数理解析研究所講究録
1041 巻 1998 年 143-150

Prefix Free Generating Sets of Formal Languages

Mikiharu TERADA, Yasuhito MUKOUCHI and Masako SATO
寺田 幹治　　向内 康人　　佐藤 優子

Department of Mathematics and Information Sciences
Osaka Prefecture University, Sakai, Osaka, 599-8531, Japan

Abstract. This paper deals with a particular type of generating set, called *prefix free*, for a language. Given a language L over an alphabet Σ, a set G is a *generating set* of L, denoted by $L \sqsubseteq G$, if $L \subseteq G^+$. It is well known that a prefix free set G has the property of *unique decipherability* for all strings in G^+.

We first show that the class $\mathcal{PF}$ of all prefix free sets is a complete lattice under the partial order $\sqsubseteq$. In terms of the result, it is shown that for a language L, the class $\mathcal{PFG}_L$ of all prefix free and reduced generating sets for L is also a complete lattice under the relation $\sqsubseteq$ and has the least element $G_L^{\inf}$ and the greatest element $G_L^{\sup}$. Especially we are concerned with the least element $G_L^{\inf}$ of the lattice. If $G_L^{\inf}$ is finite, it has the fewest number of strings among $\mathcal{PFG}_L$. We give the necessary and sufficient condition for $G_L^{\inf}$ to be finite. Moreover, we present a polynomial time algorithm for computing $G_L^{\inf}$ for a finite language L. For an infinite language, we consider a problem of identifying $G_L^{\inf}$ in the framework of *identification in the limit* proposed by Gold for language learning, and give a polynomial time learning algorithm for computing $G_L^{\inf}$, provided that the target $G_L^{\inf}$ is finite.

1. Introduction

In this paper, we consider a particular set of strings that generates every string in a formal language L over an alphabet Σ. A set S of strings is a *generating set* of a language L, denoted by $L \sqsubseteq S$, if $L \subseteq S^+$, that is, every string of L is represented as a concatenation of strings in S. For instance, let $\Sigma = \{a, b\}$ and $L = \{(aab)^i b(aa)^j \mid i, j \in N\}$ be a regular language. Then the sets $\{aab, b, aa\}$ and $\{aa, b\}$ of strings are generating sets of L. Clearly the alphabet Σ is a generating set of any language L consisting of nonempty strings, and moreover the language L itself is a generating set of L.

We introduce a particular type of generating set, called *prefix free*, for a language. A set S of strings is prefix free, if for any string of S, there is no proper prefix in S of the string. In the above example, the string aa is a proper prefix of the string aab and thus $S = \{aab, b, aa\}$ is not prefix free. For a string u, a sequence $u_1, u_2, \cdots, u_n$ of strings in S is a *factorization* of u in S, if $u = u_1 u_2 \cdots u_n$. A prefix free set S has the property of so-called *unique decipherability* in terminology of coding theory in the sense that any string in S^+ has a unique factorization in S. In coding theory, such a set is called *code* and has been discussed in connection with unique decipherability. Refer in detail to e.g. [3].

In this paper, we introduce a binary relation $\sqsubseteq$ for subsets of Σ^+. First we show that the class $\mathcal{PF}$ of all prefix free sets over Σ has a lattice structure with respect to the relation $\sqsubseteq$, where Σ is the greatest element of $\mathcal{PF}$.

Next, we investigate the class $\mathcal{PFG}_L$ of all prefix free and reduced generating sets of a given language L. We show that the class $\mathcal{PFG}_L$ is also a lattice with both the least element and the greatest element of the class. In particular, we are interested in the least prefix free generating

set, denoted by $G_L^{\inf}$, of a language L. We present an explicit expression of $G_L^{\inf}$ by introducing some operation for sets of strings.

Finally, we present a polynomial time algorithm for computing the least prefix free generating set of a finite language. Furthermore, for an infinite language L, we propose a polynomial time inductive inference algorithm for identifying the least prefix free generating set $G_L^{\inf}$ *in the limit* in terms of the framework for language learning introduced by Gold[4].

