E_x(k) = xw' \vee E_x(k)$ は未定義] となる. よって $xw' \in E_x$ となることが必要である. しかし, $E_x \subset E$ より, $w = xw' \notin E$ ならば $w \notin E_x$ となることより, $b(\delta^v(q_{ck}, x), w') = 0$ となる.

さらに, 集合 E が w_* より長さ優先辞書式順序で前にある文字列を全てふくむことから, あるの $j \in \{1,2,\ldots,|w|\}$ において, $x_j x_{j+1} \cdots x_{|w|} \notin E$ かつ $x_{j+1} \cdots x_{|w|} \in E$ となる.

よって式 (3) より, $b(q_{ck_j}, x_{j+1} \cdots x_{|w|}) = 0$ となる. $w' \in E$ であるので事実 2. を用いる事ができ $w \in E$ と同じ議論より, $w \notin E$ についてもこの補題が成り立つ. □

この補題を用い, DFA A_v が最小状態 DFA である事を次に示す.

補題 2. 任意の DFA $A_v (v \in \{0,1,2,\cdots,m\}^m)$ は言語 $L(A_v)$ を識別する最小状態 DFA である.

[proof] 一般的に DFA の状態 $q_x, q_y \in Q$ が本質的に違う状態というのは,

$$\exists w \in \Sigma^* \left[\begin{array}{l}[\delta(q_x, w) \in F \wedge \delta(q_y, w) \notin F] \\ \vee[\delta(q_y, w) \in F \wedge \delta(q_x, w) \notin F]\end{array}\right]$$

となることである.

よって異なる任意の状態 $p, q \in Q$ に対して上式を成立させる文字列 $w \in \Sigma^*$ が存在することを示せばよい.

- 状態 p_0 と以下の状態とを識別する文字列 w を示す.
  1. 状態 $q_{ak} (k = 1,2,\ldots,m)$
     $w = (a)^{m-k+1}$ が, $\delta^v(q_{ak}, w) \in F \wedge \delta^v(q_0, w) \notin F$ を満たす文字列である.
  2. 状態 $q_{bk} (k = 1,2,\ldots,m)$
     $w = (b)^{m-k+1}$ が, $\delta^v(q_{bk}, w) \in F \wedge \delta^v(q_0, w) \notin F$ を満たす文字列である.
  3. 状態 $q_{ck} (k = 1,2,\ldots,m)$
     補題 1 より E に含まれない $(a)^{m+1}$ で, $\delta^v(q_{ck}, (a)^{m+1}) \notin F$ となるので, $w = (a)^{m+1}$ が, $\delta^v(q_{ck}, w) \notin F \wedge \delta^v(q_0, w) \in F$ を満たす文字列である.
  4. 状態 q_f
     $w = \lambda$ が, $\delta^v(q_f, w) \in F \wedge \delta^v(q_0, w) \notin F$ を満たす文字列である.
- 状態 $q_{ak} (k = 1,2,\ldots,m)$ と以下の状態との識別する文字列 w を示す.

1. 状態 $q_{ak'}(k'=1,2,\ldots,m)$, ただし $k \neq k'$
   $(i)k' < k$
   $w = (a)^{m-k+1}$ が, $\delta^v(q_{ak'},w) \notin F \wedge \delta^v(q_{ak},w) \in F$ を満たす文字列である.
   $(ii)k < k'$
   $w = (a)^{m-k'+1}$ が, $\delta^v(q_{ak'},w) \in F \wedge \delta^v(q_{ak},w) \notin F$ を満たす文字列である.
2. 状態 $q_{bk'}(k'=1,2,\ldots,m)$
   $w = (b)^{m-k'+1}$ が, $\delta^v(q_{bk'},w) \in F \wedge \delta^v(q_{ak},w) \notin F$ を満たす文字列である.
3. 状態 $q_{ck'}(k'=1,2,\ldots,m)$
   補題 *1* より E に含まれない $(a)^m$ で, $\delta^v(q_{ck},(a)^m) \notin F$ となるので, $w=(a)^m$ が $\delta^v(q_{ck},w) \notin F \wedge \delta^v(q_{ak},w) \in F$ を満たす文字列である.
4. 状態 q_f
   $w=\lambda$ が, $\delta^v(q_f,w) \in F \wedge \delta^v(q_{ak},w) \notin F$ を満たす文字列である.

- $q_{bk}(k=1,2,\ldots,m)$ と以下の状態との識別する文字列 w を示す.

