数理解析研究所講究録
1148 巻 2000 年 70-75

質問と正の反例による正則言語の多項式時間学習

但馬 康宏 (TAJIMA Yasuhiro)、富田 悦次 (TOMITA Etsuji)
電気通信大学大学院電気通信学研究科電子情報学専攻
182-8585 東京都調布市調布ヶ丘 1-5-1
tajima@ice.uec.ac.jp, tomita@ice.uec.ac.jp

1 はじめに

正則言語の学習問題は、機械学習において最も古くから研究されているテーマの一つであるが、その困難性 [6][4] ゆえに言語理論のみならず学習モデルの発展に影響を与えてきた [1][2][8][10]。特に Angluin[2] による質問による学習モデルは、正則言語に対して最小限と思われる教師 (Minimally Adequate Teacher: MAT) モデルを定義し、多項式時間学習について大きな成果をもたらした。ここで正則言語に関する MAT は、所属性質問及び等価性質問の 2 種類の質問に答えられる教師と定義される。また Angluin[3] は、いくつかの質問を提案し、その質問を用いた学習可能性に関して論じた。

本論文では Angluin[3] により紹介されている、以下のような質問を用いた正則言語の多項式時間学習可能性について考察する。

- 包含性質問 : 仮説の提示に対し、学習対象言語が仮説言語に含まれるか否かが返答される質問。
- 部分性質問 : 仮説の提示に対し、仮説言語が学習対象言語に含まれるか否かが返答される質問。

このとき正則言語は、その最簡形の決定性有限状態機械の状態数が既知ならば、上記の質問のうちどちらか一方のみを用いて多項式時間学習可能であることを示す。これは、状態数及び所属性質問では多項式時間学習不可能であること [1]、また、状態数及び等価性質問では多項式時間学習は難しいこと [6] と考えあわせると、教師にとって解決するのに難しい質問ほど学習者にとって有効であることを示している。

2 諸定義

有限状態機械を $M = (Q, \Sigma, \delta, q_0, F)$ と定義する。ここで Q は状態の有限集合、Σ は入力記号の有限集合、δ は推移規則の有限集合、$q_0 \in Q$ は開始状態、$F \subseteq Q$ は受理状態の集合である。推移規則 δ は、状態 $q \in Q$ 及び入力記号 $a \in \Sigma$ の組に対する関数である場合、すなわち $\delta(q, a) = q' \in Q$ として表せる場合、決定性であるといい、M を決定性有限状態機械 (DFA) と呼ぶ。決定性でない δ を持つ M を非決定性有限状態機械 (NFA) と呼ぶ。空記号列を ε で表す。本論文においては、すべての有限状態機械は、$\varepsilon-$ 推移なしであり、かつ最簡形であるとする。NFA M における状態 $q \in Q$ から $a \in \Sigma$ により推移可能な状態を $q'_1, q'_2, \cdots, q'_n$ としたとき、$\delta(q, a) = \{q'_1, q'_2, \cdots q'_n\}$ と表す。

DFA M において、$q \in Q$, $a \in \Sigma$ 及び $w \in \Sigma^*$ とする。このとき $\hat{\delta}$ を

1. $\hat{\delta}(q, \varepsilon) = q$
2. $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$

と定める。また、NFA M において $q \in Q$, $a \in \Sigma$ 及び $w \in \Sigma^*$ とする。$\hat{\delta}$ を

1. $\hat{\delta}(q, \varepsilon) = \{q\}$
2. $\hat{\delta}(q, wa) = \bigcup_{r \in \hat{\delta}(q,w)} \delta(r, a)$

と定める。M を NFA もしくは DFA とし、$q \in Q$ とする。記号列 $w \in \Sigma^*$ は、M が DFA のとき $\hat{\delta}(q, w) \in F$、M が NFA のとき $\hat{\delta}(q, w) \cap F \neq \{\}$ ならば、M において q に受理される語と呼ばれる。また、$L(M, q) = \{w \in \Sigma^* \mid w$ は M において q に受理される語 $\}$ と定め、q の受理する言語と呼ぶ。さらに、M において q_0 に受理される語を M に受理される語と定め、$L(M) = L(M, q_0)$ を M の受

理する言語と呼ぶ。このほかの定義に関しては、文献 [7] によるものとする。

2.1 質問の定義

[等価性質問] 正則言語 L_t に対する等価性質問を以下のように定義する。

入力: DFA M

出力: $yes \ldots if(L_t = L(M))$
$no, w \in (L_t - L(M)) \cup (L(M) - L_t)$
$\ldots if(L_t \neq L(M))$

