Intern. J. Computer Math., Vol. 35, pp. 117–132
Reprints available directly from the publisher
Photocopying permitted by license only

CONSTRUCTIONS FOR ALTERNATING FINITE AUTOMATA*

A. FELLAH

Department of Mathematics and Computer Science, Kent State University, Kent, Ohio 44242, USA

H. JÜRGENSEN and S. YU†

Department of Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5B7

(*Received 25 January 1990; in final form 16 February 1990*)

Alternation is a natural generalization of nondeterminism. The model of alternating finite automata was first introduced and studied by Chandra *et al.* in [2]. Although alternating finite automata are no more powerful than deterministic finite automata with respect to language recognition, special features of alternating finite automata may provide new approaches and techniques for solving theoretical and practical problems concerning regular languages. In this paper we present direct constructions for the usual language theoretic operations in terms of alternating finite automata. Moreover, we discuss minimization and direct transformations between alternating, non-deterministic, and deterministic finite automata.

KEY WORDS: Alternating finite automata, finite automata, alternation, nondeterminism, NFA with nondeterministic starting state.

C.R. CATEGORIES: F.1.1, F.1.2.

1. INTRODUCTION

The notion of alternation is a natural generalization of non-determinism. It received its first formal treatment in [1], published as [2]. That seminal paper and most of the subsequent research focused on relating various types of alternating machines to complexity classes (see, for example, [1, 5–15]). Such machines are useful for a better understanding of many questions in complexity theory. For alternating (one-way, single-head) finite automata, it is proved in [2] that they are precisely as powerful as deterministic finite automata as far as language recognition is concerned. This result would seem to close the story. However, beyond this seemingly negative result one should still ask the question whether the presence of alternation can lead to simplified constructions in the area of finite automata. This is the focus of the present paper.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada, Grants OGP0000243 and OGP0041630.

†On leave from Kent State University, Kent, Ohio, USA.

2. PRELIMINARIES AND NOTATIONS

In this section we introduce the basic notions and notations used in this paper. The symbol $\mathbb{B}$ denotes the Boolean semiring, $\mathbb{B}=\{0,1\}$. Let Q be a set. Then $\mathbb{B}^Q$ is the set of all mappings of Q into $\mathbb{B}$; note that $u\in\mathbb{B}^Q$ can also be considered as a Q-vector over $\mathbb{B}$. For $u\in\mathbb{B}^Q$ and $q\in Q$ we write qu or u_q to denote the image of q under u (depending on the context). If P is a subset of Q then $u|_P$ is the restriction of u to P.

An *alphabet* is a finite, nonempty set. Without loss of generality we assume in the sequel that alphabets do not contain any of the "special" symbols

$$\emptyset, \varepsilon, \cup, *, \vee, \wedge, ^{-}, +, \cdot, (,).$$

The elements of an alphabet are called *symbols* or *letters*. A *word* over an alphabet Σ is a finite sequence of symbols from Σ. We use ε to denote the *empty word* and Σ^* to denote the set of all words over Σ.

Any subset of Σ^* is called a *language* over Σ. For languages L_1 and L_2 over Σ we consider the operations of complement $\bar{L}_1=\Sigma^*\backslash L_1$, union $L_1\cup L_2$, intersection $L_1\cap L_2$, concatenation L_1L_2, power L_1^n, and star $L_1^*=\bigcup_{i=0}^{\infty}L_1^i$. We denote the *reversal* of a word w by w^R. The reversal of L is defined as $L^R=\{w^R|w\in L\}$.

A *deterministic finite automaton* (DFA) M is specified by a quintuple $M=(Q,\Sigma,\delta,s,F)$ where Q is the finite set of states, Σ is the input alphabet, $\delta: Q\times\Sigma\to Q$ is the transition function, $s\in Q$ is the starting state, and $F\subseteq Q$ is the set of final states. A *non-deterministic finite automaton* (NFA) $M=(Q,\Sigma,\delta,s,F)$ is similar to a DFA, except that δ is a mapping of $Q\times\Sigma$ into 2^Q. An NFA with non-deterministic starting state (NNFA) is defined by a quintuple $M=(Q,\Sigma,\delta,S,F)$ where $S\subseteq Q$ is the set of starting states. Let M as above be a DFA, an NFA, or an NNFA. Without loss of generality, assume that Σ and Q are disjoint. A *configuration* of M is a word in $Q\Sigma^*$. Let $v,w\in\Sigma^*$ and $p,q\in Q$. By $pv\vdash qw$, $pv\vdash^i qw$, and $pv\vdash^* qw$ we denote the facts that there exist single-step, i-step, or arbitrary-step transitions, respectively, from pv to qw.

For general information about finite automata and regular languages see [4] or [16], for example.

3. DEFINITION OF AN ALTERNATING FINITE AUTOMATON

Alternating automata (of any type) are a generalization of non-deterministic automata in the following sense: If in a given state q the automaton reads an input symbol a, it will activate all states of the automaton to work on the remaining part of the input in parallel. Once the states have completed their tasks, q will evaluate their results using a Boolean function and pass on the resulting value to the state by which it was activated. A word w is accepted if the starting state computes the value of 1. Otherwise, it is rejected. We now formalize this idea.

DEFINITION 3.1 An *alternating finite automaton* (AFA) is a quintuple

$$A(Q, \Sigma, s, F, g)$$

with the following properties:

a) Q is a finite set, the set of *states*;

b) Σ is an alphabet, the *input alphabet*;

c) $s \in Q$ is the *starting state*;

d) $F \subseteq Q$ is the set of *final states*;

e) g is a mapping of Q into the set of all mappings of $\Sigma \times \mathbb{B}^Q$ into $\mathbb{B}$.

