数理解析研究所講究録
1093巻 1999年 105-110

Extended Reversible 言語とその正例からの多項式時間帰納推論

植村 仁 * (Jin Uemura)　　佐藤 優子 ** (Masako Sato)
* 大阪府立大学総合科学研究科　** 大阪府立大学総合科学部

概 要: 本稿では、Angluin によって導入された k-reversible オートマトンを拡張し、simple prefix free (SPF) と呼ばれる語の有限集合上で遷移する (l,k)-extended reversible オートマトンを定義する。このオートマトンで受理される言語の特徴付け及びそれらの言語族の間の関係等について論じる。
また、(l,k)-extended reversible 言語の族は正例から多項式時間で推論可能となることを示す。

1 序

1967年, Gold[5] は「極限における同定」という枠組みで帰納推論の数学的モデルを提案し, 形式言語と帰納的関数の帰納推論に理論的基盤を与えた。

推論アルゴリズムは, 入力要求 ⇒ 計算 ⇒ 出力という過程の繰り返しである。本稿で扱う正例からの言語の帰納推論では, 入力として正の事例, すなわち, 目標言語に含まれる語だけが与えられる。出力は, 目標言語を記述するオートマトンや文法等の表現である。推論可能な言語の族の特徴付け定理が Angluin [2] によって証明されている。本稿で扱う言語は, Chomsky 階層の最下位にある正則 (regular) 言語であるが, この族は, 正例から推論可能ではないことが示されている ([5])。これまで, 正例から推論可能となる正則言語の部分族が提案されてきた。例えば, 語上の正則表現に連接・和・Kleene の $*$ 演算を高々 n 回施して得られる正則言語の族は正例から推論可能である ([9])。また, 最近, Head 等 [6] は, 本稿で扱う reversible 言語族と, strictly locally testable な言語の族を統一的に記述できる k-DB$_n$ と呼ばれる言語族を導入し, 正則言語の族が, k-DB$_n$ 言語族を用いて, 正例から近似的に推論可能であることを示した。

正則言語の推論に関しては, 効率的な推論という観点から様々な正則言語が導入されてきた。Angluin [3] が導入した k-reversible 言語は, 正例から多項式時間で推論可能である。また, 線形文法に対する Szilard 言語 (Makinen[7]) や, この言語を simple prefix free (SPF) と呼ばれる語の有限集合上へ拡張した strictly regular 言語 (Tanida & Yokomori[10], Yokomori [13]), 同じく SPF 上で定義された k-simple regular 言語 (Sato & Sato [8], Watanabe & Sato [11], [12]) 等も正例から多項式時間推論可能である。

本稿では Angluin によって導入されたアルファベット Σ 上の k-reversible オートマトンを拡張し, SPF 集合上で遷移する (l,k)-extended reversible((l,k)-extR と略記) オートマトンを定義する。(l,k)-extR オートマトンで受理される言語を (l,k)-extR 言語と呼ぶ。Angluin の k-reversible 言語は, SPF 集合が Σ である場合の $(0,k)$-extR 言語に対応する。$(0,k)$- extR 言語は, reversible 言語であるが, $l \geq 1$ ならば, reversible ではない (l,k)-extR 言語が存在する。本稿では, (l,k)-extR 言語の特徴付け定理を与え, これらの言語族の間の包含関係を調べる。本稿の後半は, (l,k)-言語の族が正例から多項式時間で推論可能となることを示す。

2 Extended Reversible 言語

Σ 上の文字列を語と呼び, 語の集合を Σ^*, 空語 λ を除く語の集合を Σ^+ で表す。Σ^* の部分集合を言語という。言語 L_1, L_2 に対して連接 $L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$ を定義する。言語 L を

n 回連接したものを L^n とし、さらに $L^+ = \bigcup_{n=1}^{\infty} L^n$, $L^* = L^+ \cup \{\lambda\}$ とする。また, 有限集合 S の要素の個数を $\sharp S$ とする。

語 $w \in \Sigma^+$ の長さを $|w|$ とかく。語 u が語 w の prefix であるとは $w = uv$ となる語 $v \in \Sigma^*$ が存在することである。

