Theoretical Computer Science 32 (1984) 321–330
North-Holland

ALTERNATING FINITE AUTOMATA ON ω-WORDS

Satoru MIYANO
Department of Mathematics, Kyushu University 33, Fukuoka 812, Japan

Takeshi HAYASHI
College of General Education, Kyushu University 01, Ropponmatsu, Fukuoka 810, Japan

Communicated by M. Nivat
Received July 1983

Abstract. Alternating finite automata on ω-words are introduced as an extension of nondeterministic finite automata which process infinite sequences of symbols. The classes of ω-languages defined by alternating finite automata are investigated and characterized under four types of acceptance conditions. It is shown that for one type of acceptance condition alternation increases the power in comparison with nondeterminism and for other three acceptance conditions nondeterministic finite automata on ω-words have the same power as alternating ones.

1. Introduction

An ω-word is an infinite sequence of symbols of finite kinds and various types of automata which process ω-words have been investigated [1, 3–7]. In particular, nondeterministic and deterministic finite automata on ω-words have been extensively studied through a variety of acceptances and the corresponding ω-language classes have been precisely determined [1, 8–11, 13–19]. It is known, in contrast with the case of finite length words, that for some acceptance condition nondeterministic finite automata define a larger class of ω-languages than deterministic ones. The reader is suggested to look up the table in [19] for details. As an extended notion of nondeterminism, alternation has been introduced and many kinds of alternating automata have been studied in relation to the complexity theory [2, 12]. It has been shown that alternating finite automata on finite length words accept only regular languages, whether they have one-way input heads or two-way input heads [2, 12].

Alternation can be also introduced to finite automata on ω-words. When finite automata are viewed as a method of describing the sets of ω-words, alternation has a benefit that it makes the descriptions easy and simple. The purpose of this paper is to investigate how alternation affects the recognition power of ω-languages by finite automata in comparison with nondeterministic ones. We consider four basic ω-word acceptances including ones introduced in [1, 13].

2. Preliminaries

Let Σ be a finite alphabet. We denote by Σ^ω the set of all infinite sequences over Σ. Elements in Σ^ω are called *ω-words* and a subset L of Σ^ω is called an *ω-language.*

Definition 2.1. An *alternating finite automaton* (abbreviated *afa*) is a sextuple $M = (Q, f, \Sigma, \Delta, q_0, \mathcal{F})$, where:

(1) Q is a finite set of states and f is a mapping $f: Q \to \{and, or\}$. If $f(q) = and$ (resp. or), q is called a *universal state* (resp. *existential state*).

(2) Σ is a finite alphabet.

(3) Δ is a subset of $Q \times \Sigma \times Q$ such that the set $\Delta(q, a) = \{p \mid (q, a, p) \text{ is in } \Delta\}$ is not empty for any q in Q and any a in Σ. Elements in Δ are called *transitions.*

(4) q_0 is a state in Q called the *initial state.*

(5) $\mathcal{F}$ is a family of subsets of Q. For F in $\mathcal{F}$, F is called a *final set* and elements in F are called *final states.* If $\mathcal{F}$ consists of a single final set, then we say that M is *simple.*

Definition 2.2. A *nondeterministic finite automaton* (abbreviated *nfa*) is an alternating finite automaton whose states are all existential states and we omit the function f from notation. A *deterministic finite automaton* (abbreviated *dfa*) is an *nfa* with the property that $|\Delta(q, a)| = 1$ for each q in Q and each a in Σ.

Definition 2.3. Let $x = x_1x_2x_3\ldots$ be in Σ^ω, where x_n is in Σ for $n \geq 1$. A *computation tree* $T(M, x)$ *of* M *on* x is an infinite labelled tree satisfying the following conditions:

(1) $T(M, x)$ has no leaf.

(2) The nodes are labelled with the elements in Q. In particular, the root of $T(M, x)$ is labelled with q_0.

(3) For each $n \geq 1$, the edges between level n and level $n+1$ are labelled with x_n.

(4) If node v in level n is labelled with a universal state q, then v has a child labelled with p for each p in $\Delta(q, x_n)$.

(5) If node v in level n is labelled with an existential state q, then v has exactly one child labelled with p for some p in $\Delta(q, x_n)$.