1. 状態 $q_{bk'}(k'=1,2,\ldots,m)$
   $(i)k' < k$
   $w = (b)^{m-k+1}$ が, $\delta^v(q_{bk'},w) \notin F \wedge \delta^v(q_{bk},w) \in F$ を満たす文字列である.
   $(ii)k < k'$
   $w = (b)^{m-k'+1}$ が, $\delta^v(q_{bk'},w) \in F \wedge \delta^v(q_{bk},w) \notin F$ を満たす文字列である.
2. 状態 $q_{ck'}(k=1,2,\ldots,m)$
   補題 *1* より E に含まれない $(b)^m$ で, $\delta^v(q_{ck},(b)^m) \notin F$ となるので, $w=(b)^m$ が, $\delta^v(q_{ck'},w) \notin F \wedge \delta^v(q_{bk},w) \in F$ を満たす文字列である.
3. 状態 q_f
   $w=\lambda$ が, $\delta^v(q_f,w) \in F \wedge \delta^v(q_{bk},w) \notin F$ を満たす文字列である.

- 状態 $q_{ck}(k=1,2,\ldots,m)$ と以下の状態との識別する文字列 w を示す.

1. 状態 $q_{ck'}(k'=1,2,\ldots,m)$, ただし $k' \neq k$
   補題 *1* より, 図 *2* の表中の $b(q_{ck},w) \neq b(q_{ck'},w)$ となる w で $[\delta^v(q_{ck},w) \in F \wedge \delta^v(q_{ck'},w) \notin F] \vee [\delta^v(q_{ck},w) \notin F \wedge \delta^v(q_{ck'},w) \in F]$ を満たす.
2. 状態 q_f
   補題 *1* より E に含まれない $(a)^m$ で, $\delta^v(q_{ck},(a)^{m+1}) \notin F$ となるので, $w=(a)^{m+1}$ が, $\delta^v(q_{ck},w) \notin F \wedge b\delta^v(q_f,w) \in F$ を満たす文字列である.

以上より, 異なる任意の状態間で

$$[\delta(q_x,w) \in F \wedge \delta(q_y,w) \notin F] \vee [\delta(q_y,w) \in F \wedge \delta(q_x,w) \notin F]$$

を満たす w が存在したので, このDFA A_v は最小状態DFAである. □

次に, 集合 $\mathcal{Y}$ に含まれる言語と $A_{\mathcal{Y}}$ の関係を示す.

補題 3. 任意の $L_1, L_2 \in \mathcal{Y}$ に対して以下のことが成り立つ.

$$w \in \Sigma^* - A_{\mathcal{Y}} \Rightarrow L_1(w) = L_2(w)$$

[proof] 言語 L_1 ,L_2 に対し, DFA A_{v_1}, A_{v_2} を $L_1 = \mathcal{L}(A_{v_1}), L_2 = \mathcal{L}(A_{v_2})$ となる, DFAとする.

以下では w に対する次の条件で場合分けし, 証明する.

1. **任意の $x \in \{1,2\}$ に対し, A_{v_x} の w に対する計算に, $\delta^{v_x}(q_{bk},a), k=1,2,\ldots,m$ の計算が現れない.**
   A_{v_1}, A_{v_2} が異なる部分は $\delta^v(q_{bk},a), k=1,2,\ldots,m(v \in \{v_1,v_2\})$ 中のある遷移である. A_{v_1} と A_{v_2} が文字列 w を受理非受理判定するときに v_1, v_2 によって異なる可能性のある遷移 $\delta^v(q_{bk},a), k=1,2,\ldots,m(v \in \{v_1,v_2\})$ を全て用いないならば, 必ず任意の A_{v_1}, A_{v_2} に対し $A_{v_1}(w) = A_{v_2}(w)$ となる.

2. **ある $x \in \{1,2\}$ に対し, A_{v_x} の w に対する計算に, ある k に対する $\delta^{v_x}(q_{bk},a)$ の計算が現れる.**
   A_v の定義よりある $x \in \{1,2\}$ に対し, A_{v_x} の w に対する計算に, ある k に対する $\delta^{v_x}(q_{bk},a)$ の計算が現れならば, 開始状態 p_0 から状態 q_{bk} に遷移するためには $\delta^{v_x}(p_0, u_{A_{v_x}}(q_{bk}))$ しかなく, $u_{A_{v_x}}(q_{bk})$ は任意の v で等しい.
   $w \in \Sigma^* - A_{\mathcal{Y}}$ を A_{v_x} で判定するときに, DFA $A_{v_x}, v_x \in \{1,2,\cdots,m\}^m$ があるkに対する遷移

$\delta^{v_x}(q_{bk}, a)$ を用いるならば，w の接頭文字列が $u_{A_{v_x}}(q_{bk})a$ となる．よって $y \in \{1,2\}$ かつ $y \neq x$ となる A_{v_y} においても，遷移 $\delta^{v_y}(q_{bk}, a)$ を通る．