We turn to defining the sequential behavior of an AFA. For $q \in Q$, let

$$g_q : \Sigma \times \mathbb{B}^Q \to \mathbb{B} : (a, u) \mapsto g(q)(a, u)$$

where $a \in \Sigma$ and $u \in \mathbb{B}^Q$. Later we also need the mappings $g(a)$ of Q into the set of all mappings of $\mathbb{B}^Q$ into $\mathbb{B}$ and the mappings $g_q(a)$ of $\mathbb{B}^Q$ into $\mathbb{B}$ defined by

$$g(a)(q)(u) = g_q(a)(u) = g_q(a, u)$$

for $a \in \Sigma$, $q \in Q$, and $u \in \mathbb{B}^Q$.

Now define $f \in \mathbb{B}^Q$ by the condition

$$f_q = 1 \Leftrightarrow q \in F.$$

f is called the *characteristic vector* of F. We extend g to a mapping of Q into the set of all mappings of $\Sigma^* \times \mathbb{B}^Q$ into $\mathbb{B}$ as follows:

$$g_q(w, \Sigma) = \begin{cases} u_q, & \text{if } w = \varepsilon, \\ g_q(a, g(v, u)), & \text{if } w = av \text{ with } a \in \Sigma \text{ and } v \in \Sigma^*, \end{cases}$$

where $w \in \Sigma u^*$ and $u \in \mathbb{B}^Q$.

Definition 3.2 Let $A = (Q, \Sigma, s, F, g)$ be an AFA. A word $w \in \Sigma^*$ is *accepted* by A if and only if $g_s(w, f) = 1$. The *language accepted* by A is the set $L(A) = \{w \mid w \in \Sigma^* \wedge g_s(w, f) = 1\}$.

Occasionally, we need to change the start state of an AFA. In this case, $L_q(A) = \{w \mid w \in \Sigma^* \wedge g_q(w, f) = 1\}$ is the language accepted by A with start state q. We summarize the results of [2] for AFA.

Theorem 3.3 [2] *If A is an AFA with k states then $L(A)$ is accepted by a DFA with 2^{2^k} states and $L(A)^R$ is accepted by a DFA with 2^k states. Moreover, for every k, $k \geqq 1$, there is an AFA A with k states such that the reduced DFA accepting $L(A)$ has exactly 2^{2^k} states.*

The first part of this result is not surprising as any AFA can enter only finitely many "internal situations," a notion to be made precise later. However, it also provides valuable information about the relative sizes of the state sets of an AFA and a DFA.

In the remainder of this paper we investigate certain natural automaton theoretic constructions in detail. In Sections 4 and 5, we present two direct constructions of an NFA from a given AFA; the former actually results in an NNFA in its usual form; the latter yields an NNFA or an NFA in its matrix representation. In Section 6, we explore the converse construction to represent a given DFA or NFA as an AFA. Section 7 focuses on normal forms and minimization of AFA. One of the results states that the class of ε-free regular languages coincides with the class of languages accepted by AFA without final states. In Section 8, we provide direct AFA constructions for union, intersection, and complement. The operations of concatenation and star are dealt with in Section 9. Finally, Section 10 summarizes the results and addresses further questions.

4. AFA TO NFA—THE FIRST CONSTRUCTION

The proof of Theorem 3.3. given in [2], though being constructive in principle, provides little insight in the way a DFA or an NFA would simulate an AFA. In this section we exhibit one construction of an NFA from a given AFA. Another quite different construction is given in a later section. These constructions illustrate various aspects of the connection between AFA and NFA.

Let $A=(Q,\Sigma,s,F,g)$ be an AFA. Consider the NNFA

$$A_v=(\mathbb{B}^Q,\Sigma,\delta,S,\{f\})$$

where

$$(u,a,u')\in\delta\Leftrightarrow g(a,u')=u$$

for $u,u'\in\mathbb{B}^Q$ and $a\in\Sigma$, and where

$$S=\{u\,|\,u\in\mathbb{B}^Q,u_s=1\}.$$

THEOREM 4.1 $L(A_v)=L(A)$.

Proof As an auxiliary result, we prove that

$$uw\vdash^* f\Leftrightarrow g(w,f)=u$$

for $u\in\mathbb{B}^Q$ and $w\in\Sigma^*$. For $w=\varepsilon$, one has $u=f$ and $g(\varepsilon,f)=f$. Now assume the statement is correct for all words up to length l, and let $w=av$ with $a\in\Sigma$ and $v\in\Sigma^l$.

Let $u=g(w,f)$. By the definition of g, one has

$$u_q=g_q(w,f)=g_q(a,g(v,f)).$$

Let $u' = g(v, f)$. The induction hypothesis implies that

$$u'v \vdash^* f,$$

hence, by the definition of δ one obtains

$$uw = uav \vdash u'v \vdash^* f.$$

For the converse, let $uw \vdash^* f$. Then there is a state u' of the NNFA such that

$$uw = uav \vdash u'v \vdash^* f.$$

By the induction hypothesis $u' = g(v, f)$, and $u = g(a, u')$ by the definition of δ. Hence, $u = g(w, f)$. $\square$

The construction of Theorem 4.1 gives an interpretation to the states of the AFA. A transition is possible only if a certain combination of successor states can be successful with respect to acceptance. Note that part of Theorem 3.3 follows from the above construction. If the given AFA has n states, then the NNFA we have constructed has 2^n states. Therefore, we can construct an equivalent 2^{2^n}-state DFA. One can also observe that the reversal of A_v is deterministic. This proves the following result which was already stated in [2].