Remark. For nondeterministic finite automata, a computation tree is an infinite path.

Definition 2.4. An infinite path α in $T(M, x)$ beginning at the root is called a *run in* $T(M, x)$. For a run α, we define

(1) $I(\alpha) = \{q \mid \text{state } q \text{ occurs in } \alpha \text{ infinitely many times}\}$,

(2) $O(\alpha) = \{q \mid \text{state } q \text{ occurs in } \alpha\}$.

Definition 2.5. For an alternating finite automaton $M = (Q, f, \Sigma, \Delta, q_0, \mathcal{F})$ and x in Σ^ω, we say that M accepts x in the sense of C_i $(i = 1, \ldots, 4)$ if there exists a computation tree $T(M, x)$ such that for each run α in $T(M, x)$, there exists F in $\mathcal{F}$

satisfying the condition C_i, where

(C_1) $I(\alpha)\cap F\neq\emptyset$,

(C_2) $I(\alpha)\subseteq F$,

(C_3) $O(\alpha)\cap F\neq\emptyset$,

(C_4) $O(\alpha)\subseteq F$.

We call $T(M,x)$ an *accepting computation tree* of M on x in the sense of C_i, respectively. For $i=1,\ldots,4$, we denote by $L_i(M)$ the set of ω-words accepted by M in the sense of C_i and we say that M *recognizes* an ω-language L in the sense of C_i if $L=L_i(M)$.

Definition 2.6. For $i=1,\ldots,4$, we define
(1) $\mathscr{A}_i=\{L_i(M)\mid M$ is an *afa*$\}$,
(2) $\mathscr{A}_i^s=\{L_i(M)\mid M$ is a simple *afa*$\}$,
(3) $\mathscr{N}_i=\{L_i(M)\mid M$ is an *nfa*$\}$,
(4) $\mathscr{N}_i^s=\{L_i(M)\mid M$ is a simple *nfa*$\}$,
(5) $\mathscr{D}_i=\{L_i(M)\mid M$ is a *dfa*$\}$,
(6) $\mathscr{D}_i^s=\{L_i(M)\mid M$ is a simple *dfa*$\}$.

3. Overview of the classes concerned

The classes $\mathscr{D}_i$ and $\mathscr{N}_i$ $(i=1,\ldots,4)$ have been characterized in terms of general topology and the representations of the ω-languages in these classes have been obtained by applying several operations to regular languages. In the following sections we will determine the classes for alternating finite automata on ω-words by establishing the relationship between $\mathscr{A}_i$ and $\mathscr{N}_i$ for $i=1,\ldots,4$. We will be concerned with the classes denoted by R, G^R, F^R, G_δ^R and F_σ^R, which are defined as follows:

(1) R: An ω-language in R is of the form $\bigcup_{i=1}^n U_i\cdot V_i^\omega$ for some regular languages U_i and V_i, where for $U\subseteq\Sigma^*$ we define $U^\omega=\{y_1y_2\ldots y_n\ldots\mid y_j$ is a nonempty word in U for $j\geq 1\}$. R is called the class of ω-regular languages [14].

(2) G^R: An ω-language in G^R is of the form $U\cdot\Sigma^\omega$, where U is a regular language and Σ a finite alphabet.

(3) F^R: For a set $U\subseteq\Sigma^*$, $A(U)$ denotes the set $\{w\mid w\in\Sigma^*$ and w is an initial segment of some word in $U\}$. An ω-language in F^R is described as $C(U)=\{x\mid x\in\Sigma^\omega$ and each initial segment of x is in $A(U)\}$ for some regular language U.

(4) F_σ^R: An ω-language in F_σ^R is of the form $\bigcup_{i=1}^n U_i\cdot C(V_i)$ for some regular languages U_i and V_i.

(5) G_δ^R: An ω-language L in G_δ^R is written as $L=\{x\mid x\in\Sigma^\omega, A(x)\cap W$ is infinite$\}$ for some regular language $W\subseteq\Sigma^*$, where $A(x)\subseteq\Sigma^*$ denotes the set of the initial segments of x.