In article[6], Watanabe has shown that a simple prefix free set (a generating set of L) has a finite lattice structure. Another, by article[5], Yokomori has discussed inductive learnability of languages with *strict prefix property* from the viewpoint of polynomial-time learnability in terms of *strictly deterministic automata.* In this note, we shall extend this strict prefix property to the prefix free property.

2. Prefix Free Generating Sets of Formal Language

2.1. Preliminaries

We start with some basic definitions and notations used in this paper.

Let Σ be a finite *alphabet.* Let Σ^* be the set of all strings over Σ, and Σ^+ be the set of all finite *nonempty* strings over Σ. The *empty* string is denoted by λ. For a string $w \in \Sigma^*$, $|w|$ denote the *length* of w. In particular, the length of λ is 0.

The concatenation of strings u and v is denoted by uv. For a string w, a string $u \in \Sigma^*$ is a *prefix* of w, if there is a string $v \in \Sigma^*$ such that $w = uv$, particularly when $v \in \Sigma^+$, u is a *proper* prefix of w.

For subsets S_1 and S_2 of Σ^*, let us denote $S_1S_2 = \{xy \mid x \in S_1, y \in S_2\}$. Define $S^{k+1} = SS^k$ for nonnegative integer k, where $S^0 = \{\lambda\}$. Let $S^* = \bigcup_{k=0}^{\infty} S^k$, and $S^+ = \bigcup_{k=1}^{\infty} S^k$.

Let N be the set of nonnegative integers, and for a finite set S, $|S|$ be the cardinality of S.

For sets $S_1, S_2 \subseteq \Sigma^+$, we define the following binary relation:

$$S_1 \sqsubseteq S_2 \text{ if and only if } S_1 \subseteq S_2^+.$$

Clearly $S \sqsubseteq \Sigma$ for any set $S \subseteq \Sigma^+$. As easily seen, the relation $\sqsubseteq$ is reflective and transitive but not antisymmetric. Indeed, for $S_1 = \{a, b\}$ and $S_2 = \{a, b, ab\}$, clearly $S_1 \sqsubseteq S_2$ and $S_2 \sqsubseteq S_1$ but $S_1 \neq S_2$. As shown below, the relation $\sqsubseteq$ is antisymmetric for prefix free sets.

By the definition of $\sqsubseteq$, it immediately follows that:

Lemma 2.1. *Let $\mathcal{S}$ be a set of subsets of Σ^+, and S_1, S_2 and T be subsets of Σ^+. Then*
(1) if $S \sqsubseteq T$ for any $S \in \mathcal{S}$, then $\bigcup_{S\in\mathcal{S}} S \sqsubseteq T$ and $\bigcap_{S\in\mathcal{S}} S \sqsubseteq T$,
(2) if $S_1 \subseteq S_2$ and $S_2 \sqsubseteq T$, then $S_1 \sqsubseteq T$.

2.2. Prefix Free Sets

Definition 2.1. A set $S \subseteq \Sigma^+$ is *prefix free*, if any string in S is not proper prefix of another string in S. By $\mathcal{PF}$ we denote the set of all prefix free sets.

As well known in coding theory, a prefix free set has the property of *unique decipherability.* That is, any message (string) has a unique factorization in terms of strings in the prefix free set. Using the property, it is easily shown that:

Lemma 2.2. *The set $(\mathcal{PF}, \sqsubseteq)$ is a partially ordered set.*

Lemma 2.3. *Let S be a prefix free set. Then*
(1) for any string $w \in S^+$, there is a unique factorization $u_1, u_2, \cdots, u_n$ of strings in S such that $w = u_1 u_2 \cdots u_n$,
(2) for any strings $u, v \in \Sigma^+$, if $uv, u \in S^+$ then $v \in S^+$.