ここで、w を反例と呼ぶ、さらに $w \in L_t$ ならば正の反例、$w \notin L_t$ ならば負の反例と呼ぶ。

[包含性質問] 正則言語 L_t に対する包含性質問を以下のように定義する。

入力: DFA M

出力: $yes \ldots if(L_t \subseteq L(M))$
$no, w \in (L_t - L(M)) \ldots if(L_t \not\subseteq L(M))$

[部分性質問] 正則言語 L_t に対する部分性質問を以下のように定義する。

入力: DFA M

出力: $yes \ldots if(L(M) \subseteq L_t)$
$no, w \in (L(M) - L_t) \ldots if(L(M) \not\subseteq L_t)$

[所属性質問] 正則言語 L_t に対する所属性質問を以下のように定義する。

入力: 記号列 $w \in \Sigma^*$

出力: $yes \ldots if(w \in L_t)$
$no \ldots if(w \notin L_t)$

このとき、以下の定理が得られている。

定理 1 (Angluin[2]) 任意の正則言語は、等価性質問及び所属性質問を用いて多項式時間厳密学習可能である。 □

2.2 representative sample と live-complete set

定義 2 DFA M について、$Rep \subseteq L(M)$ が representative sample であるとは、以下の条件を満たすこととする。

- $L(M,p) \neq \{\}$ なるすべての $p \in Q$ および $L(M,\delta(p,a)) \neq \{\}$ なるすべての $a \in \Sigma$ について、$\hat{\delta}(q_0, w_1) = p$ かつ $w_2 \in L(M, \delta(p,a))$ なる $w_1 \cdot a \cdot w_2 = w \in Rep$ が存在する。

すなわち、すべての推移規則が使用されている正の例の集合である。

また、正則言語 L について $Rep \subseteq L$ が representative sample であるとは、$L(M) = L$ なる DFA M について Rep が representative sample であることとする。 □

定義 3 DFA M について、$LC \subset \Sigma^*$ が live-complete set であるとは、以下の条件を満たすこととする。

- $L(M,p) \neq \{\}$ なるすべての $p \in Q$ について、$\hat{\delta}(q_0, w) = p$ なる $w \in \Sigma^*$ が LC に含まれている。

すなわち、すべての状態に到達可能な記号列の集合である。

また、正則言語 L について $LC \subset \Sigma^*$ が live-complete set であるとは、$L(M) = L$ なる DFA M について LC が live-complete set であることとする。 □

明らかに、representative sample Rep について $LC = \{w \in \Sigma^* \mid wv \in Rep, v \in \Sigma^*\}$ は live-complete set である。このとき、以下の定理が得られている。

定理 4 (Angluin[1]) 任意の正則言語は、所属性質問及び live-complete set を用いて多項式時間厳密学習可能である。 □

3 正則言語の学習

正則言語の学習は、質問等をとおして得られたサンプルに対し、そのサンプルのみを正しく認識する

自明な DFA を構成し、その DFA の状態を適切に統合していく過程と見ることができる [9]。

以後、学習対象となる正則言語を L_t と表し、$M_t = (Q_t, \Sigma, \delta_t, q_{t0}, F_t)$ を $L(M_t) = L_t$ なる最簡形の DFA とする。

定義 5 (接頭辞木オートマトン) 任意の記号列の集合 $R \subset \Sigma^*$ について、接頭辞木オートマトン $PTA(R)$ を以下のように定める。

$$\begin{aligned} Q &= \{u \in \Sigma^* \mid uv \in R, v \in \Sigma^*\} \cup \{\varepsilon, d_0\} \\ \delta(w,a) &= \begin{cases} wa & (a \in \Sigma \text{について、} w \in Q \\ & \quad \text{かつ } wa \in Q \text{のとき}) \\ d_0 & (otherwise) \end{cases} \\ q_0 &= \varepsilon \\ F &= \{w \in Q \mid w \in L_t\} \end{aligned}$$

また、Q 上のある同値関係にもとづいた同値類の集合 π を Q の分割と呼び、π を用いた商オートマトン $M/\pi = (Q', \Sigma, \delta', q'_0, F')$ を以下のように定める。

$$\begin{aligned} Q' &= Q/\pi = \{B(q,\pi) \mid q \in Q\} \\ \delta'(B(q,\pi),a) &= B(r,\pi) \quad (\delta(q,a) = r \text{のとき}) \\ q'_0 &= B(q_0,\pi) \\ F' &= \{B(q,\pi) \in Q' \mid \\ & \qquad B(q,\pi) \cap F \neq \{\}\} \end{aligned}$$