We summarize in Table 1 our new results together with the former results on deterministic and nondeterministic finite automata on ω-words. For $\mathscr{D}_i$ and $\mathscr{N}_i$, we refer the reader to the table in [19, p. 129].

As shown in Table 1, C_1 type acceptance separates the nondeterministic and deterministic classes and C_2 type acceptance separates the alternating and nondeterministic classes. On the other hand, no changes occur under C_3 and C_4 type acceptances.

Table 1

	C_1	C_2	C_3	C_4
$\mathscr{D}_i$	G_δ^R	F_σ^R	G^R	F^R
$\mathscr{N}_i$	R	F_σ^R	G^R	F^R
$\mathscr{A}_i$	R	R	G^R	F^R

4. Simple alternating finite automata

It is known that a single final set is sufficient for nondeterministic and deterministic cases under all acceptance conditions C_i, i.e., $\mathscr{D}_i = \mathscr{D}_i^s$ and $\mathscr{N}_i = \mathscr{N}_i^s$ for $i = 1, \ldots, 4$ [17]. The purpose of this section is to show that the same fact holds for alternating finite automata on ω-words.

Theorem 4.1. *$\mathscr{A}_i = \mathscr{A}_i^s$ for $i = 1, \ldots, 4$.*

Proof. Let $M = (Q, f, \Sigma, \Delta, q_0, \mathscr{F})$. For each $i = 1, \ldots, 4$ we construct a simple *afa* which recognizes the same ω-language as M in the sense of C_i.

Case (i). C_1: Consider a simple *afa* $M^s = (Q, f, \Sigma, \Delta, q_0, \{F^s\})$, where $F^s = \{q \mid q \in F, F \in \mathscr{F}\}$. Let x be in Σ^ω and let $T(M, x)$ be a computation tree of M on x. $T(M, x)$ is also a computation tree of M^s on x. Then observe that the following two statements are equivalent:

(1) For each run α in $T(M, x)$ there exists an F in $\mathscr{F}$ such that $I(\alpha) \cap F \neq \emptyset$.

(2) For each run α in $T(M, x)$, $I(\alpha) \cap F^s \neq \emptyset$.

Hence M and M^s recognize the same ω-language in the sense of C_1.

Case (ii). C_2: We define a simple *afa* $M^s = (Q^s, f^s, \Sigma, \Delta^s, q_0^s, \{F^s\})$ in the same way as in [17, Lemma 7]:

(1) $Q^s = \{(q, \mathscr{F}') \mid q \in Q, \mathscr{F}' \subseteq \mathscr{F}, q$ is in F for each F in $\mathscr{F}'\}$.

(2) $f^s(q, \mathscr{F}') = f(q)$ for $(q, \mathscr{F}')$ in Q^s.

(3) $q_0^s = (q_0, \emptyset)$.

(4) For each transition (q, a, p) in Δ, Δ^s contains $((q, \mathscr{F}'), a, (p, \{F \mid F \in \mathscr{F}', p \in F\}))$ for all $\emptyset \neq \mathscr{F}' \subseteq \mathscr{F}$ and $((q, \emptyset), a, (p, \{F \mid F \in \mathscr{F}$, and $p \in F\}))$.

(5) $F^s = \{(q, \mathscr{F}') \mid (q, \mathscr{F}') \in Q^s$ and $\mathscr{F}' \neq \emptyset\}$.

Let x be in Σ^ω. Then it should be noticed that there is a one-to-one correspondence between the computation trees of M on x and the computation trees of M^s on x. Namely, for a computation tree $T(M, x)$ of M on x, we associate it with a computation tree $T(M^s, x)$ of M^s on x that begins with the root $(q_0, \emptyset)$ and is created by expanding it according to the transitions corresponding those in $T(M, x)$. Conversely, for $T(M^s, x)$ we associate it with $T(M, x)$ by replacing label $(q, \mathcal{F}')$ by q for each node. Let $T(M, x)$ and $T(M^s, x)$ be the computation trees of M on x and M^s on x corresponding to each other. For a run α in $T(M, x)$ we denote by α^s the corresponding run in $T(M^s, x)$ and vice versa. We will show that the following two statements are equivalent:

(6) For each run α in $T(M, x)$, there exists an F in $\mathcal{F}$ such that $I(\alpha) \subseteq F$.