2.1 Extended 有限オートマトン

まず, 文字単位ではなく, 語単位で遷移するオートマトンを定義する。

定義 2.1. 語の集合 $W \subseteq \Sigma^+$ が simple prefix-free(SPF と略す) であるとは, 任意の異なる $u_1, u_2 \in W$ の最初の文字が互いに異なることをいう。

上記の定義から, SPF 集合 W は有限集合であり, 任意の語 $w \in W^*$ は, W の語で一意的に分解される。すなわち, $w = uv, u \in W^*$ ならば, $v \in W^*$ である。言語 L が $L \subseteq W^*$ を満たすとき, W 上の言語という。以下 W, W' が SPF であることを仮定する。

定義 2.2. Σ 上の extended 有限オートマトンとは, 次の条件を満たす 5 つ組 $M = (Q, W, \Delta, Q_0, F)$ をいう。(1) Q は状態の有限集合, (2) $W \subseteq \Sigma^+$ は語の有限集合, (3) $Q_0 \subseteq Q$ は初期状態の集合, (4) $F \subseteq Q$ は受理状態の集合, (5) $\Delta \subseteq Q \times W \times Q$ 遷移の有限集合。

$\sharp Q_0 \leq 1$ かつ任意の $q \in Q, u \in W$ に対して, $\sharp\{(q,u,p) \in \Delta \mid p \in Q\} \leq 1$ のとき, M は決定性という。Δ の代わりに遷移部分関数 $\delta : Q \times W \to 2^Q$ を用いることもある。δ に対し $\delta^* : Q \times W^* \to 2^Q$ を帰納的に $\delta^*(q, \lambda) = q, \delta^*(q, wu) = \delta(\delta^*(q, w), u)$ $(q \in Q, u \in W, w \in W^*)$ と定義する。この $*$ は省略されることがある。また, $\{q\}$ のように状態の集合がただ 1 つの元からなるとき q と略記する。

extended 有限オートマトン M で受理される言語を $L(M)$ で表す。明らかに, $L(M)$ は W 上の言語で正則である。上記の extended 有限オートマトンを W 上のオートマトン, もしくはただ単にオートマトンと呼ぶ。

W 上の言語 L に対して, $\Pr^W(L)$ を以下のように定義する。

$$\Pr^W(L) = \{u \in W^* \mid \exists v \in W^* \ s.t. \ uv \in L\}$$

また, 語 w に対する L の tail 集合を

$$T_L(w) = \{u \in \Sigma^* \mid wu \in L\}$$

とする。

言語 L が正則であるための必要十分条件は, $\{T_L(w) \mid w \in \Sigma^*\}$ が有限であり, また, 正則言語を受理する最小オートマトン (状態数が最小) は, tail 集合を用いる定義できることが知られている [1]。同様の方法で, 与えられた W 上の正則言語を受理する最小オートマトンを導入する。

定義 2.3. 言語 $L \neq \varnothing$ を W 上の正則言語とする。W 上の canonical オートマトン $M_0^W(L) = (Q, W, \delta, q_0, F)$ を次のように定義する。

$$Q = \{T_L(w) \mid w \in \Pr^W(L)\}$$
$$q_0 = T_L(\lambda,), F = \{T_L(w) \mid w \in L\}$$
$$\delta(T_L(w), u) = T_L(wu), \ w, wu \in \Pr^W(L), \ u \in W$$

定理 2.4. W 上の正則言語 L に対して, canonical オートマトン $M_0^W(L)$ は L を受理する W 上の最小オートマトンである。

定義 2.5. SPF W, W' に対して, 次の順序関係を導入する。

$$W \preceq W' \Longleftrightarrow W \subseteq W'^+$$

定義 2.6. SPF W が言語 L の SPF 生成集合であるとは, $L \subseteq W^*$ かつ, 任意の $W' \subsetneq W$ に対して, $L \not\subseteq W'^*$ であることをいう。L の SPF 生成集合全体を $\mathcal{W}^L$ であらわす。