For a set $S \subseteq \Sigma^+$, we define

$$\mathrm{Pre}(S) = \{u \in S \mid \text{ there is no proper prefix of } u \text{ in } S\}.$$

By the above definition, the next result immediately follows:

Lemma 2.4. *For any set $S \subseteq \Sigma^+$, the set $\mathrm{Pre}(S)$ satisfies the following conditions:*
(1) $\mathrm{Pre}(S) \subseteq S$. (2) $\mathrm{Pre}(S) \in \mathcal{PF}$.
(3) For any string $w \in S$, there is a prefix $u \in \mathrm{Pre}(S)$ of w.

Definition 2.2. Define a binary operation $O(x, y)$ for two strings $x, y \in \Sigma^+$, and a set operation $O(S)$ for a set $S \subseteq \Sigma^+$:

$$O(x,y) = \begin{cases} \{x\}, & \text{if } x = y, \\ \{x, y'\}, & \text{if } \exists y' \in \Sigma^+ \text{ s.t. } y = xy', \\ \{x', y\}, & \text{if } \exists x' \in \Sigma^+ \text{ s.t. } x = yx', \\ \{x, y\}, & \text{otherwise,} \end{cases}$$

$$O(S) = \bigcup_{(x,y) \in S \times S} O(x,y).$$

For a string w, a set $\{x, y\}$ of two strings is a *direct ancestor* of w, if $w \in O(x, y)$ and $w \neq x, y$. Furthermore, we define $O^0(S) = S$ and $O^{n+1}(S) = O(O^n(S))$ $(n \in N)$. And define the closure $O^*(S)$ as follows:

$$O^*(S) = \bigcup_{n \in N} O^n(S).$$

Clearly $O(O^*(S)) = O^*(S)$ and if $O^n(S) = O^{n+1}(S)$ for some $n \in N$ then $O^*(S) = O^m(S)$ for any $m \geq n$.

As a direct result of the definition of $O(S)$, it follows that:

Lemma 2.5. *Let S, S_1 and S_2 be subsets of Σ^+. Then*
(1) $S \subseteq O^n(S) \subseteq O^{n+1}(S) \subseteq O^*(S)$ for any $n \in N$,
(2) $S_1 \subseteq S_2$ implies $O^*(S_1) \subseteq O^*(S_2)$.

Note that for any $w \in O^{n+1}(S) - O^n(S)$, there is a direct ancestor $\{x, y\} \subseteq O^n(S)$ of w, and moreover x (or y) is contained in $O^n(S) - O^{n-1}(S)$, where $O^{-1}(S) = \phi$.

Lemma 2.6. *Let S and T be subsets of Σ^+ such that $S \sqsubseteq T$. If T is prefix free, then $O^*(S) \sqsubseteq T$ and $\mathrm{Pre}(O^*(S)) \sqsubseteq T$.*

Lemma 2.7. *For any set $S \subseteq \Sigma^+$, $O^*(S) \sqsubseteq \mathrm{Pre}(O^*(S))$.*

Hereafter, we investigate a lattice structure of $\mathcal{PF}$. For a subset $\mathcal{P}$ of $\mathcal{PF}$, $\sup(\mathcal{P})$ and $\inf(\mathcal{P})$ denote the least upper bound and the greatest lower bound of $\mathcal{P}$ in $\mathcal{PF}$ under the partially ordered relation $\sqsubseteq$, respectively.

Lemma 2.8. *For any $\mathcal{P} \subseteq \mathcal{PF}$, $\sup(\mathcal{P}) = \mathrm{Pre}(O^*(\bigcup_{S \in \mathcal{P}} S))$.*

Proof. Put $T = \mathrm{Pre}(O^*(\bigcup_{S\in\mathcal{P}} S))$. By Lemma 2.7, $O^*(\bigcup_{S\in\mathcal{P}} S) \sqsubseteq T$, and thus by Lemma 2.1(2), $S \sqsubseteq T$ for any $S \in \mathcal{P}$. Hence T is an upper bound of $\mathcal{P}$.