(7) For each run α^s in $T(M^s, x)$, $I(\alpha^s) \subseteq F^s$.

Let α be

$$q_0 \xrightarrow{x_1} q_1 \xrightarrow{x_2} \cdots \xrightarrow{x_n} q_n \xrightarrow{x_{n+1}} \cdots$$

and α^s be

$$(q_0, \mathcal{F}_0) \xrightarrow{x_1} (q_1, \mathcal{F}_1) \xrightarrow{x_2} \cdots \xrightarrow{x_n} (q_n, \mathcal{F}_n) \xrightarrow{x_{n+1}} \cdots.$$

Then note that there exists an integer k_0 such that, for any $k \geq k_0$, $I(\alpha) = \{q_n \mid n \geq k\}$. Similarly, there exists an integer $k_1 \geq k_0$ such that $I(\alpha^s) = \{(q_n, \mathcal{F}_n) \mid n \geq k\}$ for all $k \geq k_1$. Assume that (6) holds. If $\mathcal{F}_m = \emptyset$ for some $m \geq k_1$, then $\mathcal{F}_n = \emptyset$ for infinitely many $n \geq k_1$. Since $\mathcal{F}_m = \emptyset$ and $I(\alpha) = \{q_n \mid n \geq m\} \subseteq F$, we can see by the definition of Δ^s that $\mathcal{F}_{m+1}$ contains F. Therefore, again by the definition of Δ^s, $\mathcal{F}_n$ contains F for all $n > m$. Hence $\mathcal{F}_n \neq \emptyset$ for all $n > m$. This is a contradiction. Therefore $\mathcal{F}_n \neq \emptyset$ for all $n \geq k_1$. Hence $I(\alpha^s) \subseteq F^s$. Conversely assume that $I(\alpha^s) \subseteq F^s$. Then by the definition of Δ^s, $\mathcal{F}_{n+1} \subseteq \mathcal{F}_n$ and $\mathcal{F}_n \neq \emptyset$ for all $n \geq k_1$. Since $\mathcal{F}_n$ is finite, there exists an F such that F is in $\mathcal{F}_n$ for all $n \geq k_1$. Hence q_n is in F for all $n \geq k_1$. Therefore $I(\alpha) \subseteq F$. Hence M and M^s define the same set of ω-words.

Case (iii). C_3: The same as case (i).

Case (iv). C_4: We define a simple *afa* $M^s = (Q^s, f^s, \Sigma, \Delta^s, q_0^s, \{F^s\})$ as follows:

(1) $Q^s = \{(q, \mathcal{F}') \mid q \in Q, \mathcal{F}' \subseteq \mathcal{F}, q$ is in F for all F in $\mathcal{F}'\}$.

(2) $f^s(q, \mathcal{F}') = f(q)$ for $(q, \mathcal{F}')$ in Q^s.

(3) $q_0^s = (q_0, \mathcal{F}_0)$, where $\mathcal{F}_0 = \{F \mid q_0 \in F, F \in \mathcal{F}\}$.

(4) For each transition (q, a, p) in Δ, Δ^s contains $((q, \mathcal{F}'), a, (p, \{F \mid F \in \mathcal{F}', p \in F\}))$ for all $\mathcal{F}' \subseteq \mathcal{F}$.

(5) $F^s = \{(q, \mathcal{F}') \mid (q, \mathcal{F}') \in Q^s$ with $\mathcal{F}' \neq \emptyset\}$.

In the same way as Case (ii) there is a one-to-one correspondence between the computation trees of M on x and the computation trees of M^s on x. We use the same notations as in Case (ii). We will show the equivalence of the following two statements:

(6) For each run α in $T(M, x)$ there exists an F in $\mathcal{F}$ such that $O(\alpha) \subseteq F$.

(7) For each run α^s in $T(M^s, x)$, $O(\alpha^s) \subseteq F^s$.