ここで $B(q,\pi)$ は、q を含む π の要素であり、ブロックと呼ばれる。

分割 $\pi_1 = \{B_1, B_2, \cdots, B_n\}$ 及び $j \neq k$ なる、ある $1 \leq j,k \leq n$ について $\pi_2 = \{B_j \cup B_k\} \cup (\pi_1 - \{B_j, B_k\})$ とする。このとき、分割 π_1 は分割 π_2 より精密であるといい、$\pi_1 \preceq \pi_2$ と表す。さらに $\preceq$ の推移閉包を $\ll$ と表す。 □

DFA M について、$\pi_1 \ll \pi_2$ ならば $L(M/\pi_1) \subseteq L(M/\pi_2)$ が成り立つ。また、関係 $\ll$ によって順序づけられた有限状態機械全体の集合はブール束となり、最下限は M そのもの、最上限はすべての記号列を受理する 1 状態の有限状態機械となる。このブール束を $LAT(M)$ と表す。このとき、以下の定理が得られている。

定理 6 (Dupont[5]) DFA M_t に対して $Rep \subseteq L(M_t)$ を M_t の representative sample とする。このとき、M_t は $LAT(PTA(Rep))$ に含まれる。 □

3.1 所属性質問を用いた学習アルゴリズム

定理 6 にもとづき、まず、文献 [1] において示されている学習アルゴリズムを $LAT(PTA(Rep))$ における探索として表す。その後、文献 [2] において示されたアルゴリズムと文献 [1] において示されたアルゴリズムとの関係を示す。最後にその関係を利用し、包含性質問を用いた学習アルゴリズムを示す。

まず、文献 [1] における学習アルゴリズムを $LAT(PTA(Rep))$ における探索として表す。

(文献 [1] のアルゴリズム) LC をアルゴリズムへ入力される学習対象の live-complete set とする。

1. $LC_0 = \{ua \mid u \in LC, a \in \Sigma\}$ とする。
2. $PTA(LC_0) = (Q, \Sigma, \delta, q_0, F)$ ついて、学習アルゴリズムが保持している Q の分割を π とする。(実行開始時は $\pi_0 = \{B_0 = \{q \in Q \mid q \notin L_t\}, B_1 = \{q \in Q \mid q \in L_t\}\}$ とする。) 検査用の記号列集合を W とする。(実行開始時は $W = \{\varepsilon\}$ とする。)
3. もし、$PTA(LC_0)/\pi = (Q', \Sigma, \delta', q'_0, F')$ が NFA であれば、以下の手順で π を細分化する。もし、$PTA(LC_0)/\pi$ が DFA であれば、仮説として提示し、終了する。
4. 非決定性であるブロックを $B(q,\pi)$ とする。すなわち、ある $a \in \Sigma$ について、$\delta'(B(q,\pi),a) \supseteq \{B(q_1,\pi), B(q_2,\pi)\}$ かつ $B(q_1,\pi) \neq B(q_2,\pi)$ であるとする。
5. すべての $w \in W$ について、$q_1 \cdot w$ 及び $q_2 \cdot w$ の所属性質問を行い、結果が食い違う W の要素を w' とする。
6. すべての $r \in B(q,\pi)$ について $r \cdot a \cdot w'$ の所属性質問を行い、$B_0 = \{r \in B(q,\pi) \mid r \cdot a \cdot w' \notin L_t\}$、$B_1 = \{r \in B(q,\pi) \mid r \cdot a \cdot w' \in L_t\}$ とする。
7. $\pi = \pi - \{B(q,\pi)\} + \{B_0, B_1\}$ とする。
8. $a \cdot w'$ を W に新たに加え、ステップ 3. に戻る。

すなわち、$LAT(PTA(Rep))$ の最上限から探索を始め、所属性質問を用いて最下限に向けて仮説を更新している。さらに、上記アルゴリズムが終了した時点での DFA $PTA(LC_0)/\pi$ は、アルゴリズム中で行った所属性質問の結果及び LC に対して矛盾せず、かつ $LAT(PTA(Rep))$ 中で最も大きな言語を生成する。

