Polynomial Time Identification of Strict Prefix Deterministic Finite State Transducers*

Mitsuo Wakatsuki and Etsuji Tomita

Graduate School of Informatics and Engineering,
The University of Electro-Communications
Chofugaoka 1–5–1, Chofu, Tokyo 182-8585, Japan
{wakatuki,tomita}@ice.uec.ac.jp

Abstract. This paper is concerned with a subclass of finite state transducers, called *strict prefix deterministic finite state transducers* (*SPDFST*'s for short), and studies a problem of identifying the subclass in the limit from positive data. After providing some properties of languages accepted by SPDFST's, we show that the class of SPDFST's is polynomial time identifiable in the limit from positive data in the sense of Yokomori.

1 Introduction

A reasonable definition for *polynomial time identifiability in the limit* [3] from positive data has been proposed by Yokomori [4]. He has also proved that a class of languages accepted by strictly deterministic automata (SDA's for short) [4] and a class of very simple languages [5] are polynomial time identifiable in the limit from positive data. As for a class of transducers, Oncina et al. [2] have proved that a class of onward subsequential transducers (OST's for short), which is a proper subclass of finite state transducers, is polynomial time identifiable in the limit from positive data.

The present paper deals with a subclass of finite state transducers called *strict prefix deterministic finite state transducers* (*SPDFST*'s for short), and discusses the identification problem of the class of SPDFST's. The class of SDA's forms a proper subclass of associated automata with SPDFST's. Moreover, the class of languages accepted by SPDFST's is incomparable to the class of languages accepted by OST's. After providing some properties of languages accepted by SPDFST's, we show that the class of SPDFST's is polynomial time identifiable in the limit from positive data in the sense of Yokomori [4]. The main result in this paper provides another interesting instance of a class of transducers which is polynomial time identifiable in the limit. This identifiability is proved by giving an exact *characteristic sample* of polynomial size for a language accepted by an SPDFST.

2 Basic Definitions and Notation

An *alphabet* Σ is a finite set of symbols. We denote by Σ^* the set of all finite-length strings over Σ. The string of length 0 (the empty string) is denoted by

* This work was supported in part by Grants-in-Aid for Scientific Research Nos. 18500108 and 20500007 from the MEXT of Japan.

J.M. Sempere and P. García (Eds.): ICGI 2010, LNAI 6339, pp. 313–316, 2010.

ε. Let $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. We denote by $|w|$ the length of a string w and by $|S|$ the cardinality of a set S. A *language* over Σ is any subset L of Σ^*. For a string $w \in \Sigma^+$, $\mathrm{first}(w)$ denotes the first symbol of w. For $w \in \Sigma^*$, $\mathrm{alph}(w)$ denotes the set of symbols appearing in w. For $w \in \Sigma^*$ and its prefix $x \in \Sigma^*$, $x^{-1}w$ denotes the string $y \in \Sigma^*$ such that $w = xy$. For $S \subseteq \Sigma^*$, $\mathrm{lcp}(S)$ denotes the *longest common prefix* of all strings in S.

Let Σ be any alphabet and suppose that Σ is totally ordered by some binary relation $\prec$. Let $x = a_1 \cdots a_r$, $y = b_1 \cdots b_s$, where $r, s \geq 0$, $a_i \in \Sigma$ for $1 \leq i \leq r$, and $b_i \in \Sigma$ for $1 \leq i \leq s$. We write that $x \prec y$ if (i) $|x| < |y|$, or (ii) $|x| = |y|$ and there exists $k \geq 1$ so that $a_i = b_i$ for $1 \leq i < k$ and $a_k \prec b_k$. The relation $x \preceq y$ means that $x \prec y$ or $x = y$.