Let α be

$$q_0 \xrightarrow{x_1} q_1 \xrightarrow{x_2} \cdots \xrightarrow{x_n} q_n \xrightarrow{x_{n+1}} \cdots$$

and α^s be

$$(q_0, \mathscr{F}_0) \xrightarrow{x_1} (q_1, \mathscr{F}_1) \xrightarrow{x_2} \cdots \xrightarrow{x_n} (q_n, \mathscr{F}_n) \xrightarrow{x_{n+1}} \cdots .$$

Assume that $O(\alpha) = \{q_n \mid n \geq 0\} \subseteq F$ for some F in $\mathscr{F}$. By definition, $F \in \mathscr{F}_0$ and therefore $F \in \mathscr{F}_n$ for all $n \geq 0$. Hence $O(\alpha^s) \subseteq F^s$. Conversely assume that $O(\alpha^s) \subseteq F^s$. Since $\mathscr{F}_{n+1} \subseteq \mathscr{F}_n$ and $\mathscr{F}_n \neq \emptyset$ for all $n \geq 0$, there exists an F such that $F \in \mathscr{F}_n$ for all $n \geq 0$. By the definition of Δ^s, $q_n \in F$ for all $n \geq 0$. Hence $O(\alpha) \subseteq F$. □

5. Comparison with the nondeterministic classes

In this section we determine the classes $\mathscr{A}_i$, hence $\mathscr{A}_i^s$, for $i = 1, \ldots, 4$.

Theorem 5.1. $\mathscr{A}_1 = \mathscr{N}_1$.

Proof. The containment $\mathscr{N}_1 \subseteq \mathscr{A}_1$ is obvious. Let L be an ω-language in $\mathscr{A}_1$ and let $M = (Q, f, \Sigma, \Delta, q_0, \{F\})$ be a simple *afa* which recognizes L in the sense of C_1. To show the containment $\mathscr{A}_1 \subseteq \mathscr{N}_1$ we construct a nondeterministic finite automaton M' which recognizes L in the sense of C_1. Before describing M' we need several preliminary notations and observations. Let x be in Σ^ω and let T be a computation tree of M on x. For a node u of T we denote by $T(u)$ the subtree of T with root u. For $n \geq 1$, $V(n, T)$ denotes the set of the nodes in level n. Given $n \geq m \geq 1$, we divide $V(n, T)$ as follows:

(1) $V_0(n, m, T) = \{u \mid u$ is in $V(n, T)$ and the path between u and the node in level m does not hold any element in $F\}$.

(2) $V_1(n, m, T) = V(n, T) - V_0(n, m, T)$.

$V_0(n, m, T) = \emptyset$ means that all the paths between level n and level m hold some elements in F. For $n \geq m \geq 1$ we define an equivalence relation $R(n, m, T)$ on $V(n, T)$ by $uR(n, m, T)v$ if (3) and (4) hold.

(3) Both u and v are in $V_0(n, m, T)$ or both u and v are in $V_1(n, m, T)$.

(4) u and v hold the same label.

Assume that M accepts x in the sense of C_1. Then we show that there exist an accepting computation tree T and a sequence $0 = r_0 < r_1 < \cdots < r_n < \cdots$ of integers satisfying (5) and (6).

(5) $r_{n+1} = \min\{m \mid m > r_n$ and $V_0(m, r_n + 1, T) = \emptyset\}$ for $n \geq 0$.

(6) For each n, m with $r_n < m \leq r_{n+1}$, if $uR(m, r_n + 1, T)v$, then the transitions taken at u are the same as those at v.