以下の結果が得られている。

補助定理 2.7. (Watanabe [11]) 任意の言語 L に対して, $\mathcal{W}^L$ は有限束である。

$\mathcal{W}^L$ の最大元及び最小元をそれぞれ, $W^L_{\sup}, W^L_{\inf}$ と書く。明らかに, $W^L_{\sup}$ は L の語に出現する Σ の文字の集合である。

補助定理 2.8. (Watanabe [11]) L, L' を言語とすると, $L \subseteq L'$ ならば $W^L_{\inf} \preceq W^{L'}_{\inf}$

補助定理 2.9. L を正則言語とする。任意の $W \in \mathcal{W}^L$ に対して, L を受理する W 上の canonical オートマトンが存在する。

上記の結果から, 任意の正則言語 L に対して, L を受理する $W^L_{\inf}$ 上のオートマトン及び $W^L_{\sup}$ 上のオートマトンが存在する。$M^L_{\inf}$ と $M^L_{\sup}$ をそれぞれ, L を受理する $W^L_{\inf}$ と $W^L_{\sup}$ 上の canonical オートマトンとする。

定理 2.4, 補助定理 2.9 から, 次の定理が得られる。

定理 2.10. (1) $M^L_{\inf}$ は, L を受理する SPF 集合上のオートマトンの中で最小の状態数をもつ。(2) $M^L_{\sup}$ は, 通常のアルファベット Σ 上の最小オートマトンである。

W 上の正則言語 L に対して, 次の集合を定義する。

$$\Pr^W_l(L) \quad = \{w \in \Pr^W(L) \mid |w|^W = l\}$$

ここで, $|w|^W$ は語 $w \in W^*$ の W における長さを意味する。すなわち, $w = u_1u_2\cdots u_n (u = i \in W)$ に対して, $|w|^W = n$ とする。また, W のオートマトン M が与えられているとき, 上記の $\Pr^W_l(L(M))$ を単に $\Pr^W_l(M)$, と表す。

$M = (Q, W, \delta, q_0, F)$ を決定性オートマトンとする。$\delta^*(q, v) = q$ となる語 $v \in W^+$ が存在しないとき, 状態 q は loop-free という。

定義 2.11. $M = (Q, W, \delta, q_0, F)$ を W 上の決定性オートマトンとする。$\delta(q_0, v) \in Q$ となる $v \in W^*$ に対して, 次のような M の部分オートマトン $M_v = (Q_v, W_v, \delta_v, q_v, F_v)$ を定義する。

(1) $Q_v = \{q \in Q \mid \exists w \in W^* \ s.t. \ \delta^*(q_v, w) \in Q\}$
(2) $W_v = \{u \in W \mid \exists q \in Q_v \ s.t. \ \delta(q, u) \in Q_v\}$
(3) $\delta_v(q, u) = q'$
$\Leftrightarrow \delta(q, u) = q', (q, q' \in Q_v, u \in W_v)$
(4) $F_v = Q_v \cap F$

定義 2.12. $l \geq 1$ とする。W 上の決定性オートマトン $M = (Q, W, \delta, q_0, F)$ が l-tree 型オートマトンというのは, 任意の $i < l$ と任意の語 $v \in \Pr^W_i(M)$ に対して, 状態 $\delta(q_0, v)$ が roop-free であることをいう。

定理 2.13. $l \geq 1$ とする。正則言語 L の canonical オートマトン $M = M_{\inf}(L)$ に対して, 次の条件を満たす $W^L_{\inf}$ 上の l-tree 型オートマトン M' が存在する。任意の $v \in \Pr^W_l(L)$ に対して, M' の部分オートマトン M'_v は M の部分オートマトン M_v と状態の名前を除いて等しく, 各 M'_v の状態の集合は交わりがない。ただし, $W = W^L_{\inf}$ とする。

上記の定理で述べた M' を言語 L の l-tree 型 canonical オートマトンと呼ぶ。

定義 2.14. $M = (Q, W, \Delta, q_0, F)$,
$M' = (Q', W', \Delta', q_0, F)$, $W \preceq W'$ とする。
M' が M の W' による refined であるとは, $u \in W$, $u = u_1u_2\cdots u_n (u_j \in W')$ のとき, $p, q \in Q, (p, u, q) \in \Delta$ であることと $p, q_1, \cdots, q_{n-1}, q \in Q'$ かつ $(p, u_1, q_1), (q_1, u_2, q_2), \cdots, (q_{n-1}, u_n, q) \in \Delta'$ となることが同値となることである。