ここで、L_t に対する live-complete set ではなく、かつ接頭辞について閉じた $LC' \subset \Sigma^*$ を live-complete set と見立てて上記アルゴリズムを実行し、得られる DFA を $M_h = (Q_h, \Sigma, \delta_h, q_{h0}, F_h)$ であるとする。また、任意の $u \in (L_t - L(M_h)) \cup (L(M_h) - L_t)$ なる語について、$Q_u = \{q \in Q_t \mid q = \hat{\delta}_t(q_{t0}, u'), u'u'' = u, u'' \in \Sigma^*\}$ すなわち M_t において u の接頭辞で到達可能な状態の集合とし、$Q_{LC'} = \{q \in Q_t \mid q = \hat{\delta}_t(q_{t0}, v'), v'v'' \in LC', v'' \in \Sigma^*\}$ すなわち LC' のすべての語で到達可能な状態の集合とする。$Q_u \subseteq Q_{LC'}$ であると仮定する。このとき M_h は、Q_t におけるある分割 π について M_t/π と表された有限状態機械のいくつかの推移規則を取り除いたものとして表せる。さらに π は、M_t における受理状態と非受理状態を統合しないので、$u \in L(M_t)$ ならば、かつそのときに限り $u \in L(M_h)$ である。これは $u \in (L_t - L(M_h)) \cup (L(M_h) - L_t)$ に矛盾する。したがって、$Q_u \not\subseteq Q_{LC'}$ である。

以上より、文献 [2] におけるアルゴリズムは以下のように記述できる。

(文献 [2] のアルゴリズム)

1. $LC' = \{\varepsilon\}$ とする。
2. $LC' = LC' \cup \{u \cdot a \in \Sigma^* \mid u \in LC', a \in \Sigma\}$ とする。この LC' を用いて **(文献 [1] のアルゴリズム)** を実行する。
3. 得られた DFA M_h について等価性質問を行う。
4. 反例が得られなければ終了し、反例 w が得られた場合は、w のすべての接頭辞を LC' に加えステップ 2. に戻る。

4 包含性質問による live-complete set の獲得

学習者は、M_t の状態数 n を既知であるとする。このとき、所属性質問と包含性質問による学習を行うことができる。アルゴリズムの方針は、**(文献 [1] のアルゴリズム)** で求めた DFA に対する反例を包含性質問を用いて獲得し、L_t に対する live-complete set となるようにすればよい。

学習者は、live-complete set の候補としている集合 LC' について **(文献 [1] のアルゴリズム)** を行い、得られた DFA M_h に対して包含性質問を行う。もし、反例が得られるならば、その反例のすべての接頭辞を新たに LC' に加える。もし、反例がないならば、M_h のある一つの推移規則を無効にし、かつ今まで得られた例に対し矛盾しない DFA M_h'' を構成し、包含性質問を行う。このとき、$L(M_t) \subseteq L(M_h)$ が成り立つので LC' が不完全ならば、すなわち L_t についての live-complete set になっていなければ、M_h'' の中に LC' の記号列では到達不可能な状態に到達できる反例を得るものが存在する。したがって、上記の操作を n 回繰り返すことにより、LC' は L_t について live-complete set となる。

以下に詳しい手順を示す。ここで、学習者が L_t の live-complete set の候補としている記号列集合を LC'、さらにそれを用いて **(文献 [1] のアルゴリズム)** にて得られた DFA を $M_h = PTA(LC')/\pi = (Q_h, \Sigma, \delta_h, q_{h0}, F_h)$ とする。

(包含性質問によるアルゴリズム)

1. $LC' = \{\varepsilon\}$ とする。
2. 以下 3. から 11. を n 回繰り返す。
3. **(文献 [1] のアルゴリズム)** を実行し、DFA M_h を得る。
4. M_h について包含性質問を行い、反例 w が得られた場合は、そのすべての接頭辞を LC' に加え、ステップ 3. に戻りループを繰り返す。反例が得られなかった場合は、以下を実行する。
5. $W = \{\}$ とし、すべての $q \in Q_h$ 及び $a \in \Sigma$ の組について、以下の 6. から 9. を行う。
6. $\delta_h'(r, b)$ を $r = q$ かつ $b = a$ のとき $undefined$ とし、それ以外の場合は $\delta_h'(r, b) = \delta_h(r, b)$ とする。さらに、$M_h' = (Q_h, \Sigma, \delta_h', q_{h0}, F_h)$ とする。すなわち、q から a による推移を無効にする。
7. 今までに行われたすべての所属性質問の結果、及び得られた反例に矛盾しないように M_h' を修正する。すなわち、そのような記号列 u すべてについて $Z_u = \{v' \in \Sigma^+ \mid u', u'' \in \Sigma^*, u = u'u'', \delta_h(q_{h0}, u') = q, v'v'' = u'', v'' \in \Sigma^*\}$（ここで q は、ステップ 5. にて仮定した $q \in Q_h$ とする）を求め、$PTA(\cup Z_u)$ の初期状態を q とする。M_h' にこのような修正をしたものを M_h'' とする。
8. M_h'' について包含性質問を行い、反例 w が得られた場合は $W = \{w\} \cup W$ とする。