In order to show that such T and $\{r_n\}_{n=0}^{\infty}$ exist, we define a sequence $\{T_n\}_{n=0}^{\infty}$ of accepting computation trees of M on x. We define T_n and r_n inductively. Let $r_0=0$ and T_0 be an arbitrary accepting computation tree of M on x. Assume that T_n and r_m are defined but T_{n+1} and r_{m+1} are not. Moreover assume that $n \geqslant r_m$ and $V_0(l, r_m+1, T_n) \subseteqq V_0(l, r_m+1, T_{r_m})$ for $l > r_m$. Let $C[u_1], \ldots, C[u_k] \subseteqq V(n+1, T_n)$ be the equivalence classes of $R(n+1, r_m+1, T_n)$, where u_j $(j=1, \ldots, k)$ are their representative nodes in level $n+1$. Then, in T_n, for each j $(j=1, \ldots, k)$ we replace the subtree $T_n(u)$ at u by $T_n(u_j)$ for all u in $C[u_j]$. Let T_{n+1} be the tree obtained by the above replacement. Then notice that T_{n+1} is an accepting computation tree of M on x. If $V_0(n+1, r_m+1, T_n)=\emptyset$, then we define $r_{m+1}=n+1$ else r_{m+1} is not yet defined. Since T_{r_m} is an accepting computation tree, there exists an integer $t > r_m$ such that $V_0(t, r_m+1, T_{r_m})=\emptyset$. By (3) and (4) we see that $V_0(l, r_m+1, T_{n+1}) \subseteqq V_0(l, r_m+1, T_n)$ for all $l > r_m$. Therefore, $V_0(l, r_m+1, T_{n+1}) \subseteqq V_0(l, r_m+1, T_{r_m})$. Since $V_0(t, r_m+1, T_{r_m})=\emptyset$, r_{m+1} will be eventually defined. Then note that r_{m+1} is defined to be $\min\{l \mid l > r_m$ and $V_0(l, r_m+1, T_l)=\emptyset\}$. Hence $\{r_n\}_{n=0}^{\infty}$ and $\{T_n\}_{n=0}^{\infty}$ are defined. Then there exists a computation tree T such that T coincides with T_n up to level n for all $n \geqslant 0$. Then, by the choice of $\{r_n\}_{n=0}^{\infty}$, $V_0(r_{n+1}, r_n+1, T)=\emptyset$ for all $n \geqslant 0$. Hence T is an accepting computation tree of M on x. Obviously, T and $\{r_n\}_{n=0}^{\infty}$ satisfy (5) and (6).

We now describe the moves of M'. Let $x=x_1x_2x_3\ldots$ be in Σ^{ω}. If M accepts x in the sense of C_1, then by the above observation there exist an accepting computation tree T and a sequence $\{r_n\}_{n=0}^{\infty}$ satisfying (5) and (6). M' has two variables Q_0 and Q_1 that take subsets of Q as values. M' on x simulates T level by level. In simulating the level n $(r_m < n \leqslant r_{m+1})$ of T, M' keeps the labels of the nodes in $V_i(n, r_m+1, T)$ in Q_i for $i=0, 1$. By (3)–(6), M' need not keep all the nodes in level n. Initially, if q_0 is in F, then $Q_0=\emptyset$ and $Q_1=\{q_0\}$ else $Q_0=\{q_0\}$ and $Q_1=\emptyset$. Assume that, in level n, $Q_0=Q_0^{(n)}$ and $Q_1=Q_1^{(n)}$. In level $n+1$, variables Q_0 and Q_1 are changed in the following way:

(1) $Q_0^{(n)} \neq \emptyset$: For each universal state q in $Q_0^{(n)}$ and for each state p in $\Delta(q, x_n)$, if p is in F, then p is put into Q_1 else into Q_0. For each existential state q in $Q_0^{(n)}$, M' nondeterministically chooses a state p in $\Delta(q, x_n)$ and puts p into Q_1 if p is in F else into Q_0. For each universal state q in $Q_1^{(n)}$ all states in $\Delta(q, x_n)$ are put into Q_1 and, for each existential state q in $Q_1^{(n)}$, exactly one state is nondeterministically chosen from $\Delta(q, x_n)$ and is put into Q_1.

(2) $Q_0^{(n)}=\emptyset$: In this case we say that M' is in the reset mode. For each universal state q in $Q_1^{(n)}$ and for each state p in $\Delta(q, x_n)$, if p is in F, then p is put into Q_1 else into Q_0. For each existential state q in $Q_1^{(n)}$, a state p in $\Delta(q, x_n)$ is nondeterministically chosen and M' puts p into Q_1 if p is in F else into Q_0.

Then it should be noticed that M' on x can pass the reset mode infinitely often. Conversely, if M' on x can pass the reset mode infinitely often, then it is easily observed that there is an accepting computation tree of M on x in the sense of C_1. By this observation, we define the set F' of final states of M' by the set of states in the reset mode. Then L is recognized by M' in the sense of C_1. □

Theorem 5.2. $\mathscr{A}_3 = \mathscr{N}_3$.