COROLLARY 4.2 [2] *If L is accepted by a k state AFA then L^R is accepted by a DFA with at most 2^k states.*

5. AFA to NFA—THE SECOND CONSTRUCTION

In this section we provide another direct construction of an NFA from a given AFA. This second one yields an NNFA in its matrix representation. At the end of this section, we show that using a slightly modified approach an NFA (rather than an NNFA) in its matrix representation can be constructed which has one more state.

Recall that any NNFA $A = (Q, \Sigma, \delta, S, F)$ can be represented using a homomorphism μ of Σ^* into the multiplicative monoid $\mathbb{B}^{Q \times Q}$ of all $Q \times Q$-matrices over $\mathbb{B}$, and using a row vector $\pi \in \mathbb{B}^{1 \times Q}$ and a column vector $\eta \in \mathbb{B}^{Q \times 1}$, such that

$$w \in L(A) \Leftrightarrow \pi\mu(w)\eta = 1$$

for $w \in \Sigma^*$. Note that μ is completely defined by its restriction to Σ. This representation is achieved by

$$\pi_q = 1 \Leftrightarrow q \in s,$$

$$\eta_q = 1 \Leftrightarrow q \in F,$$

and

$$\mu(a)_{p,q}=1 \Leftrightarrow (p,a,q)\in\delta$$

where $p,q\in Q$ and $a\in\Sigma$.

Now, let $A=(Q,\Sigma,s,F,g)$ be an AFA. Let $X=\{x_q|q\in Q\}$ and $\bar{X}=\{\bar{x}_q|q\in Q\}$ be two disjoint alphabets, each in one-to-one correspondence with Q. The elements X will be used as *variables* in Boolean expressions with $^-$ denoting negation as usual. Let T be the set of terms of the form $\bigwedge_{q\in Q}y_q$ where $y_q\in\{x_q,\bar{x}_q\}$ for $q\in Q$. Every $t\in T$ can be considered as a mapping of $\mathbb{B}^Q$ into $\mathbb{B}$ as follows: Suppose $t=\bigwedge_{q\in Q}y_q$ and $u\in\mathbb{B}^Q$. Then $t(u)=\bigwedge_{q\in Q}z_q$ where

$$z_q=\begin{cases}u_q, & \text{if } y_q=x_q,\\ \bar{u}_q, & \text{if } y_q=\bar{x}_q.\end{cases}$$

One then extends every $t\in T$ to a mapping of $(\mathbb{B}^Q)^Q$ into $\mathbb{B}^Q$, that is to a functional, in the obvious manner.

The set T serves as the set of states of the NNFA to be constructed. For $a\in\Sigma$, define a $T\times T$-matrix $\mu(a)$ by the condition

$$\mu(a)_{r,t}=1\Leftrightarrow t \text{ implies } r(g(a))$$

where $a\in\Sigma$ and $r,t\in T$. Extend the definition of μ to a homomorphism of Σ^* into $\mathbb{B}^{T\times T}$ in the natural way. Let η be the column vector over $\mathbb{B}$ with $\eta_t=t(f)$ for $t\in T$. Finally, let π be the row vector with $\pi_t=1$ if and only if t implies x_s.

Let $A_m=(T,\Sigma,\delta_m,S_m,F_m)$ be the NNFA having the triple (π,μ,η) as its matrix representation.

THEOREM 5.1. $L(A_m)=L(A)$.

Proof For $w\in\Sigma^*$ and $u\in\mathbb{B}^Q$, define the mapping $g(w,u)$ of Q into $\mathbb{B}$ by $g(w,u)(q)=g_q(w,u)$ where $q\in Q$. Clearly, $g(av,u)=g(a,g(v,u))$, for any $a\in\Sigma$, $v\in\Sigma^*$, and $u\in\mathbb{B}^Q$. One proves by induction that

$$t(g(w,f))=(\mu(w)\eta)_t$$

for all $w\in\Sigma^*$ and $t\in T$. Indeed, for $w=\varepsilon$ one has $t(g(w,f))=t(f)=\eta_t=(\mu(\varepsilon)\eta)_t$. Now consider $w=av$ with $a\in\Sigma$ and $v\in\Sigma^*$. By induction hypothesis, $r(g(v,f))=(\mu(v)\eta)_r$ for all $r\in T$. Let $u=g(v,f)$. Then $t(g(w,f))=t(g(a,u))$. One obtains:

$$\begin{aligned} t(g(a,u))=1 &\Leftrightarrow \exists r\in T: r \text{ implies } t(g(a)) \text{ and } r(u)=1\\ &\Leftrightarrow \mu(a)_{t,r}=1 \text{ and } (\mu(v)\eta)_r=1\\ &\Leftrightarrow (\mu(a)(\mu(v)\eta))_t=(\mu(w)\eta)_t=1.\end{aligned}$$

To prove $L(A)=L(A_m)$, we still have to show that $g_s(w,f)=\pi\mu(w)\eta$ for $w\in\Sigma^*$:

$$
\begin{aligned}
g_s(w,f)=1 &\Leftrightarrow x_s=1 \\
&\Leftrightarrow \exists t \in T{:}\, t \text{ implies } x_s \text{ and } t(g(w,f))=1 \\
&\Leftrightarrow \pi_t=1 \text{ and } (\mu(w)\eta)_t=1 \\
&\Leftrightarrow \pi\mu(w)\eta=1.
\end{aligned}
$$

□

Note that A_m *has* 2^k states if $|Q|=k$. By the last part of Theorem 3.3 this cannot be improved in general. Indeed, let be a k-state AFA such that the smallest equivalent DFA as 2^{2^k} states and let A' be an equivalent NNFA with j states. Then A' has an equivalent DFA with a most 2^j states, that is, $2^j \geqq 2^{2^k}$, hence $j \geqq 2^k$.