一方，$A_{\mathcal{Y}} = \{u(q_{bk})a \cdot c | k \in \{1,2,\ldots,m\}, e \in E\}$ である．よって，任意の $w \in \Sigma^*$ において，

$$[w = u(q_{bk_1})aw' \wedge w \in \Sigma^* - A_{\mathcal{Y}}] \Rightarrow w' \notin E$$

をみたす．さらに A_v の定義より，任意の $v \in \{1,2,\cdots,m\}^m$ で

$$\forall k \in \{1,2,\cdots,m\}, \exists k' \in \{1,2,\cdots,m\}[\delta^v(q_{bk}, a) = q_{ck'}]$$

となり，補題 *1* より，

$$1 \leq \forall k \leq m, \forall w \notin E[\delta^v(q_{ck}, w) = q_{c1}]$$

となる．以上より，$\delta^v(p_0, w') = q_{c1}$ なので，この 2. の条件を満たす w で

$$\delta^{v_1}(p_0, w) = q_{c1} \wedge \delta^{v_2}(p_0, w) = q_{c1}$$

となる．したがって，

$$L_1(w) = L_2(w) = 0(\text{非受理})$$

となる．

よって補題が成り立つ． □

補題 4. 任意の $M \in ALG$ に対して，

$$\forall L_1, L_2 \in \mathcal{Y}[L_1 \neq L_2 \Rightarrow s^{A_{\mathcal{Y}}}_{M,L_1} \neq s^{A_{\mathcal{Y}}}_{M,L_2}]$$

が成り立つ．

[proof] 背理法で示す．ある $M \in ALG$ で，

$$\exists L_1, L_2 \in \mathcal{Y}[L_1 \neq L_2 \wedge s^{A_{\mathcal{Y}}}_{M,L_1} = s^{A_{\mathcal{Y}}}_{M,L_2}] \qquad (4)$$

であったと仮定する．

φ を s_{M,L_1} を含む最小の有限列とする．十分例集合の定義より

$$\mathcal{L}(M(\varphi)) = L_1$$

となる．

さらに，φ に続けて集合 $\{(w, L_1(w)) | w \in W(s_{M,L_2} - s^{A_{\mathcal{Y}}}_{M,L_2})\}$ を全てを含むように φ' を連結し，有限列 $\varphi \cdot \varphi'$ を考える．M の保守性から

$$\mathcal{L}(M(\varphi \cdot \varphi')) = L_1$$

となる．

以下では，$\varphi \cdot \varphi'$ には L_2 の十分例集合 s_{M,L_2} も含んでいることを示めし，十分例集合の定義の無矛盾を導き出す．

まず，φ には $s^{A_{\mathcal{Y}}}_{M,L_1}$ を含んでいるので，仮定より $s^{A_{\mathcal{Y}}}_{M,L_1}$ と等しい $s^{A_{\mathcal{Y}}}_{M,L_2}$ を含んでいることになる．

次に，

$$W(s_{M,L_2} - s^{A_{\mathcal{Y}}}_{M,L_2}) \subset \Sigma^* - A_{\mathcal{Y}}$$

となり，補題 *3* より，任意の $w \in \subset \Sigma^* - A_{\mathcal{Y}}$ に対して $L_1(w) = L_2(w)$ となる．よって，

$$\{(w, L_1(w)) | w \in W(s_{M,L_2} - s^{A_{\mathcal{Y}}}_{M,L_2})\}$$
$$= \{(w, L_2(w)) | w \in W(s_{M,L_2} - s^{A_{\mathcal{Y}}}_{M,L_2})\}$$

となる．

この 2 点より，$\varphi \cdot \varphi'$ は s_{M,L_2} を含む．

以上より十分例集合 s_{M,L_2} を $\varphi \cdot \varphi'$ は含んでいるので，$M(\varphi \cdot \varphi') = L_2$ とならなければならない．これは十分例集合の定義に矛盾，仮定が誤り． □

次に定理 *1* の証明に，用いる事実 3．を記す．

事実 3.

$n \geq 3$ のとき，整数の変数 $x, 1 \leq x \leq n/8$ は以下の不等号を満たす．

$$2^{n-x} > x\binom{n}{x} \qquad (5)$$

ここで以上の補題と事実をふまえ，定理 *1* の証明を行う．つまり

$$|s_{*,REG_n}| = \Omega(n \log n)$$

を示す．

[proof] 正規言語の部分集合 $\mathcal{Y}$ に含まれている言語に対して，サイズが最大の十分例集合に必要な大きさを考える．

まず，(i),(ii) を求め，(iii) で (i)(ii) の関係から下界を示す

(i). $|\mathcal{Y}|$ の大きさを求める．

補題 *2* より，

$$\forall v_1, v_2 \in \{1,2,\cdots,m\}^m, [v_1 \neq v_2 \Leftrightarrow L(A_{v_1}) \neq L(A_{v_2})]$$

であることから，DFA A_v の種類，つまり $v \in \{1,2,\ldots,m\}^m$ の種類をを考えればよい．よって $|\mathcal{Y}| =$

m^m となる.