明らかに, $L(M) = L(M')$ となる。

2.2 Extended Zero Reversible 言語

W 上のオートマトン $M = (Q, W, \delta, Q_0, F)$ の reversal オートマトンとは, $M^r = (Q, W, \delta^r, F, Q_0)$ である。ただし, $\delta^r(p, u) = \{q \mid p \in \delta(q, u)\}$, $(p \in Q, u \in W)$ とする。M^r は, M の遷移の矢印を逆向きにした遷移図をもつ。

定義 2.15. W 上のオートマトン M=(Q, W, δ, Q_0, F) が extended zero reversible(extZR と略す) オートマトンであるとは, M と M^r がともに決定性オートマトンとなることである。

定義 2.16. 言語 L が extended zero reversible (extZR と略す) 言語であるとは, L を受理する extZR オートマトン M が存在することである。M が W 上のオートマトンのとき, L を W 上の extZR 言語という。extZR 言語の族を $ext\mathcal{ZR}$ とする。

上記の定義で, $W = \Sigma$ とすると, Σ 上の extZR オートマトンは Angluin [3] が導入した zero reversible オートマトン (ZR オートマトンと略す) となる。

ZR 言語の族を $\mathcal{ZR}$ とする。明らかに, ZR 言語は Σ 上の extZR 言語でもある。しかし, 逆は成り立たない。例えば, $\Sigma = \{a, b\}$ 上の言語 $L = \{b, ab\}$ は extZR 言語であるが, ZR 言語ではない。従って, 次の定理が成り立つ。

定理 2.17. $\mathcal{ZR} \subsetneq ext\mathcal{ZR}$

2.3 Extended Reversible 言語

まず, Angluin による k-reversible オートマトンを SPF 集合上のオートマトンに拡張する。

$M=(Q,W,\delta,q_0,F)$ を W 上のオートマトンとし，k を非負整数とする。語 $w\in W^*$ が状態 $q\in Q$ の k-leader というのは，$\delta^r(w_W^r,q)\neq\varnothing$ であることをいう。ただし，$w=u_1u_2\cdots u_n(u_i\in W)$ に対して，$w_W^r=u_n\cdots u_2u_1$ とする。

定義 2.18. W 上のオートマトン $M=(Q,W,\delta,q_0,F)$ が k-extended reversible(k-ext$R$ と略記) オートマトンというのは，異なる2つの状態 $p,q\in Q$ が同じ k-leader をもつならば，p,q が共には受理状態でなくかつ，$\delta(p,u)\neq\delta(q,u)(u\in W)$ となることをいう。

定義 2.19. W 上のオートマトン $M=(Q,W,\delta,q_0,F)$ が (l,k)-extended reversible オートマトン ((l,k)-extR オートマトンと略す) であるとは，任意の $v\in \mathrm{Pr}_l^W(M)$ に対して，各部分オートマトン M_v の状態の集合が互いに素となる k-extZR オートマトンとなることである。

(l,k)-extR オートマトンに受理される言語を (l,k)-extended reversible 言語 ((l,k)-extR 言語) と呼ぶ。(l,k)-extR 言語の全体を (l,k)-ext$\mathcal{R}$ で表し，ext$\mathcal{R}=\cup_{l,k\geq 0}(l,k)$-ext$\mathcal{R}$ とする。ext$\mathcal{R}$ に含まれる言語を extended reversible 言語 (ext$R$ 言語 と略記) と呼ぶ。

(l,k)-extR 言語族の間には次の包含関係が成り立つ。ここで，正則表現 $a^{l+k+1}a^*$ は，$(l,k+1)$-extR 言語かつ $(l+1,k)$-extR 言語を表す。しかし，(l,k)-extR 言語ではないことに注意する。