COROLLARY 5.2 *For every $k, k \geqq 1$, and every k-state AFA there exists an equivalent NNFA with at most 2^k states. Moreover, this bound is tight for all k.*

While the above construction yields an NNFA, a slight modification of it yields an equivalent NFA with one more state. The state set of the NFA is $T'=T \cup \{\alpha\}$ where α is a new symbol. For $a \in \Sigma$, define $\mu'(a)$ to be a $T' \times T'$ matrix such that $\mu'(a)_{r,t}=\mu(a)_{r,t}$ for all $r,t \in T$, and $\mu'(a)_{\alpha,t}=1$ if and only if t implies $g_s(a)$ and $\mu'(a)_{t,\alpha}=0$ for all $t \in T$. The row vector π' and the column vector η' over B are defined by

$$
\pi_t=\begin{cases} 1, & \text{if } t=\alpha, \\ 0, & \text{if } t \in T, \end{cases} \quad \text{and} \quad \eta_t=\begin{cases} 0, & \text{if } t=\alpha \text{ and } \varepsilon \notin L(A), \\ 1, & \text{if } t=\alpha \text{ and } \varepsilon \in L(A), \\ t(f), & \text{if } t \in T, \end{cases}
$$

for $t \in T'$. One verifies that again

$$
\pi'\mu'(w)\eta'=1 \Leftrightarrow w \in L(A)
$$

which proves the claim.

COROLLARY 5.3 *The NFA A' given by π', μ', and η' is equivalent to the AFA A. Moreover, if A has k states then A' has 2^k+1 states.*

Thus every k-state AFA has an equivalent NFA with at most 2^k+1 states. We conjecture that this bound is tight. However, we do not have a proof. Note that this would not contradict the fact that 2^{2^k} states in an equivalent DFA are enough.

6. DFA OR NFA TO AFA

In this section we show, how a DFA or NFA can be represented in the AFA formalism. Let $A=(Q,\Sigma,\delta,s,F)$ be an NFA. We define an AFA

$$A_a=(Q,\Sigma,s,F,g)$$

as follows. Let

$$g_q(a,u)=0 \Leftrightarrow u_p=0 \text{ for all } p\in Q \text{ with } (q,a,p)\in\delta$$

where $q\in Q$, $a\in\Sigma$, and $u\in\mathbb{B}^Q$.

THEOREM 6.1 $L(A_a)=L(A)$.

Proof By induction, one proves the more general statement that

$$g_q(w,f)=1 \Leftrightarrow qw\vdash^* p \text{ for some } p\in F$$

where $q\in Q$ and $w\in\Sigma^*$. If $w=\varepsilon$ then $g_q(w,f)=f_q$, and $f_q=1$ holds if and only if $q\in F$, that is, $\varepsilon\in L(A)$. Now suppose that $w=av$ where $a\in\Sigma$ and $v\in\Sigma^*$. Then one has

$$\begin{aligned} g_q(w,f)=g_q(av,f)=1 &\Leftrightarrow g_q(a,g(v,f))=1 \\ &\Leftrightarrow \exists p\in Q: qa\vdash p \text{ and } g_p(v,f)=1 \\ &\Leftrightarrow \exists p\in Q\,\exists r\in F: qa\vdash p \text{ and } pv\vdash^* r, \end{aligned}$$

hence, using the induction hypothesis,

$$\Leftrightarrow \exists r\in F: qw=qav\vdash^* r.$$

□

The above construction starts from an NFA or a DFA. To start from an NNFA $A=(Q,\Sigma,\delta,S,F)$ one introduces a new state $s, s\notin Q$, and defines the AFA $A_a=(Q',\Sigma,s,F,g)$ as above with $Q'=Q\cup\{s\}$ and $g_s(a,u)=\bigvee_{p\in S} g_p(a,u)$.

7. NORMAL FORMS AND MINIMIZATION OF AFA

In this section, we show that every AFA has an equivalent AFA with at most one final state and we show how to transform an AFA into an equivalent one in which negation is not used. We also consider a special kind of AFA, the so-called s-AFA. We then prove a theorem concerning the minimum number of states in an s-AFA.