9. ステップ 6. に戻り、ループを繰り返す。
10. W に含まれる語のすべての接頭辞を LC' に加える。
11. ステップ 3. に戻り、ループを繰り返す。
12. 得られた LC' を使い **(文献 [1] のアルゴリズム)** を実行し、仮説 M_h を得る。
13. M_h を提示し終了する。

以下の定理が成り立つ。

定理 7 正則言語は、その最簡形の DFA の状態数、所属性質問及び包含性質問を用いて、多項式時間学習可能である。

(証明) 学習アルゴリズムにおいて LC' が L_t の live-complete set になっていれば、仮説が正しいことは明らか。L_t について live-complete set ではない LC' から DFA M_h を得たとする。$L(M_h)$ は、得られた例に矛盾しない $LAT(PTR(Rep))$ におけるもっとも大きな言語であるので、$L(M_h'')$ に対する包含性質問の反例には M_h'' において削れらた推移規則が使われる。すなわち、学習アルゴリズムにおける W には、M_t において LC' の語では到達できない状態に到達可能な語が含まれる。また、M_h に対する包含性質問の結果、反例が得られた場合は、その反例で到達可能な M_t における状態の中に LC' の語では到達できない状態が存在する。M_t の状態数は n であるため、n 回の繰り返し後に LC' は M_t について live-complete set となる。すなわち、題意が成り立つ。 □

5 質問のおきかえ

正則言語は、その補集合について閉じている。したがって、以下のような学習者と教師の中継ぎを考えれば、包含性質問を部分性質問に置き換えることができる。

- 教師からの所属性質問の結果は、正負を反転して学習者に伝える。
- 学習者からの仮説 M_h の包含性質問に対して、その補集合を受理言語とする $\bar{M_h}$ を構成し、教師に対し部分性質問を行う。得られた結果をそのまま学習者に返す。
- 学習終了後の仮説も、その補集合を受理する $\bar{M_h}$ を最終的な仮説とする。

したがって、以下の系が成り立つ。

系 8 正則言語は、その最簡形の DFA の状態数が既知ならば、所属性質問及び部分性質問を用いて、多項式時間学習可能である。 □

また、正則言語においては、一つの記号列のみからなる言語に対する部分性質問で所属性質問を置き換えることができるので、以下の系が成り立つ。

系 9 正則言語は、その最簡形の DFA の状態数が既知ならば、部分性質問を用いて多項式時間学習可能である。 □

仮説の補集合を提示する中継ぎを用いれば、同様に以下の系も成り立つ。

系 10 正則言語は、その最簡形の DFA の状態数が既知ならば、包含性質問を用いて多項式時間学習可能である。 □

6 具体例

具体例として、L_t を 0,1 ともに偶数個である記号列の集合とした場合を考える。すなわち、$M_t = (Q_t, \{0,1\}, \delta_t, t_0, F_t)$ を $Q_t = \{t_0, t_1, t_2, t_3\}$, $\delta_t(t_0, 0) = t_1$, $\delta_t(t_0, 1) = t_2$, $\delta_t(t_1, 0) = t_0$, $\delta_t(t_1, 1) = t_3$, $\delta_t(t_2, 0) = t_3$, $\delta_t(t_2, 1) = t_0$, $\delta_t(t_3, 0) = t_2$, $\delta_t(t_3, 1) = t_1$, $F_t = \{t_0\}$ とする。学習者は、L_t を表すことのできる最簡形 DFA の状態数 4 をあらかじめ知っており、包含性質問と所属性質問を用いることができるとする。

1. $LC' = \{\varepsilon\}$ とする。
2. 以下 3. から 11. を 4 回繰り返す。
3. **(文献 [1] のアルゴリズム)** を実行し、以下の $M_h = (Q_h, \{0,1\}, \delta_h, q_0, F_h)$ を得る。

$$
\begin{aligned}
&Q_h = \{q_0, q_1, q_2\}\\
&\delta_h(q_0, 0) = q_1, \quad \delta_h(q_0, 1) = q_2\\
&\delta_h(q_1, 0) = q_0, \quad \delta_h(q_1, 1) = q_2\\
&\delta_h(q_2, 0) = q_2, \quad \delta_h(q_2, 1) = q_0\\
&F_h = \{q_0\}
\end{aligned}
$$