3 Strict Prefix Deterministic Finite State Transducers

A *finite state* or *rational transducer* (*FST* for short) is defined as a 6-tuple $T = (Q, \Sigma, \Delta, \delta, q_0, F)$, where Q is a finite set of *states*, Σ is an *input alphabet*, Δ is an *output alphabet*, δ is a finite subset of $Q \times \Sigma^* \times \Delta^* \times Q$ whose elements are called *transitions* or *edges*, q_0 is the *initial state*, and $F (\subseteq Q)$ is a set of *final states* [1][2]. A finite automaton $M = (Q, \Sigma, \delta', q_0, F)$, where $\delta' \subseteq Q \times \Sigma^* \times Q$ and $(p, x, y, q) \in \delta$ implies that $(p, x, q) \in \delta'$, is called an *associated automaton* with an FST T. A *path* in an FST T is a sequence of transitions $\pi = (p_0, x_1, y_1, p_1)(p_1, x_2, y_2, p_2) \cdots (p_{n-1}, x_n, y_n, p_n)$, where $p_i \in Q$ for $0 \leq i \leq n$, and $x_i \in \Sigma^*, y_i \in \Delta^*$ for $1 \leq i \leq n$. When the intermediate states involved in a path are insignificant, a path is written as $\pi = (p_0, x_1 x_2 \cdots x_n, y_1 y_2 \cdots y_n, p_n)$. For $p, q \in Q$, $\Pi_T(p, q)$ denotes the set of all paths from p to q. By convention, we let $(p, \varepsilon, \varepsilon, p) \in \Pi_T(p, p)$ for any $p \in Q$. We extend this notation by setting $\Pi_T(p, Q') = \cup_{q \in Q'} \Pi_T(p, q)$ for any $Q' \subseteq Q$. A path π from p to q is *successful* iff $p = q_0$ and $q \in F$. Thus, the set of all successful paths is $\Pi_T(q_0, F)$. Here, for a state $p \in Q$, it is said to be *reachable* if $\Pi_T(q_0, p) \neq \emptyset$, and it is said to be *live* if $\Pi_T(p, F) \neq \emptyset$. For an FST T, the *language* accepted by T is defined to be $L(T) = \{(x, y) \in \Sigma^* \times \Delta^* \mid (q_0, x, y, q) \in \Pi_T(q_0, F)\}$.

Definition 1. *Let $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ be an FST. Then, T is a* strict prefix deterministic finite state transducer *(*SPDFST*) iff T satisfies the following conditions: (1) $\delta \subseteq Q \times \Sigma^+ \times \Delta^+ \times Q$. (2) For any $(p, x_1, y_1, q_1), (p, x_2, y_2, q_2) \in \delta$, if* $\mathrm{first}(x_1) = \mathrm{first}(x_2)$*, then $x_1 = x_2$, $y_1 = y_2$ and $q_1 = q_2$ (*determinism condition*). (3) For any $(p, x_1, y_1, q_1), (p, x_2, y_2, q_2) \in \delta$, if* $\mathrm{first}(x_1) \neq \mathrm{first}(x_2)$*, then* $\mathrm{first}(y_1) \neq \mathrm{first}(y_2)$*. (4) For any $(p_1, x_1, y_1, q_1), (p_2, x_2, y_2, q_2) \in \delta$ with $p_1 \neq p_2$ or $q_1 \neq q_2$, it holds that* $\mathrm{first}(x_1) \neq \mathrm{first}(x_2)$ *or* $\mathrm{first}(y_1) \neq \mathrm{first}(y_2)$ *(i.e., the uniqueness of labels). If T satisfies the conditions (3) and (4), we say that T has the* strict prefix property*.*

An SPDFST $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ is said to be in *canonical form* if, for any $p \in Q$, p is reachable and live, and for any $p \in Q - \{q_0\}$, it holds that $p \in F$ or $|\{(p, x, y, q) \in \delta \mid x \in \Sigma^+, y \in \Delta^+, q \in Q\}| \geq 2$. For any SPDFST T', there exists an SPDFST T in canonical form such that $L(T) = L(T')$, and we can construct an algorithm that outputs such T. Hereafter, we are concerned with SPDFST's in canonical form.

The following lemmas are derived from Definition 1.

Lemma 1. *Let $T = (Q,\ \Sigma,\ \Delta,\ \delta,\ q_0,\ F)$ be an SPDFST, and let $p, p', q, q' \in Q$, $x, x' \in \Sigma^+$, and $y, y' \in \Delta^+$. Then, the followings hold. (1) If $(p, x, y, q) \in \Pi_T(p, q)$ and $(p, x, y', q') \in \Pi_T(p, q')$, then $y = y'$ and $q = q'$. (2) If $(p, x, y, q) \in \Pi_T(p, q)$ and $(p', x, y, q') \in \Pi_T(p', q')$, then $p = p'$ and $q = q'$. (3) For some $\pi = (p, x, y, q) \in \Pi_T(p, q)$ and $\pi' = (p, x', y', q') \in \Pi_T(p, q')$, if $\mathrm{first}(x) = \mathrm{first}(x')$ and $\mathrm{first}(y) = \mathrm{first}(y')$, then π can be divided into (p, x_c, y_c, r) and $(r, {x_c}^{-1}x, {y_c}^{-1}y, q)$, and π' can be divided into (p, x_c, y_c, r) and $(r, {x_c}^{-1}x', {y_c}^{-1}y', q')$, where $x_c = \mathrm{lcp}(\{x, x'\})$, $y_c = \mathrm{lcp}(\{y, y'\})$, and $r \in Q$.*