THEOREM 7.1 *For any AFA A with k states, $k>0$, there exists an equivalent k-state AFA A' with at most one final state. More precisely, A' has no final state if $\varepsilon\notin L(A)$; otherwise, the only final state of A' coincides with the start state.*

Proof Let $A=(Q,\Sigma,s,F,g)$ be a k-state AFA. For $u\in\mathbb{B}^Q$, let $u'\in\mathbb{B}^Q$ be given by

$$u'_q = \begin{cases} \bar{u}_q, & \text{if } q \in F \setminus \{s\}, \\ u_q, & \text{otherwise,} \end{cases}$$

for $q \in Q$. We construct an AFA $A' = (Q, \Sigma, s, F', g)$ with

$$F' = \begin{cases} \{s\}, & \text{if } s \in F, \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$g'_q(a, u) = \begin{cases} \overline{g_q(a, u')}, & \text{if } q \in F \setminus \{s\}, \\ g_q(a, u'), & \text{otherwise.} \end{cases}$$

Let f' be the characteristic vector of F'. Obviously, this definition of f' is consistent with the above. By induction on the length of $w \in \Sigma^*$, one shows that

$$(g(w, f))' = g'(w, f').$$

As $u_s = u'_s$ for all $u \in \mathbb{B}^Q$, this implies that $g_s(w, f) = g'_s(w, f')$ for all $w \in \Sigma^*$. Therefore, $L(A) = L(A')$. □

At first glance, this result is quite surprising as it allows for non-empty languages being accepted by AFA without final states. More precisely, one has:

COROLLARY 7.2 *The class of languages accepted by AFA without final states coincides with the class of ε-free regular languages.*

This result is in contrast to the situation for DFA, but actually parallels the situation for NFA. For DFA, the family of languages accepted with k final states is properly contained in the family of languages accepted with $k+1$ final states for all k, $k \geq 0$. For NFA, 2 final states are always sufficient. More precisely, the ε-free regular languages require at most one final state while the regular languages containing the empty word may need two final states.

The next results states that negations in the Boolean functions defining an AFA can be avoided at the cost of increasing the number of states by a factor of 2.

THEOREM 7.3 *For every AFA $A = (Q, \Sigma, s, F, g)$ one can construct an equivalent AFA $A' = (Q', \Sigma, s', F', g')$ with $|Q'| = 2|Q|$ such that, for every $q \in Q'$, g'_q is without negations, that is, can be defined using $\wedge$ and $\vee$ only.*

Proof For each $q \in Q$ let q' be a new symbol and let

$$Q' = Q \cup \{q' \mid q \in Q\},$$

$$s' = s,$$

$$F' = F \cup \{q' \mid q \in Q \setminus F\},$$

and

$$g'_p = \begin{cases} g_p, & \text{if } p \in Q, \\ \bar{g}_q, & \text{if } p = q' \text{ with } q \in Q. \end{cases}$$

One verifies by induction that $g'_{q'}(w) = \overline{g_q(w)}$ for all $w \in \Sigma^*$.

So far every g'_p can be considered as given by a Boolean expression involving negations only at the level of the variables x_q for $q \in Q$. In order to obtain a Boolean expression without negations for g'_p, one replaces every occurrence of a negated variable by the corresponding primed variable, that is, $\bar{x}_q$ would be replaced by $x_{q'}$. This does not change the AFA A'. □

In the sequel we refer to an AFA whose functions g_q can be defined using $\wedge$ and $\vee$ only as an *AFA without negations.* As far as final states are concerned, AFA without negations are similar to NFA.

THEOREM 7.4 *Let A be a k-state AFA without negations. One can construct an equivalent $(k+1)$-state AFA without negations which has 1 final state if $\varepsilon \notin L(A)$ and at mot 2 final states otherwise.*

Proof Let $A = (Q, \Sigma, s, F, g)$ and let r be a new state. Let $q_1, \ldots, q_k$ be the states of A and, for $p \in Q$, $a \in \Sigma$, and $u \in \mathbb{B}^Q$, let $\sigma_{p,a,u}(x_1, \ldots, x_k)$ be a Boolean expression using $\wedge$ and $\vee$ only with variables x_i for the states $q_i \in Q$ such that

$$g_p(a, u) = \sigma_{p,a,u}(x_1, \ldots, x_k).$$

Let x_r be a variable corresponding to the state r. We define Boolean functions $f_p(a, u)$ by

$$f_p(a, u) = \sigma(x'_1, \ldots, x'_k)$$

where

$$x'_i = \begin{cases} x_i, & \text{if } q_i \notin F, \\ x_r, & \text{if } q_i \in F. \end{cases}$$

Now we define the AFA without negations $A' = (Q', \Sigma, s, F', g')$ by $Q' = Q \cup \{r\}$, $F' = (\{s\} \cap F) \cup \{r\}$, and

$$g'_p(a, u) = \begin{cases} 0, & \text{if } p = r, \\ g_p(a, \rho(u)) \vee f_q(a, \rho(u)), & \text{if } p \neq r, \end{cases}$$

for $p \in Q'$, $a \in \Sigma$ and $u \in \mathbb{B}^{Q'}$, where ρ is the projection from $\mathbb{B}^{Q'}$ onto $\mathbb{B}^Q$ which is induced by the inclusion $Q \subseteq Q'$. One verifies that A and A' are equivalent by induction on the length of input words. □

With respect to minimization of AFA, we only consider a special kind of AFA. An s-AFA is an AFA $A=(Q,\Sigma,s,F,g)$ such that for every $a\in\Sigma$ and every $u\in\mathbb{B}^Q$, $g(a,u)$ does not depend on u_s. Intuitively, this means that the start state s cannot be reached in any computation. Obviously, for every AFA one can construct an equivalent s-AFA which has just one more state. On the other hand, if A is a $(k+1)$-state s-AFA then there need not exist an equivalent k-state AFA. For example, the language $\{\varepsilon,a,a^2\}$ is accepted by a 3-state s-AFA, but not by any 2-state AFA.

The s-AFA are particularly useful to simplify certain constructions. For instance, given an n-state s-AFA A and an m-state s-AFA B it is particularly easy to construct $(m+n-1)$-state s-AFA C and D and an m-state s-AFA E such that $L(C)=L(A)\cup L(B)$, $L(D)=L(A)\cap L(B)$, and $L(E)=\overline{L(A)}$.