4. さらに、M_h を提示し包含性質問を行う (yes の返答がある)。

5. $W = \{\}$ とし、$(q_0, 0)$, $(q_0, 1)$, $(q_1, 0)$, $(q_1, 1)$, $(q_2, 0)$, $(q_2, 1)$ の 6 組について以下の 6. から 9. のループを繰り返す。
6. ここで、$(q_2, 0)$ の場合を考える。$M'_h = (Q_h, \{0,1\}, \delta'_h, q_0, F_h)$ は、$Q_h = \{q_0, q_1, q_2\}$, $\delta_h(q_0, 0) = q_1$, $\delta_h(q_0, 1) = q_2$, $\delta_h(q_1, 0) = q_0$, $\delta_h(q_1, 1) = q_2$, $\delta_h(q_2, 1) = q_0$, $F_h = \{q_0\}$ となる。
7. 得られた M'_h を Rep' 及び所属性質問の結果に矛盾しないように修正し、以下の $M''_h = (Q_h \cup \{tmp_1, tmp_2\}, \{0,1\}, \delta''_h, q_0, F_h)$ を得る。

$$
\begin{aligned}
&\delta''_h(q_0, 0) = q_1, \quad \delta''_h(q_0, 1) = q_2 \\
&\delta''_h(q_1, 0) = q_0, \quad \delta''_h(q_1, 1) = q_2 \\
&\delta''_h(q_2, 1) = q_0, \quad \delta''_h(q_2, 0) = tmp_1 \\
&\delta''_h(tmp_1, 0) = tmp_2 \\
&F_h = \{q_0\}
\end{aligned}
$$

8. M''_h を提示し包含性質問を行う (反例 1001 を得たとする)。
9. $W = \{1001\} \cup W$ とする。(他の組み合わせについても同様に行うために、ステップ 6. に戻る。)
10. W に含まれる語のすべての接頭辞を LC' に加える。(このとき、$1001 \in W$ なので LC' は、M_t の live-complete set となっている。)
11. ステップ 3. に戻り、ループを繰り返す。
12. 最終的な仮説 M_h を LC' を用いて生成する。
13. M_h を提示して終了する。

このとき、最終的に提示された仮説 M_h は、L_t の live-complete set である LC' をもとに構成されているため、$L_t = L(M_h)$ であることが保証される。

7 おわりに

正則言語は、その最簡形の状態数 n が既知ならば、包含性質問、もしくは部分性質問のみから多項式時間厳密学習可能であることを示した。今後の課題として n が未知の場合や、質問を例の提示のみにした場合の近似学習可能性が挙げられる。

参考文献

[1] D. Angluin, "A note on the number of queries needed to identify regular languages," Information and Control, vol.51, no.1, pp.76–87, Oct. 1981.

[2] D. Angluin, "Learning regular sets from queries and counterexamples," Information and Computation, vol.75, no.2, pp.87–106, Nov. 1987.

[3] D. Angluin, "Queries and concept learning," Machine Learning, vol.2, no.3, pp.319–342, 1988.

[4] D. Angluin, "Negative results for equivalence queries," Machine Learning, vol.5, no.2, pp.121–150, June 1990.

[5] P. Dupont, L. Miclet and E. Vidal, "What is the search space of the regular inference?," Proc. of Second International Colloquium on Grammatical Inference (ICGI-94), Lecture Notes in Artificial Intelligence 862, pp.26–37, 1994.

[6] E. M. Gold, "Complexity of automaton identification from given data," Information and Control, vol.37, no.3, pp.302–320, June 1978.

[7] J. E. Hopcroft and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley, Reading MA, 1979.

[8] L. Pitt, "Inductive inference, DFAs, and computational complexity," Proc. of Workshop on Analogical and Inductive Inference (AII-89), Lecture Notes in Computer Science 397, pp.18–44, 1989.

[9] 榊原 康文, 小林 聡, "形式言語の帰納的学習," 人工知能学会誌, vol.14, no.5, pp.781–789, Sep. 1999.

[10] T. Yokomori, "On polynomial-time learnability in the limit of strictly deterministic automata," Machine Learning, vol.19, no.2, pp.153–179, June 1995.