Let $T' \in \mathcal{PF}$ be any upper bound of $\mathcal{P}$. Then by Lemma 2.1(1), $\bigcup_{S\in\mathcal{P}} S \sqsubseteq T'$. Since T' is prefix free, it implies by Lemma 2.6 that $T \sqsubseteq T'$. Hence T is the least upper bound of $\mathcal{P}$ in $\mathcal{PF}$. ∎

As mentioned before, Σ is a prefix free set, and $S \sqsubseteq \Sigma$ for any $S \in \mathcal{PF}$. Hence Σ is the greatest element of $\mathcal{PF}$, i.e., $\sup(\mathcal{PF}) = \Sigma$.

Lemma 2.9. *For any* $\mathcal{P} \subseteq \mathcal{PF}$, $\inf(\mathcal{P}) = \mathrm{Pre}(O^*(\bigcap_{S\in\mathcal{P}} S^+))$.

Proof. Put $T = \mathrm{Pre}(O^*(\bigcap_{S\in\mathcal{P}} S^+))$. By Lemma 2.7, $T \subseteq O^*(\bigcap_{S\in\mathcal{P}} S^+) \sqsubseteq T$.

We first prove that T is a lower bound of $\mathcal{P}$, i.e., $T \sqsubseteq S$ for any $S \in \mathcal{P}$. Assume that $\bigcap_{S\in\mathcal{P}} S^+ \subsetneq O(\bigcap_{S\in\mathcal{P}} S^+)$ and let $w \in O(\bigcap_{S\in\mathcal{P}} S^+) - \bigcap_{S\in\mathcal{P}} S^+$. Then there is a direct ancestor $\{u, v\} \subseteq \bigcap_{S\in\mathcal{P}} S^+$ such that $u = vw$. Since each $S \in \mathcal{P}$ is prefix free, by Lemma 2.3(2), we have $w \in \bigcap_{S\in\mathcal{P}} S^+$, and a contradiction. Hence we have $O(\bigcap_{S\in\mathcal{P}} S^+) = \bigcap_{S\in\mathcal{P}} S^+$, and thus $O^*(\bigcap_{S\in\mathcal{P}} S^+) = \bigcap_{S\in\mathcal{P}} S^+$. Since $T \subseteq O^*(\bigcap_{S\in\mathcal{P}} S^+)$, $T \subseteq \bigcap_{S\in\mathcal{P}} S^+$. Consequently $T \sqsubseteq S$ for any $S \in \mathcal{P}$, i.e., T is a lower bound of $\mathcal{P}$.

Nextly, we prove that T is the greatest lower bound of $\mathcal{P}$. Let $T' \in \mathcal{PF}$ be any lower bound of $\mathcal{P}$. Then $T' \subseteq S^+$ for any $S \in \mathcal{P}$, and thus $T' \subseteq \bigcap_{S\in\mathcal{P}} S^+$. Since $\bigcap_{S\in\mathcal{P}} S^+ \sqsubseteq T$, we get $T' \sqsubseteq T$. Therefore T is the greatest lower bound of $\mathcal{P}$. ∎

By Lemma 2.8 and Lemma 2.9, the next result on $\mathcal{PF}$ immediately follows:

Theorem 2.10. *The set* $(\mathcal{PF}, \sqsubseteq)$ *is a complete lattice and* Σ *is the greatest element of* $\mathcal{PF}$.

2.3. Prefix Free Generating Sets

A language over Σ is a subset of Σ^+. In this subsection, we consider a particular set of strings generating all strings in a given language.

Definition 2.3. Let $G \subseteq \Sigma^+$ and L be a language. G is a *generating set* of L if $L \sqsubseteq G$. A generating set G of L is *reduced* (with respect to L) if $L \not\sqsubseteq G'$ for any proper subset G' of G. By $\mathcal{PFG}_L$ we denote the set of all prefix free and reduced generating sets of L.