THEOREM 7.5 *L is accepted by an s-AFA with $k+1$ states if and only if L^R is accepted by a DFA with 2^k states.*

Proof Let A be a $(k+1)$-state s-AFA and $L=L(A)$. The construction of a 2^k-state DFA, which accepts L^R, is similar to the one described in the proof of Theorem 4.1.

Let $D=(Q_D,\Sigma,\delta,s_D,F_D)$ be a 2^k-state DFA and $L=L(D)$. Let $K=\{1,2,\dots,k\}$ and $K_0=K\cup\{0\}$. Without loss of generality, we assume that $Q_D=\mathbb{B}^K$ and $s_D=(0,\dots,0)$. For $u\in\mathbb{B}^{K_0}$, let $\hat{u}\in\mathbb{B}^K$ be defined by $\hat{u}_i=u_i$ for all $i\in K$. We now define a $(k+1)$-state s-AFA $A=(Q_A,\Sigma,s_A,F_A,g)$ by

$$Q_A=K_0,$$

$$s_A=0,$$

$$F_A=\begin{cases}\{0\} & \text{if } s_D\in F_D,\\ \emptyset, & \text{otherwise,}\end{cases}$$

and

$$g(a,u)_i=\begin{matrix}\delta(\hat{u},a)_i, & \text{if } i\in K,\\ 1, & \text{if } i=0 \text{ and } \hat{u}\in D_D,\\ 0, & \text{if } i=0 \text{ and } \hat{u}\notin F_D.\end{matrix}$$

for $i\in K_0$, $a\in\Sigma$ and $u\in\mathbb{B}^{K_0}$. The function g is well defined since A is deterministic. By induction on the length of $w\in\Sigma^*$, one shows that $u=g(w,f)$ if and only if $\delta(s_D,w^R)=\hat{u}$. Since $u_0=1$ if and only if $\hat{u}\in F_D$, $w\in L(A)$ if and only if $w^R\in L(D)$. □

COROLLARY 7.6 *Let A be an s-AFA such that $(L(A))^R$ is accepted by a minimized DFA with n states. Then A has at least $1+\lceil(\log_2 n)\rceil$ states.*

8. UNION, INTERSECTION AND NEGATION ON AFA

Of course, as every AFA language is regular, the class of AFA languages is closed under the Boolean operations. However, this is only a "surface" result. Rather than this type of existence result one would like to have concrete constructions for AFA.

Let $A=(Q,\Sigma,s,F,g)$ be an AFA. First we construct an AFA

$$\bar{A}=(Q,\Sigma,s,F',g')$$

such that $L(\bar{A})=\Sigma^*\backslash L(A)$. The set F' of final states is defined by the condition

$$q\in F' \Leftrightarrow \begin{cases} q\in F, & \text{if } q\neq s, \\ q\notin F, & \text{if } q=s, \end{cases}$$

for $q\in Q$. For $u\in\mathbb{B}^Q$, let u' be the mapping given by

$$u'_q=\begin{cases} u_q, & \text{if } q\neq s, \\ \bar{u}_q, & \text{if } q=s, \end{cases}$$

for $q\in Q$. The function g' is given by

$$g'_q(a,u)=\begin{cases} g_q(a,u'), & \text{if } q\neq s, \\ \overline{g_q(a,u)}, & \text{if } q=s, \end{cases}$$

where $q\in Q$, $a\in\Sigma$, and $u\in\mathbb{B}^Q$.

THEOREM 8.1 $L(\bar{A})=\Sigma^*\backslash L(A)$.

Proof We prove that $g'(w,f')=g(w,f)'$ for all $w\in\Sigma^*$. This is obviously true for $w=\varepsilon$. Now assume that the statement holds for $v\in\Sigma^*$ and consider $w=av$ with $a\in\Sigma$. For $q\in Q$, one has

$$g'_q(w,f')=g'_q(a,g(v,f')')=g'_q(a,g(v,f)')=g_q(w,f)'.$$

Thus, $g'_s(w,f')=\overline{g_s(w,f)}$, that is, $w\in L(\bar{A})\Leftrightarrow w\notin L(A)$ □

Our next construction is for the union of languages accepted by AFA. For $i=1,2$ let $A^{(i)}=(Q^{(i)},\Sigma,s^{(i)},F^{(i)},g^{(i)})$ be two AFA with disjoint state sets. We construct an AFA

$$A=A^{(1)}\vee A^{(2)}=(Q,\Sigma,s,F,g)$$

such that $L(A^{(1)}\vee A^{(2)})=L(A^{(1)})\cup L(A^{(2)})$. Let s be a new state symbol, $s\notin Q^{(1)}\cup Q^{(2)}$, let

$$Q=Q^{(1)}\cup Q^{(2)}\cup\{s\},$$

and let

$$F=\begin{cases} F^{(1)}\cup F^{(2)}, & \text{if } s^{(1)}\notin F^{(1)} \text{ and } s^{(2)}\notin F^{(2)},\\ F^{(1)}\cup F^{(2)}\cup\{s\}, & \text{otherwise.}\end{cases}$$

The function g is given as follows

$$g_q(a,u)=\begin{cases} g_q^{(i)}(a,u|_{Q^{(i)}}) & \text{if } q\in Q^{(i)} \text{ with } i\in\{1,2\},\\ g_{s^{(1)}}^{(1)}(a,u|_{Q^{(1)}})\vee g_{s^{(2)}}^{(2)}(a,u|_{Q^{(2)}}), & \text{if } q=s.\end{cases}$$

where $q\in Q$, $a\in\Sigma$, and $u\in\mathbb{B}^Q$.