(ii). 大きさ k 以下の $s^{A_\mathcal{Y}}_{M_*,L}$ になりうる集合の種類の上界を求める.

まず, 大きさが k の $s^{A_\mathcal{Y}}_{M_*,L}$ になりうる集合の種類の上界を考える. $A_\mathcal{Y}$ 中からどの文字列 k 個を選ぶかと, その文字列のラベルがどうなっているかを考えることによって, 大きさが k の $s^{A_\mathcal{Y}}_{M_*,L}$ になりうる集合の種類の上界は

$$s^{A_\mathcal{Y}}_{M_*,L}\text{となりうる集合の種類} \leq 2^k \binom{m\log m}{k} \quad (6)$$

となる.

このことをふまえ, 次に k 以下の $s^{A_\mathcal{Y}}_{M_*,L}$ になりうる集合の種類数の上界を考える. 式 6 より, 求める上界は

$$\sum_{i=1}^{i=k} 2^i \binom{m\log m}{i}$$

となる. この式は $1 \leq k \leq \frac{m\log m}{2}$ の範囲で以下のように抑えられる.

$$\sum_{i=1}^{i=k} 2^i \binom{m\log m}{i} \leq k2^k \binom{m\log m}{k}$$

以上より (ii) は求まった.

(iii). (i) と (ii) の関係より, 求める下界を示す.

補題 4 より,

$$\forall L_1, L_2 \in \mathcal{Y}[L_1 \neq L_2 \Rightarrow s^{A_\mathcal{Y}}_{M,L_1} \neq s^{A_\mathcal{Y}}_{M_*,L_2}]$$

を満たすので, 十分例集合の部分集合 $s^{A_\mathcal{Y}}_{M_*,L}$ となりうる集合の種類よりも $|\mathcal{Y}|$ の大きさの方が大きくはなりえない. 集合 $\mathcal{Y}$ 中で $s^{A_\mathcal{Y}}_{M_*,L}$ が最大のものを, $s^{A_\mathcal{Y}}_{M_*,L_y}$ とすると, 任意の $k, 1 \leq k \leq \frac{m\log m}{2}$ に対し,

$$|s^{A_\mathcal{Y}}_{*,L_y}| \leq k \Rightarrow m^m \leq k2^k \binom{m\log m}{k}$$

となる.

つまり対偶をとると, 任意の $k', 1 \leq k' \leq \frac{m\log m}{2}$ に対し,

$$m^m > k'2^{k'} \binom{m\log m}{k'} \Rightarrow |s^{A_\mathcal{Y}}_{*,L_y}| > k'$$

となる.

さらに,

$$|s_{*,Reg_n}| \geq |s_{*,L_y}| \geq |s^{A_\mathcal{Y}}_{*,L_y}|$$

であるのは明らかなので,

$$m^m > k'2^{k'} \binom{m\log m}{k'} \quad (7)$$

を満たす k' が $|s_{*,REG_n}|$ の下界となる.

補題 5 より, $k' = \frac{m\log m}{8}$ で (7) の式を満たす.

さらに, $n = 3m+2$ より $k' = \frac{n-2}{24}\log\frac{n-2}{3}$ となり,

$$|s_{*,REG_n}| \geq \frac{n-2}{24}\log\frac{n-2}{3}$$

となる. よって

$$|s_{*,REG_n}| = \Omega(n\log n)$$

となり, 下界が示せた. □

参考文献

[Oci92] Ocina, J.and Garcia, P. :Inffering regular languages in polynomial update time, *In Pattern Recognition and Image Analysis* :49-61, 1992.

[Gold67] Gold, E. M. :Language identigication in the limit, *Information and Control*, 10:447-474, 1967.

[John71] John, E. H. An n log n algorithm for minimizing the states in a finite automaton, *Z. The Theory of Machines and Computations*, Academic Press 1971.

[Cris07] Tirnauca, C. and Kobayashi, S. A Characterization of the Language Classes Learnable with Correction Queries, *TAMC2007*, 398-407, 2007.

[Lan04] Lange, S. and Zilles, S. :Formal language identification: querylearning vs. Gold-style learning, *Information Processing Letters*, 91:285-292, 2004.

[Angl87] Angluin, D. :Learning regular sets fron queries and counterexamples, *Information and Computation*, 75:87-106, 1987.

[SKY01] Sakakibara, Y. Kobayashi, S. and Yokomori, T. :計算論的学習. 培風館,2001.