定理 2.20. $l,k\geq 0$ に対して，以下の包含関係が成り立つ。

$$(l,k)\text{-ext}\mathcal{R}\subsetneq(l+1,k)\text{-ext}\mathcal{R},$$
$$(l,k)\text{-ext}\mathcal{R}\subsetneq(l,k+1)\text{-ext}\mathcal{R},$$

前節の定理 2.13 より，extR 言語 L に対して，ある l が存在して l-tree 型 canonical オートマトン が存在する。

定理 2.21. 正則言語 L が (l,k)-extR 言語であるための必要十分条件は，L の l-tree 型 canonical オートマトンが (l,k)-extR オートマトンとなることである。

定理 2.22. ((l,k)-extR 言語の特徴付け定理)
L を正則言語,$W=W_{\mathrm{inf}}^L$ とする。L が (l,k)-extR 言語であるための必要十分条件は，任意の $w,u_1,u_2,w',v\in W^*$ に対して

$$|w|^W=l,\ |w'|^W=k,\ wu_1w'v,wu_2w'v\in L \Rightarrow\quad T_L(wu_1w')=T_L(wu_2w')$$

定理 2.23. (1) 任意の $(0,k)$-extR 言語は，reversible 言語である。(2) 任意の k-reversible 言語は，$(0,k)$-extR 言語である。逆は成り立たない。

定理 2.24. $l\geq 1,k\geq 0$ とする。reversible ではない (l,k)-extR 言語が存在する。

3 extR 言語の正例からの極限同定

本節では、(l,k)-extended reversible 言語の正例からの効率的な帰納アルゴリズムについて考察する。

3.1 正例からの多項式時間帰納推論

本節では，(l,k)-extR 言語 L の正の事例，すなわち，L に含まれる語の例だけから効率的に L を受理する (l,k)-extR オートマトンを推論・学習する問題を考える。この問題を正例からの帰納推論 ([5]) の枠組みに基づいて定式化する。「入力要求 $\Rightarrow$ 計算 $\Rightarrow$ 出力」という過程を繰り返すアルゴリズムを推論アルゴリズムという。推論アルゴリズムの出力は，推測或いは仮説とも呼ばれる。本稿では，推論アルゴリズムに入力として提示されるのは，未知である (l,k)-extR 言語の正提示であり，出力は，(l,k)-extR オートマトンである。一般に，言語 L の正提示とは，語の無限列 $w_1,w_2,\cdots$ で $L=\{w_n\mid n\geq 1\}$ を満たすものをいう。(l,k) 言語 L の正提示 $\sigma=w_1,w_2,\cdots$ に対して，その初期断片 $\sigma[n]=w_1,w_2,\cdots,w_n$ に対するアルゴリズム $\mathcal{IA}$ の推測を M_n とする。入力 σ に対する推測の列 $M_1,M_2,\cdots$ がある1つの (l,k)-extR オートマトン M に収束し，$L(M)=L$ ならば，その推論アルゴリズム $\mathcal{IA}$ は，目標言語 L の正提示 σ から L を極限同定するという。L の任意の正提示に対して，L を極限同定するとき，$\mathcal{IA}$ は目標言語を正例から極限同定するという。また，アルゴリズム $\mathcal{IA}$ が，任意の (l,k)-extR 言語を正例から極限同定するとき，(l,k)-extR 言語の族は，推論アルゴリズム $\mathcal{IA}$ によって正例から推論可能であるという。また，

・各時点 n におけるアルゴリズムの出力 M_n がそれまでに入力された事例（語）を受理するオートマトンならば，$\mathcal{IA}$ は無矛盾という。

・$w_n\in L(M_{n-1})$ ならば，$M_n=M_{n-1}$ のとき，$\mathcal{IA}$ は保守的という。

・w_n を受け取ってから, 推測 M_n を出力するまでの計算時間がそれまでに入力された語の長さの総和に関する多項式のオーダーで抑えられるとき, $\mathcal{IA}$ は多項式時間推論アルゴリズムという。
(l,k)-extR 言語の族は, 無矛盾, 保守的かつ多項式時間推論アルゴリズムによって正例から推論可能であるとき, 正例から多項式時間推論可能という。