THEOREM 8.2 $L(A^{(1)}\vee A^{(2)})=L(A^{(1)})\cup L(A^{(2)})$.

Proof By induction one verifies that

$$g_q(w,f)=\begin{cases} g_{s^{(1)}}^{(1)}(w,f^{(1)})\vee g_{s^{(2)}}^{(2)}(w,f^{(2)}) & \text{if } q=s,\\ g_q^{(i)}(w,f^{(i)}), & \text{if } q\in Q^{(i)} \text{ with } i\in\{1,2\},\end{cases}$$

for $q\in Q$ and $w\in\Sigma^*$. □

The construction of an AFA.

$$A=A^{(1)}\wedge A^{(2)}=(Q,\Sigma,s,F,g)$$

such that $L(A^{(1)}\wedge A^{(2)})=L(A^{(1)})\cap L(A^{(2)})$ is similar. With Q as above, one defines

$$F=\begin{cases} F^{(1)}\cup F^{(2)}, & \text{if } s^{(1)}\notin F^{(1)} \text{ or } s^{(2)}\notin F^{(2)},\\ F^{(1)}\cup F^{(2)}\cup\{s\}, & \text{otherwise.}\end{cases}$$

and

$$g_q(a,u)=\begin{cases} g_q^{(i)}(a,u|_{Q^{(i)}}), & \text{if } q\in Q^{(i)} \text{ with } i\in\{1,2\},\\ g_{s^{(1)}}^{(1)}(a,u)_{Q^{(1)}})\wedge g_{s^{(2)}}^{(2)}(a,u|_{Q^{(2)}}), & \text{if } q=s.\end{cases}$$

where $q \in Q$, $a \in \Sigma$, and $u \in \mathbb{B}^Q$. One then proves that $L(A)$ is indeed the intersection of the two languages.

THEOREM 8.3 $L(A^{(1)} \wedge A^{(2)}) = L(A^{(1)}) \cap L(A^{(2)})$.

9. CONCATENATION AND STAR ON AFA

The operations of concatenation and star operation on AFA are more difficult than the Boolean operations. We found only a partial solution so far.

Let Q be a finite, non-empty set. One defines a partial order on $\mathbb{B}^Q$ by

$$u \leqq v \Leftrightarrow \forall q \in Q : u_q \leqq v_q$$

with u, $v \in \mathbb{B}^Q$ and $0 < 1$. If $A = (Q, \Sigma, s, F, g)$ is an AFA such that, for every $q \in Q$ and $a \in \Sigma, g_q(a)$ is either constant or a Boolean function with $\vee$-operators only then A actually behaves like an NFA. In this case we say that *A is an NFA in AFA representation.* When there is no risk of confusion we would also just say that A is an NFA.

LEMMA 9.1 *Let $A = (Q, \Sigma, s, F, g)$ be an NFA in AFA representation. Then for every $u, v \in \mathbb{B}^Q$ and every $a \in u$ the following statements hold true:*

1) $u \leqq v$ implies $g(a)(u) \leqq g(a)(v)$.

2) $g(a)(u \vee v) = g(a)(u) \vee g(a)(v)$.

Proof The properties are immediate consequences of the definition. □

THEOREM 9.2 *Let $A^{(1)} = (Q^{(1)}, \Sigma, s^{(1)}, F^{(1)}, g^{(1)})$ be an NFA in AFA representation, and let $A^{(2)} = (Q^{(2)}, \Sigma, s^{(2)}, F^{(2)}, g^{(2)})$ be an arbitrary AFA with $Q^{(1)}$ and $Q^{(2)}$ disjoint. Consider the AFA*

$$A^{(1)} \cdot A^{(2)} = (Q^{(1)} \cup Q^{(2)}, \Sigma, s^{(1)}, F^{(2)}, g)$$

where

$$g_q(a) = \begin{cases} g_q^{(1)}(a), & \text{if } q \in Q^{(1)} \text{ and } q \notin F^{(1)}, \\ g_q^{(1)}(a) \vee g_{s^{(2)}}^{(2)}(a), & \text{if } q \in Q^{(1)} \text{ and } q \in F^{(1)}, \\ g_q^{(2)}(a), & \text{if } q \in Q^{(2)} \end{cases}$$

for $q \in Q^{(1)} \cup Q^{(2)}$ and $a \in \Sigma$. Then $L(A^{(1)} \cdot A^{(2)}) = L(A^{(1)})L(A^{(2)})$. Moreover, if $A^{(1)}$ and $A^{(2)}$ have m_1 and m_2 states, respectively, then $A^{(1)} \cdot A^{(2)}$ has $m_1 + m_2$ states.

Proof Let $w \in L(A^{(1)})L(A^{(2)})$. Then $w = xy$ for some $x \in L(A^{(1)})$ and $y \in L(A^{(2)})$. Then $g_{s^{(2)}}^{(2)}(y) = 1$ and, therefore, $g_q(y) = 1$ for all $q \in F^{(1)}$. Hence $f^{(1)} \leqq g(y)|_{Q^{(1)}}$ and, using Lemma 9.1, $g^{(1)}(x)(f^{(1)}) \leqq g^{(1)}(x)(g(y)|_{Q^{(1)}})$. Since $g_{s^{(1)}}^{(1)}(x)(f^{(1)}) = 1$ one has $g_{s^{(1)}}^{(1)}(x)g(y)|_{Q^{(1)}} = 1$, that is, $w = xy \in L(A^{(1)} \cdot A^{(2)})$.