Note that if a generating set G of a language L is prefix free, by Lemma 2.3(1) each string of L has a unique factorization of strings in G. Thus for any prefix free generating set G of L, we have a unique *reduced* generating set $G_0 \subseteq G$ by deleting strings of G not used in factorizations of strings of L. Thus we get:

Lemma 2.11. *Let* L *be a language and* G *be a prefix free generating set of* L. *Then there uniquely exists a prefix free and reduced generating set* $G_0 \in \mathcal{PFG}_L$ *of* L *such that* $G_0 \subseteq G$.

We first show that $\mathrm{Pre}(O^*(L))$ introduced in the previous section is a prefix free and reduced generating set of a language L and moreover, the greatest element in the class $\mathcal{PFG}_L$ under the relation $\sqsubseteq$.

Lemma 2.12. *For any language* L, *the set* $\mathrm{Pre}(O^*(L))$ *is a prefix free and reduced generating set of* $O^*(L)$.

Lemma 2.13. *Let* $S \subseteq \Sigma^+$. *For any* $w \in O^*(S)$, *there is a string* $u \in S$ *such that* $u = vw$ *for some* $v \in (O^*(S))^*$.

Theorem 2.14. *For any language L, $\mathrm{Pre}(O^*(L)) \in \mathcal{PFG}_L$.*

Theorem 2.15. *For any language L, $\mathrm{Pre}(O^*(L))$ is the least element of $\mathcal{PFG}_L$ under the partially ordered relation $\sqsubseteq$.*

Proof. Put $T = \mathrm{Pre}(O^*(L))$. By Theorem 2.14, T is a prefix free and reduced generating set of L. Thus we show that T is the least element of $\mathcal{PFG}_L$.

Let G be any prefix free and reduced generating set of L. Then since $L \sqsubseteq G$ and G is prefix free, it implies from Lemma 2.6 that $O^*(L) \sqsubseteq G$. This means $T \sqsubseteq G$ because of $T \subseteq O^*(L)$. Hence T is the least element of $\mathcal{PFG}_L$. ∎

Clearly $G_L^{\sup}$ consists of all symbols of Σ appearing in some string of L. In what follows, we denote by $G_L^{\inf}$ and $G_L^{\sup}$ the least element and the greatest element in $\mathcal{PFG}_L$, respectively. That is,

$$G_L^{\inf} = \mathrm{Pre}(O^*(L)), \qquad G_L^{\sup} = \{a \in \Sigma \mid a \text{ appears some string in } L\},$$

and for any $G \in \mathcal{PFG}_L$, $G_L^{\inf} \sqsubseteq G \sqsubseteq G_L^{\sup}$.

As a direct result of the above theorem, it follows that:

Corollary 2.16. *Let L be a language. If $G_L^{\inf}$ is finite, then*

$$|G_L^{\inf}| < |G|, \quad \text{for any finite } G \in \mathcal{PFG}_L, \text{ where } G \neq G_L^{\inf}.$$

Corollary 2.17. *Let L_1 and L_2 be languages. If $L_1 \subseteq L_2$, then $G_{L_1}^{\inf} \sqsubseteq G_{L_2}^{\inf}$.*

Now we investigate a lattice structure of $\mathcal{PFG}_L$ for a given language L.