次の概念は, Angluin [3] で導入された。効率的な推論アルゴリズムの構築に重要な役割を果たす概念である。

定義 3.1. L を (l,k)-extZR 言語とする。有限集合 $S \subseteq \Sigma^*$ が L の ((l,k)-extR における) characteristic sample であるとは,(1) $S \subseteq L$, (2) 任意の extZR 言語 L' に対して, $S \subseteq L'$ ならば $L \subseteq L'$ となることである。

3.2 (l,k)-extR 言語の推論アルゴリズム

先ず, (l,k)-extR 言語 L の (l,k)-extZR における characteristic sample を与える。

$M=(Q,W,\delta,q_0,F)$ を extR オートマトンとする。任意の状態 $q \in Q$ に対して次のような集合を定義する。
・Pre(q) は, 初期状態 q_0 から q へ最短の遷移回数で遷移する語の集合である。
・Post(q) は, 状態 q から最終状態 $q_f \in F$ へ最短の遷移回数で遷移する語の集合の $q_f \in F$ についての和。

定義 3.2. L を (l,k)-extR 言語とする。L の l-tree 型 canonical オートマトンを $M=(Q,W,\delta,q_0,F)$ とする。ただし, $W=W_{\inf}^L$ とする。この M を用いて, L の有限部分集合 S_L を次のように定義する。ただし,$W=W_{\inf}^L$ とする。

$$S_L = \bigcup_{q\in Q, u\in W\cup\{\lambda\}} \mathrm{Pre}(q)\cdot u\cdot \mathrm{Post}(\delta(q,u)))$$

補助定理 3.3. 任意の (l,k)-extR 言語 L に対して, $W_{\inf}^L = W_{\inf}^{S_L}$ が成り立つ。

定理 3.4. L を (l,k)-extR 言語とする。有限集合 S_L は (l,k)-extR 言語族における characteristic sample である。

以下, ある (l,k)-extR 言語の語の無限列 $w_1,w_2,\cdots$ に対して, 次々と推測を生成し出力する推論アルゴリズムを与える。

Algorithm IA
入力：語の無限列 ,$w_1,w_2,\cdots$
出力：(l,k)-extR オートマトンの無限列
begin
initialize $i:=0$;
$Q_0:=\{q_0\}; W_0:=\varnothing; \Delta_0:=\varnothing;$
$M_0:=(Q_0,W_0,\Delta_0,q_0,\varnothing);$
repeat
 $i:=i+1;$
 let $M_{i-1}=(Q_{i-1},W_{i-1},\Delta_{i-1},q_0,F_{i-1})$ be the current conjecture ;
 read the next data w_i;
 if $w_i \in L(M_{i-1})$ **then** $M_i:=M_{i-1}$
 else
  begin
   if $w_i \notin W_{i-1}^*$
    begin
     W_i :=**UPDATE**(W_{i-1},w_i);
     M_{i-1} := **REFINE**(W_i,M_{i-1});
     M_{i-1} := **BIND**(M_{i-1});
    end
   M_i := **PARSE**(W_i,M_{i-1},w_i);
   M_i := **CONSTRUCT**(M_i,w_i);
  end ;
output M_i as the i-th conjecture
forever
end

ただし,

(1) **UPDATE**(W_{i-1},w_i) : 入力 w_i を生成するため, 現在の SPF W_{i-1} を更新し, 新しい SPF $W_i = W_{\inf}^{W_{i-1}\cup\{w_i\}}$ を出力する。

(2) **REFINE**(W_i,M_{i-1}): W_{i-1} 上の k-extR オートマトン M_{i-1} を W_i 上の refined オートマトンに変換する。この操作の前後で M_{i-1} の受理する言語は変化しない。

(3) **BIND**(M_{i-1}):新しい SPF W_i と **REFINE** によって, 高さ l の tree 構造が変化する。$|v|^W=l$ に対して $(M_{i-1})_v$ が k-extR となるようまで合併を繰り返す。