Proof. For a simple *afa* M, let M' be the same simple *nfa* as the one constructed in Theorem 5.1. Then we can show in a similar way that M' recognizes the same ω-language as M in the sense of C_3. $\square$

Theorem 5.3. $\mathscr{A}_4 = \mathscr{N}_4$.

Proof. Let $M = (Q, f, \Sigma, \Delta, q_0, \{F\})$ be a simple *afa*. A simple *nfa* M' that recognizes the same ω-language as M in the sense of C_4 moves as follows: M' has a variable S which takes a subset of Q as a value. Assume that an input $x = x_1x_2x_3\ldots$ is given. Initially M' puts $S = \{q_0\}$. Assume that, at step n, $S = S_n$. S is updated as follows: For each universal state q in S_n, all states in $\Delta(q, x_n)$ are put into S and for each existential state q in S_n, exactly one state in $\Delta(q, x_n)$ is chosen and is put into S. The set F' of final states of M' is $\{Q' \mid \emptyset \neq Q' \subseteq F\}$. Then it is easily observed that M accepts x in the sense of C_4 if and only if M' accepts x in the sense of C_4. $\square$

Lemma 5.4 ([1]). *$\mathscr{N}_1$ is closed under complement.*

Theorem 5.5. $\mathscr{A}_1 = \mathscr{A}_2$.

Proof. We show that, for $L \subseteq \Sigma^\omega$, L is in $\mathscr{A}_1$ if and only if $\bar{L} = \Sigma^\omega - L$ is in $\mathscr{A}_2$. We first prove that $L \in \mathscr{A}_1$ implies $\bar{L} \in \mathscr{A}_2$. Assume that L is in $\mathscr{A}_1$ and let $M = (Q, f, \Sigma, \Delta, q_0, \{F\})$ be a simple *afa* that recognizes L in the sense of C_1. We show that $\bar{L}$ is in $\mathscr{A}_2$. By definition, the following two statements are equivalent:

(1) x is in $\bar{L}$.

(2) For any computation tree $T(M, x)$ there exists a run α in $T(M, x)$ such that $I(\alpha) \subseteq \bar{F}$, where $\bar{F} = Q - F$.

Then consider a simple *afa* $\bar{M} = (Q, \bar{f}, \Sigma, \Delta, q_0, \{\bar{F}\})$, where $\bar{f}(q) = $ *and* if $f(q) = $ *or*, $\bar{f}(q) = $ *or* if $f(q) = $ *and*. Then notice that the following statement is equivalent to (2):

(3) There exists a computation tree $T(\bar{M}, x)$ of $\bar{M}$ on x such that, for any run α in $T(\bar{M}, x)$, $I(\alpha) \subseteq \bar{F}$.

Hence $\bar{L}$ is in $\mathscr{A}_2$. The converse implication can be proved in the same way. By Theorem 5.1 and Lemma 5.4, $\mathscr{A}_1$ is closed under complement. Therefore by the above equivalence we see that $\mathscr{A}_1 = \mathscr{A}_2$. $\square$

By Theorems 5.1–5.3, 5.5 and the results already known for $\mathscr{N}_i$ ($i = 1, \ldots, 4$), the proof of Table 1 is completed.

6. Concluding remarks

In the definition of alternating finite automata we have assumed that $\Delta(q, a) \neq \emptyset$ for each q in Q and each a in Σ. In the literature [10, 19], the case allowing $\Delta(q, a) = \emptyset$

has also been discussed. We call an *afa* which allows the case $\Delta(q, a) = \emptyset$ a *partial afa*; *partial nfa* and *partial dfa* can be defined similarly. Let $\mathcal{PD}_i$, $\mathcal{PN}_i$, $\mathcal{PA}_i$ ($i = 1, \ldots, 4$) be the classes of ω-languages recognized by partial *dfa*'s, partial *nfa*'s, partial *afa*'s in the sense of C_i, respectively. It is not hard to show that the partial and nonpartial classes are equal under C_1, C_2 and C_4 type acceptances for deterministic, nondeterministic and alternating cases. This can be shown by constructing a nonpartial finite automaton by adding a trapping state into which all undefined transitions come. For C_3 type acceptance, $\mathcal{PD}_3$ and $\mathcal{PN}_3$ have been already determined [19]. To determine $\mathcal{PA}_3$, first note that the class of ω-languages recognized by simple partial *afa*'s in the sense of C_3 is equal to $\mathcal{PA}_3$. Then we can prove that $\mathcal{PA}_3 = \mathcal{PN}_3$ by taking care of the undefined transitions in the construction of the *nfa* in the proof of Theorem 5.1.