Lemma 2. *Let $T = (Q,\ \Sigma,\ \Delta,\ \delta,\ q_0,\ F)$ be an SPDFST and let $(x, y), (x_1, y_1), (x_2, y_2) \in L(T)$. Then, for each $a, a_1, a_2 \in \Sigma$ $(a_1 \neq a_2)$, $b, b_1, b_2 \in \Delta$ $(b_1 \neq b_2)$, the followings hold. (1) If $x = ax''$ and $y = by''$ for some $x'' \in \Sigma^*, y'' \in \Delta^*$, then there exists a transition $(q_0, u, v, p) \in \delta$ such that $\mathrm{first}(u) = a$ and $\mathrm{first}(v) = b$ for some $p \in Q$. (2) If $x_1 = x'a_1x_1'', x_2 = x'a_2x_2''$, $y_1 = y'b_1y_1''$ and $y_2 = y'b_2y_2''$ for some $x', x_1'', x_2'' \in \Sigma^*$, $y', y_1'', y_2'' \in \Delta^*$, then there exist $p, q_1, q_2 \in Q$, $u_1, u_2 \in \Sigma^+$, and $v_1, v_2 \in \Delta^+$ such that $(p, u_1, v_1, q_1), (p, u_2, v_2, q_2) \in \delta$ with $\mathrm{first}(u_1) = a_1$, $\mathrm{first}(u_2) = a_2$, $\mathrm{first}(v_1) = b_1$ and $\mathrm{first}(v_2) = b_2$. (3) If $x_2 = x_1ax_2''$ and $y_2 = y_1by_2''$ for some $x_2'' \in \Sigma^*$, $y_2'' \in \Delta^*$, then there exist $p \in F$, $q \in Q$, $u \in \Sigma^+$, and $v \in \Delta^+$ such that $(p, u, v, q) \in \delta$ with $\mathrm{first}(u) = a$ and $\mathrm{first}(v) = b$.*

From the definition of SDA's [4, p.159, Definition 5], we can show that the class of SDA's is a proper subclass of associated automata with SPDFST's. Moreover, from the definition of OST's [2, p.450], we can show that the class of languages accepted by OST's is incomparable to the class of languages accepted by SPDFST's.

4 Identifying SPDFST's

Let $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ be any SPDFST in canonical form. A finite subset $R \subseteq \Sigma^* \times \Delta^*$ of $L(T)$ is called a *characteristic sample* of $L(T)$ if $L(T)$ is the smallest language accepted by an SPDFST containing R, i.e., if for any SPDFST T', $R \subseteq L(T')$ implies that $L(T) \subseteq L(T')$.

For each $p \in Q$, define $\mathrm{pre}(p)$ as the *shortest* input string $x \in \Sigma^*$ from q_0 to p, i.e., $(q_0, x, y, p) \in \Pi_T(q_0, p)$ and $x \preceq x'$ for any x' such that $(q_0, x', y', p) \in \Pi_T(q_0, p)$. Moreover, for each $p \in Q$ and $q \in F$, define $\mathrm{post}(p, q)$ $(\in \Sigma^*)$ as the *shortest* input string from p to q. Then, define $R_I(T) = \{\mathrm{pre}(p) \cdot \mathrm{post}(p, q) \mid p \in Q,\ q \in F\} \cup \{\mathrm{pre}(p) \cdot x \cdot \mathrm{post}(r, q) \mid p \in Q,\ (p, x, y, r) \in \delta, q \in F\} \cup \{\mathrm{pre}(p) \cdot x_1 \cdot x_2 \cdot \mathrm{post}(s, q) \mid p \in Q,\ (p, x_1, y_1, r), (r, x_2, y_2, s) \in \delta,\ q \in F\}$ and $R(T) = \{(x, y) \in \Sigma^* \times \Delta^* \mid x \in R_I(T),\ (q_0, x, y, q) \in \Pi_T(q_0, F)\}$. $R(T)$ is called a *representative sample* of T. Note that the cardinality $|R(T)|$ of a representative sample is at most $|Q|^2\,(|\Sigma|^2 + |\Sigma| + 1)$, that is, $|R(T)|$ is polynomial with respect to the description length of T. We can prove that $R(T)$ is a characteristic sample of $L(T)$.