(4) **PARSE**(W_i,M_{i-1},w_i) : (2) では, W_{i-1} 上の

オートマトンを W_i のオートマトンへ変換した。ここでは, 変換された W_i 上のオートマトンが入力 $w_i(W_i$ 上の語である) を受理するように, 必要な状態, 受理状態及び遷移を追加する。

(5) **CONSTRUCT**(M_i, w_i): $|w_i|^{W_i} \geq l$ のとき, $v(\in W_i^+)$ を w_i の prefix であり,$|v|^{W_i} = l$ とする。(4) で得られた W_i 上のオートマトンの v に対する部分オートマトン $(M_i)_v$ は必ずしも k-extR オートマトンとは限らない。ここでは, そのオートマトンを k-extR オートマトンへと変換する。$|w_i|^{W_i} < l$ のとき,M_i を出力する。

(1),(2),(4) の手続きは, Tanida & Yokomori [10], Yokomori [13] によって多項式時間で計算する効率的なアルゴリズムが与えられている。また, (2),(4) に関しては, Angluin [3] による k-reversible 言語の推論アルゴリズムを SPF 上のアルゴリズムに変更すればよい。

入力 $w_1, w_2, \cdots w_n$ に対する推論アルゴリズムの出力を M_n とすると次の結果が成り立つ。

補助定理 3.5. $L(M_n) \subseteq L(M_{n+1}), \quad (n \in N)$

補助定理 3.6. IA の n 番目の出力は, $S_n = \{w_1, \cdots, w_n\}$ を含む最小の (l,k)-extR 言語を受理する l-tree 型 canonical オートマトンである。

定理 3.7. 手続き **IA**(S) の計算時間は, $O(nm)$ である。ただし, $m = \sharp S, n = \max\{|w| \mid w \in S\}$ とする。

これらの結果から, 次の主定理が示される :

定理 3.8. (l,k)-extR 言語の族は, 正例から多項式時間推論可能である。

参考文献

[1] Aho, a.v., Hopcroft, J.E., & Ullman, J.D. "*The Design and Analysis of Computer Algorithms.*", Addison-Wesley, Reading, Mass., 1974.

[2] D. Angluin. "*Inductive Inference of Formal Languages form Positive Data*", Information and Contoral., vol. 45. 117-135, 1980.

[3] D. Angluin. "*Inference of Reversible Language*", Journal of the Association for Computing Machinary., vol. 29, No.3, pp. 741-765, July 1982.

[4] M. D. Davis, R. Siegal & E. J. Weyuker. "*Computability, Complexity, and Languages*", Academic Press., 1994.

[5] E. M. Gold. "*Language Identification in the Limit*", Information and Control., vol. 10, pp. 447-474, 1967.

[6] T. Head, S. Kobayashi & T. Yokomori. "*Locally, reversibility, and beyond: Learning languages from from positive data*", in Proceedings of the 9th International Conference on Algorithmic Learning Theory., Lecture Notes in Artificial Intelligence., vol. 1501, pp. 191-204, 1998.

[7] E. Makinen. "*The grammatical inference problem for the Szilard languages of linear grammars*", Inpormation Processing Letters., Vol. 36, pp.203-206, 1990.

[8] K. Sato & M. Sato. "正データからの *Simple Regular* 言語の多項式時間帰納推論", 数理解析研究所講究録., Vol. 906, pp. 290-227, 1995.

[9] M. Sato. "*Inductive Inference of Formal Language*", Bull. Inf. Cybern., Vol. 27-1, pp. 85-106, 1995.

[10] N. Tanida & T. Yokomori. "*Polynomial-time identification of strictly regular languages in the limit*", IEICE Transactions on Information and Systems., Vol. E75-D, pp. 125-132, 1992.

[11] N. Watanabe & M. Sato. "完全データからの *Simple Regular* 言語の多項式時間帰納推論", 数理解析研究所講究録., Vol. 906, pp. 228-235, 1995.

[12] N. Watanabe & M. Sato. "*k-Simple Regular* 言語の多項式時間帰納推論", 情報基礎論ワークショップ., pp. 97-102, 1995.

[13] T. Yokomori. "*On Polynomial-time learnability in the limit of strictly deterministic automata*", Machine Learning., Vol. 19, pp. 153-179, 1995.