Let $w \notin L(A^{(1)})L(A^{(2)})$. We show that $w \notin L(A^{(1)} \cdot A^{(2)})$. Let $y_1, y_2, \ldots, y_n$ be those words which are suffixes of w and are contained in $L(A^{(2)})$. If $n=0$, that is, if w has no suffix y which is contained in $L(A^{(2)})$ then $g_q(w)=0$ for all $q \in Q^{(1)}$, hence $w \notin L(A^{(1)} \cdot A^{(2)})$. Therefore, suppose that $n>0$. Then there are distinct words $x_1, x_2, \ldots, x_n \in \Sigma^*$ such that $x_i y_i = w$ for $i=1,2,\ldots,n$. Moreover, one has $x_i \notin L(A^{(1)})$ for these words, that is, $g^{(1)}_{s^{(1)}}(x_i)(f^{(1)})=0$. Using the fact that $A^{(1)}$ is an NFA in AFA representation and by Lemma 9.1 one obtains $g_s(w)=\bigvee_{i=1}^{n} g^{(1)}_{s^{(1)}}(x_i)(f^{(1)})=0$. This implies $w \notin L(A^{(1)} \cdot A^{(2)})$. □

THEOREM 9.3 *Let $A^{(1)}$ and $A^{(2)}$ be two AFA with m_1 and m_2 states, respectively. Then there exists an s-AFA A with at most $2^{m_1}+m_2+1$ states which accepts the language $L(A^{(1)})L(A^{(2)})$.*

Proof Using the results of Section 5, one transforms the AFA $A^{(1)}$ into an NFA $A^{(1)'}$ with $2^{m_1}+1$ states which accepts the language $L(A^{(1)})$. Using the construction of Section 6, one obtains an equivalent NFA in AFA representation $A^{(1)''}$ with $2^{m_1}+1$ states. We now apply Theorem 9.2 to complete the proof. □

In Theorem 9.3 we show that $2^{m_1}+m_2+1$ states suffice for an AFA to accept the concatenation of two languages accepted by AFA with m_1 and m_2 states, respectively. We conjecture that this number of states is actually necessary in the worst case, but have no proof.

A modification of the construction used in Theorem 9.2 can be used to obtain an AFA which accepts the star closure of an AFA accepted language.

10. CONCLUDING REMARKS

As far as language recognition is concerned, AFA are exactly as powerful as DFA [2]. However, their mode of accepting words is quite different from that considered with DFA. This becomes strikingly clear from our result that an AFA needs at most one final state; moreover, it can do without any final states if the empty word is not to be accepted.

Given this difference it is important to derive at least the basic constructions for AFA. In this paper we provide direct transformations between AFA, NFA, and DFA and we exhibit constructions for union, intersection, and negation. Further results concern minimization, concatenation, and star operation.

Additional questions will need to be asked about AFA. On the one hand, a structure theory would need to be developed to turn AFA into a tool as easily usable as are NFA. On the other hand, a representation theory which parallels the theory of regular expressions and of linear equations over certain semirings has to be developed in order to make AFA tractable by algebraic tools. The latter problem is addressed in a paper in preparation.

References

[1] A. K. Chandra, D. C. Kozen and L. J. Stockmeyer, *Alternation*. Res. Rept. R7489, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1978.

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

[3] J. H. Chang, O. H. Ibarra and B. Ravikumar, Some observations concerning alternating turing machines using small space, *Inform. Process. Lett.* **25** (1987), 1–9.
[4] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, 1979.
[5] H. Hromkovič, Alternating multicounter machines with constant number of reversals, *Information Processing Letters* **21** (1985), 7–9.
[6] J. Hromkovič, On the power of alternation in Automata Theory, *J. of Comp. and Syst. Sci.* **31** (1985), 28–39.
[7] K. Inoue, I. Takanami and H. Tanaguchi, Two-dimensional alternating turing machines, *Proc. 14th Ann. ACM Symp. On Theory of Computing*, New York, ACM, 1982, 37–46.
[8] K. Inoue, I. Takanami and H. Tanaguchi, A note on alternating on-line turing machines, *Inform. Process. Lett.* **15** (1982), 164–168.
[9] K. N. King, *Restricted Forms of Alternation*, Ph.D. Dissertation, University of California, Berkeley, 1980.
[10] P. A. Lindsay, Alternation and ω-type turing acceptors, *Theoret. Comput. Sci.* **43** (1986), 107–115.
[11] R. E. Ladner, R. J. Lipton and L. J. Stockmeyer, Alternating pushdown automata, *Proc. 19th IEEE Symp. on Foundations of Computer Science*, IEEE, 1978, 92–106.
[12] W. J. Paul, E. J. Prauss and R. Reischuck: On Alternation, *Acta Inform.* **14** (1980), 243–255.
[13] W. L. Ruzzo, Tree-size bounded alternation, *J. Comput. System Sci.* **21** (1980), 218–235.
[14] I. H. Sudborough: Efficient algorithms for path system problems and applications to alternating and time-space complexity classes, *Proc 21st IEEE Ann. Symp. On Foundations of Computer Science*, IEEE, 1980, 62–72.
[15] K. Wagner and G. Wechsung: *Computational Complexity*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.
[16] D. Wood, *Theory of Computation*, Harper & Row, New York, 1987 (2nd printing, Wiley, 1988).