Lemma 2.18. *Let L be a language. For any subset $\mathcal{G} \subseteq \mathcal{PFG}_L$, $\sup(\mathcal{G}) \in \mathcal{PFG}_L$.*

Lemma 2.19. *Let L be a language. For any subset $\mathcal{G} \subseteq \mathcal{PFG}_L$, there is the greatest lower bound of $\mathcal{G}$ in $\mathcal{PFG}_L$.*

In general, for a subset $\mathcal{G} \subseteq \mathcal{PFG}_L$, $\inf(\mathcal{G})$ is not always the greatest lower bound of $\mathcal{G}$ in $\mathcal{PFG}_L$. In fact, let us consider a language $L = \{w\}^+$, where $w = abcdacdaab$. Let $G_1 = \{abcd, aab, acd\}$ and $G_2 = \{ab, cda\}$. As easily seen, $G_1, G_2 \in \mathcal{PFG}_L$, and $\inf\{G_1, G_2\} = \{abcdaab, w\}$. Clearly $\inf\{G_1, G_2\}$ is not reduced although $L \sqsubseteq \inf\{G_1, G_2\}$. The greatest lower bound of $\{G_1, G_2\}$ is given by $\{w\} \subseteq \inf\{G_1, G_2\}$. Note that the set $\{w\}$ is the greatest lower bound of $\mathcal{PFG}_L$, i.e., $G_L^{\inf} = \{w\}$.

By Theorem 2.14, Theorem 2.15, Lemma 2.18 and Lemma 2.19, we have the main result in this paper as follows:

Theorem 2.20. *For a language L, $(\mathcal{PFG}_L, \sqsubseteq)$ is a complete lattice under the partially ordered relation $\sqsubseteq$.*

Next, we consider a case that $G_L^{\inf}$ is finite. As shown in Corollary 2.16, $G_L^{\inf}$ has the fewest cardinality among $\mathcal{PFG}_L$.

For a finite language, the next result is given:

Lemma 2.21. *Let L be a finite language. Then there is an integer $n \in N$ such that $O^n(L) = O^*(L)$, and moreover the set $O^*(L)$ is finite.*

Lemma 2.22. *Let $S \subseteq \Sigma^+$ and $n \in N$. For any $w \in O^n(S)$, there is a finite subset S_w of $O^n(S)$ such that*

(1) $w \in S_w$, and (2) for any $u \in S_w$ with $u \neq w$, S_w contains some direct ancestor of u.

Theorem 2.23. *Let L be a language. $G_L^{\inf}$ is finite if and only if there is a finite subset S of L such that $L \sqsubseteq G_S^{\inf}$.*

For a string $w \in \Sigma^+$, head(w) represents the first letter of w. We consider a particular prefix free generating set introduced by Yokomori[5]:

A prefix free S is *simple* if head$(u) \neq$ head(v) for any $u, v \in S$ with $u \neq v$. For a language L, by $\mathcal{SPFG}_L$ be the set of all simple prefix free and reduced generating sets of L. Clearly the cardinality of any simple prefix free set is less than or equal to that of Σ.

Watanabe[6] has shown the next result on $\mathcal{SPFG}_L$:

Theorem 2.24 (Watanabe[6]). *For any language L, the set $\mathcal{SPFG}_L$ is a finite lattice.*

3. Polynomial Time Algorithms for Computing $G_L^{\inf}$

In this section, we present an efficient algorithm for computing a prefix free and reduced generating set $G_L^{\inf}$ of a given language L, provided that $G_L^{\inf}$ is finite. For an infinite language, we give an efficient learning algorithm for $G_L^{\inf}$ in the framework of *identification in the limit* due to Gold[4].

3.1. An Algorithm for a Finite Language

We first consider $G_L^{\inf}$ for a finite language L. As shown in the previous section, $G_L^{\inf} = \mathrm{Pre}(O^*(L))$. If L is finite, $O^n(L) = O^{n+1}(L)$ for some n, and $O^*(L) = O^n(L)$ as shown in Lemma 2.21. Thus it is easy to compute the set $G_L^{\inf}$, but the number n of operations O may be exponential even if L is finite. In order to avoid it, we introduce another operation instead of O as follows: For $x, y \in \Sigma^+$ and $S \subseteq \Sigma^+$,

$$\tilde{O}(x,y) = \begin{cases} \{y'\}, & \text{if } \exists y' \in \Sigma^+ \text{ s.t. } y = xy', \\ \phi, & \text{otherwise,} \end{cases}$$

$$\tilde{O}(S) = \bigcup_{\substack{x \in \mathrm{Pre}(S) \\ y \in S - \mathrm{Pre}(S)}} \tilde{O}(x,y) \cup \mathrm{Pre}(S).$$

Similarly to the definition of the operation O, we define $\tilde{O}^0(S) = S$ and $\tilde{O}^{n+1}(S) = \tilde{O}(\tilde{O}^n(S))$ $(n \in N)$.