Let T_* be a target SPDFST. The idenitification algorithm *IA* is given in the following.

```
**Input:** a positive presentation $(x_1, y_1), (x_2, y_2), \dots$ of $L(T_*)$ for $T_*$
**Output:** a sequence of SPDFST's $T_1, T_2, \dots$
**Procedure** *IA*
**begin**
  initialize $i = 0$;   $q_0 := p_{[\varepsilon]}$;   $h(p_{[\varepsilon]}) := \varepsilon$;
  let $T_0 = (\{p_{[\varepsilon]}\}, \emptyset, \emptyset, \emptyset, q_0, \emptyset)$ be the initial SPDFST;
  **repeat** (forever)
    let $T_i = (Q_i, \Sigma_i, \Delta_i, \delta_i, q_0, F_i)$ be the current conjecture;
    $i := i + 1$;   read the next positive example $(x_i, y_i)$;
    **if** $(x_i, y_i) \in L(T_{i-1})$ **then output** $T_i = T_{i-1}$ as the $i$-th conjecture
    **else**
      $Q_i := Q_{i-1}$;   $\Sigma_i := \Sigma_{i-1}$;   $\Delta_i := \Delta_{i-1}$;   $\delta_i := \delta_{i-1}$;   $F_i := F_{i-1}$;
      **if** $x_i = \varepsilon$ and $y_i = \varepsilon$ **then**
        $F_i := F_i \cup \{p_{[\varepsilon]}\}$;
        **output** $T_i = (Q_i, \Sigma_i, \Delta_i, \delta_i, q_0, F_i)$ as the $i$-th conjecture
      **else** /* the case where $x_i \neq \varepsilon$ and $y_i \neq \varepsilon$ */
        $Q_i := Q_i \cup \{p_{[x_i]}\}$;   $\Sigma_i := \Sigma_i \cup \mathrm{alph}(x_i)$;   $\Delta_i := \Delta_i \cup \mathrm{alph}(y_i)$;
        $F_i := F_i \cup \{p_{[x_i]}\}$;   $h(p_{[x_i]}) := x_i$;
        $T_i :=$ CONSTRUCT$(Q_i, \Sigma_i, \Delta_i, \delta_i \cup \{(p_{[\varepsilon]}, x_i, y_i, p_{[x_i]})\}, q_0, F_i)$;
        **output** $T_i$ as the $i$-th conjecture   **fi**   **fi**
  **until** ($false$)
**end**
```

Here, the function CONSTRUCT$(Q, \Sigma, \Delta, \delta, q_0, F)$ merges states in Q so that it satisfies Lemma 1 (2) and divides a transition in δ into two transitions so that it satisfies Lemma 1 (3) repeatedly, and outputs the updated SPDFST.

By using Lemmas 1 and 2 and analyzing the behavior of the identification algorithm *IA* in the similar way as in [4], we have the following conclusion.

Theorem 1. *The class of SPDFST's is polynomial time identifiable in the limit from positive data in the sense of Yokomori [4].*

References

1. Berstel, J.: Transductions and Context-Free Languages. Teubner Studienbücher, Stuttgart (1979)
2. Oncina, J., García, P., Vidal, E.: Learning subsequential transducers for pattern recognition interpretation tasks. IEEE Trans. on Pattern Analysis and Machine Intelligence 15(5), 448–458 (1993)
3. Pitt, L.: Inductive inference, DFAs, and computational complexity. In: Jantke, K.P. (ed.) AII 1989. LNCS (LNAI), vol. 397, pp. 18–44. Springer, Heidelberg (1989)
4. Yokomori, T.: On polynomial-time learnability in the limit of strictly deterministic automata. Machine Learning 19, 153–179 (1995)
5. Yokomori, T.: Polynomial-time identification of very simple grammars from positive data. Theoretical Computer Science 298, 179–206 (2003)