Table 2 summarizes the classes for partial finite automata on ω-words.

Table 2 [a]

	C_1	C_2	C_3	C_4
$\mathcal{PD}_i$	G_δ^R	F_σ^R	$G^R \Delta F^R$	F^R
$\mathcal{PN}_i$	R	F_σ^R	F_σ^R	F^R
$\mathcal{PA}_i$	R	R	F_σ^R	F^R

[a] $G^R \Delta F^R = \{L_1 \cap L_2 \mid L_1 \in G^R, L_2 \in F^R\}$.

We have not examined the acceptances defined by the following conditions:

(C_5) $I(\alpha) \in \mathcal{F}$.

(C_6) $O(\alpha) \in \mathcal{F}$.

In [19], the ω-languages defined by deterministic finite automata under C_5 and C_6 type acceptances were thoroughly studied.

References

[1] J.R. Büchi, On an decision method in restricted second order arithmetic, *Internat. Congress on Logic Methodology and Philosophy of Science* (Stanford, Univ. Press, CA, 1960).

[2] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer, Alternation, *J. Assoc. Comput. Mach.* **28** (1981) 114–133.

[3] Y. Choueka, Theories of automata on ω-tapes: A simplified approach, *J. Comput. System Sci.* **8** (1974) 117–142.

[4] R.S. Cohen and A.Y. Gold, Theory of ω-languages. I: Characterizations of ω-context-free languages, *J. Comput. System Sci.* **15** (1977) 169–184.

[5] R.S. Cohen and A.Y. Gold, Theory of ω-languages. II: A study of various models of ω-type generation and recognition, *J. Comput. System Sci* **15** (1977) 185–208.

[6] R.S. Cohen and A.Y. Gold, ω-computations on deterministic pushdown machines, *J. Comput. System Sci.* **16** (1978) 275–300.

[7] R.S. Cohen and A.Y. Gold, ω-computations on Turing machines, *Theoret. Comput. Sci.* **6** (1978) 1–23.

[8] S. Eilenberg, *Automata, Languages, and Machines, Vol. A* (Academic Press, New York, 1974).
[9] J. Hartmanis and R.E. Stearns, Set of numbers defined by finite automata, *Amer. Math. Monthly* **74** (1967) 539–542.
[10] R. Hossley, Finite tree automata and ω-automata, TR MAC TR-102, MIT, Cambridge, MA, 1972.
[11] K. Kobayashi, M. Takahashi and H. Yamasaki, Characterization of ω-regular languages by first-order formulas, TIT Research Report C-48, 1983; *J. Comput. System Sci.*, to appear.
[12] R.E. Ladner, R.J. Lipton and L.J. Stockmeyer, Alternating pushdown automata, *Proc. 19th IEEE Symp. on Foundations of Computer Science* (1978) 92–106.
[13] L.H. Landweber, Decision problems for ω-automata, *Math. Systems Theory* **4** (1969) 376–384.
[14] R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Inform. Contr.* **9** (1966) 521–530.
[15] M. Nivat, Behaviours of processes and synchronized systems of processes, Lecture Note, Marktoberdopf NATO Summer School, 1981.
[16] M.O. Rabin, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141** (1969) 1–35.
[17] L. Staiger and K. Wagner, Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regularer Folgenmengen, *EIK* **10** (1974) 379–392.
[18] K. Wagner, Eine topologischer Charakterisierung einiger Klassen regularer Folgenmengen, *EIK* **13** (1977) 473–487.
[19] K. Wagner, On ω-regular sets, *Inform. Contr.* **43** (1979) 123–177.