Clearly if $\tilde{O}^n(S)$ is prefix free for some n, $\tilde{O}^n(S) = \tilde{O}^m(S)$ for any $m \geq n$, and we denote it by $\tilde{O}^*(S)$.

Lemma 3.1. *For any nonempty set $S \subseteq \Sigma^+$ and any $n \in N$,*
(1) $\tilde{O}^n(S) \sqsubseteq \tilde{O}^{n+1}(S)$, (2) $\tilde{O}^n(S) \subseteq O^n(S)$.

For a finite set S of strings, we denote by $||S||$ the sum of lengths of strings contained in S.

Lemma 3.2. *Let L be a finite language. Then*
(1) $|\tilde{O}^n(L)| \geq |\tilde{O}^{n+1}(L)|$, where the equality is valid if $\tilde{O}^n(L)$ is prefix free.
(2) $||\tilde{O}^n(L)|| \leq ||\tilde{O}^{n+1}(L)||$, and the equality is valid if and only if $\tilde{O}^n(L)$ is prefix free.

Theorem 3.3. *For any finite language L, $\tilde{O}^*(L) = G_L^{\inf}$.*

We first present a procedure for computing $\tilde{O}(S)$ for a given finite set S:

Algorithm $\tilde{O}(S)$
Input: a finite set S of strings;
Output: the set $\tilde{O}(S)$;

begin
$T := \phi$;
for each $(x, y) \in (\mathrm{Pre}(S) \times (S - \mathrm{Pre}(S)))$ **do** $T := T \cup \tilde{O}(x, y)$;
output $T \cup \mathrm{Pre}(S)$
end.

Let $n = |S|$ and $m = \max\{|x| \mid x \in S\}$. In the above procedure, $\mathrm{Pre}(S)$ can be computed in $O(n^2m)$ of time, and for each pair (x, y), $\tilde{O}(x, y)$ can be computed in $O(m)$. Thus the procedure for $\tilde{O}(S)$ correctly output $\tilde{O}(S)$ in $O(n^2m)$ of time.

Now we give a polynomial time algorithm for computing $G_L^{\inf}$ as follows:

Algorithm $G_L^{\inf}$
Input: a finite language L;
Output: the set $G_L^{\inf}$;

begin
$T := L$;
repeat
$T' := T$; $T := \tilde{O}(T)$
until $T = T'$;
output T
end.

Theorem 3.4. *Let L be any finite language. Then Algorithm $G_L^{\inf}$ correctly computes $G_L^{\inf}$ in $O(n^3m^2)$ of time, where $n = |L|$ and $m = \max\{|x| \mid x \in L\}$.*

3.2. Identification of $G_L^{\inf}$ in the Limit

In this subsection, we consider a problem of identifying $G_L^{\inf}$ in the frame work of inductive inference based on *identification in the limit* introduced by Gold[4] for language learning, provided $G_L^{\inf}$ is finite.

Inductive inference is a process to guess an unknown general rule from given examples. Gold[4] proposed a mathematical model of inductive inference based on a criteria called *identification in the limit* as follows: A positive presentation σ of a language L is an infinite sequence $w_1, w_2, \cdots$ of strings such that $\{w_n \mid n \geq 1\} = L$. An inference machine M is an effective procedure that requests a string and produces a conjecture at a time. Given a positive presentation $\sigma = w_1, w_2, \cdots$, M generates an infinite sequence $g_1, g_2, \cdots$ of conjectures. In language identification, conjectures mean some devices defining languages such as automata, formal grammars and so on. Refer in detail to Angluin[1]. In this paper, conjectures generated by the inference machine are finite sets of strings. We say that M identifies $G_L^{\inf}$ in the limit from positive data of a target language L, if there is an integer n such that $g_m = G_L^{\inf}$ for any $m \geq n$.

Let $T_1, T_2, \cdots$ be an infinite sequence of sets of strings. The sequence $T_1, T_2, \cdots$ converges to a set $T \subseteq \Sigma^+$, denoted by $\lim_{n\to\infty} T_n = T$, if there exists an integer n_0 such that $T_n = T$ for any $n \geq n_0$.

Let $\sigma = w_1, w_2, \cdots$ be a positive presentation of L, and let $S_n = \{w_1, w_2, \cdots, w_n\}$ for each $n \in N$.

Lemma 3.5. *Let L be a language. If $G_L^{\inf}$ is finite, then*

$$\lim_{n\to\infty} G_{S_n}^{\inf} = G_L^{\inf}$$

Lemma 3.6. *For a finite set $S \subseteq \Sigma^+$ and a string $w \in \Sigma^+$,*

$$G^{\inf}_{S\cup\{w\}} = G^{\inf}_{G^{\inf}_S \cup\{w\}}.$$

Now we present an inference algorithm as follows:

Algorithm LA
Input: a positive presentation of a language L;
Output: a sequence of prefix free and reduced generating sets;

begin
$T_0 := \phi$; $n := 1$;
repeat
read the next data w_n;
$T_n := G^{\inf}_{T_{n-1}\cup\{w_n\}}$;
output T_n as the n-th conjecture;
$n := n + 1$
forever
end.

For each n, let $S_n = \{w_1, \cdots, w_n\}$ be a sample set of a target language L, and T_n be the n-th conjecture of the above algorithm.

Theorem 3.7. *Let L be a language. If $G^{\inf}_L$ is finite, the algorithm LA identifies $G^{\inf}_L$ in the limit, and may be implemented to update the conjecture in time $O(n^3m^2)$, where $m = \max\{|w_i| \mid i = 1, 2, \cdots, n\}$.*

Proof. By Lemma 3.6, it is easy to show that $T_n = G^{\inf}_{S_n}$ for any n. Appealing to Lemma 3.5, we obtain $\lim_{n\to\infty} T_n = G^{\inf}_L$ because $G^{\inf}_L$ is finite. Thus the algorithm identifies $G^{\inf}_L$ in the limit.

Using Theorem 3.3, $|G^{\inf}_{S_n}| \leq n$ and the length of the longest strings in $G^{\inf}_{S_n}$ is less than or equal to that in S_n. Thus by Theorem 3.4, the n-th conjecture $T_n = G^{\inf}_{T_{n-1}\cup\{w_n\}}$ may be implemented to update the conjecture in time $O(n^3m^2)$, where $m = \max\{|w_i| \mid i = 1, 2, \cdots, n\}$. ∎

References

[1] D. Angluin, *Inductive Inference of Formal Languages from Positive Data*, Information and Control, **45**, 117–135, (1980).

[2] R. Ash, "Information Theory," Interscience Publishers, 1965.

[3] R.M. Capocelli, *A Decision Procedure for Finite Decipherability and Synchronizability of Multivalued Encodings*, IEEE Transactions on Information Theory, **IT-28**, No. 2, 307–318, (1982).

[4] E.M. Gold, *Language Identification in the Limit*, Information and Control, **10**, 447–474, (1967).

[5] T. Yokomori, *On Polynomial-Time Learnability in the Limit of Strictly Deterministic Automata*, Machine Learning, **19**, 153–179, (1995).

[6] N. Watanabe, *Polynomial-Time Inductive Inference of Simple Regular Automata*, Master thesis, Osaka Prefecture University, 1996.