Journal of Automata, Languages and Combinatorics **14** (2009) w, x–y

A TECHNIQUE FOR PROVING LOWER BOUNDS ON THE SIZE OF SWEEPING AUTOMATA

Hing Leung
Department of Computer Science, New Mexico State University
Las Cruces, NM 88003, USA
e-mail: hleung@cs.nmsu.edu

ABSTRACT

A sweeping automaton is a two-way deterministic finite automaton which makes turns only at the endmarkers. Sipser [12] has proved that one-way nondeterministic finite automata can be exponentially more succinct in sizes than sweeping automata. In this paper, we propose a technique based on the work in [6] for establishing lower bounds on the size of sweeping automata. We show that Sipser's technique is a special case of our method. In addition, we prove two lower bound results with the new technique.

Keywords: Deterministic finite automata, nondeterministic finite automata, two-way finite automata, sweeping automata, descriptional complexity

1. Introduction

The simplest machine model for denoting regular languages is the one-way deterministic finite automaton (DFA). It is well known that the use of nondeterminism and two-way movements of the tape head would not change the class of languages denoted.

Tradeoffs in the succinctness of different machine models for denoting the same languages were studied in a number of research papers. Meyer and Fischer [7] showed that for each positive n, there is an n-state nondeterministic finite automaton (NFA) such that the corresponding smallest equivalent DFAs have 2^n states. The same result was also obtained by Moore [9] using a different family of NFAs.

Sakoda and Sipser [10] raised an open question regarding the tradeoff in the succinctness between two-way nondeterministic finite automata (2NFA) and two-way deterministic finite automata (2DFA). Specifically, they asked whether there exists a polynomial p such that for every n-state 2NFA there is an equivalent $p(n)$-state 2DFA.

A partial negative answer has been provided by Sipser in [12]. Sweeping automaton is introduced as a restricted model of 2DFA which can reverse the direction of the reading head only at the endmarkers on the two sides of the input. In Section 2, we give the formal definitions of 2DFA and sweeping automaton. We say that a sweeping automaton is degenerate if the automaton has no left-moving transitions. Thus, a degenerate sweeping automaton is limited to make only one sweep on the input from left to right. That is, a degenerate sweeping automaton is the same as an

incomplete one-way DFA except that the input is delimited by two endmarkers for a sweeping automaton. In this paper, we allow a sweeping automaton to be degenerate. Sipser [12] showed that for each positive integer n, there is a language B_n such that B_n is accepted by an n-state NFA whereas it is not accepted by any sweeping automaton with fewer than $2^n - 1$ states. As any n-state NFA can be simulated by an incomplete one-way DFA with at most $2^n - 1$ states, Sipser's result achieves the largest tradeoff possible in the number of states between NFA and sweeping automata. It should be noted that B_n is over an alphabet of size 2^{n^2}. It is argued [12] that with some clever encoding of the alphabet into a binary alphabet, the NFA after encoding has at most $O(n)$ states.

In [1] and [8], Sipser's technique for proving lower bounds on the size of sweeping automata was used in showing that 2DFAs can be exponentially more succinct than sweeping automata.

In [6], Leung showed the same tradeoff result as Sipser's between NFA and sweeping automata for the regular language $(0 + 0(1^*0)^{n-1})^*$ which is over a binary alphabet. Leung used a different technique than Sipser's.

Recently, the tradeoff questions about different variant models of sweeping automata have received renewed interest ([2], [3], [4]).

In this paper, we consider again the result obtained in [6]. We identify the crucial properties that are needed in the technique for establishing lower bounds on the size of sweeping automata. In Section 3, we formulate the proof technique in Theorem 1. In Section 4, we show that Sipser's technique is a special case of our technique given in Theorem 1. On the other hand, it is clear that Sipser's technique, which relies on the construction of a string of exponential length, is not equivalent to our technique which does not require the existence of such a long string. Also, we illustrate with examples that Theorem 1 can be easily applied to prove lower bound results on the size of sweeping automata for two new languages.

In Section 5, we propose an open problem about a possible generalization of Theorem 1.

2. Definitions and Notation

We say that $y \in \Sigma^+$ is *live* with respect to a language L iff $\exists\ x, z \in \Sigma^*$ such that $xyz \in L$; otherwise y is *dead* with respect to L. Sometimes, we may simply say that a string is live (or, dead) without mentioning explicitly the underlying language.

2.1. 2DFA

A 2DFA is a 7-tuple $(Q, \Sigma, \vdash, \dashv, \delta, q_1, F)$, where $Q = \{q_1, q_2, \ldots, q_k\}$ is the set of states, Σ is the alphabet set, $\vdash \notin \Sigma$ and $\dashv \notin \Sigma$ are left and right endmarkers delimiting the input string, q_1 is the starting state and F is the set of accepting states. The transition function δ is a *partial* function from $Q \times (\Sigma \cup \{\vdash, \dashv\})$ to $Q \times \{L, R\}$. An input string $w = a_1 a_2 \ldots a_n$, where $a_i \in \Sigma$ for $1 \leq i \leq n$, is presented to the 2DFA as $\vdash a_1 a_2 \ldots a_n \dashv$. The 2DFA is started in state q_1 on the symbol a_1. If w is the empty string ϵ, then the 2DFA is started in state q_1 on the right endmarker $\dashv$. The input

string w is accepted if, from the initial configuration in which the automaton is in state q_1 while reading a_1 (or $\dashv$ if $w = \epsilon$), the 2DFA enters into a configuration with the state in F while reading $\dashv$ after a sequence of moves. The sequence of moves may be empty if $w = \epsilon$. More accurately, if the sequence of moves is not empty, the 2DFA must signal acceptance by entering a state in F when it makes a right move on the symbol a_n (before detecting that it has reached the $\dashv$ symbol).

Let $q_i, q_j \in Q$ and $a \in \Sigma \cup \{\vdash, \dashv\}$. When $\delta(q_i, a) = (q_j, L)$, the meaning is that the 2DFA, when reading symbol a while at state q_i, would move the tape head to the left and change the state to q_j. Similarly, the meaning of $\delta(q_i, a) = (q_j, R)$ is that the 2DFA, when reading symbol a while at state q_i, would move the tape head to the right and change the state to q_j. Another possibility is that $\delta(q_i, a)$ could be undefined.

Given a definition of δ, we extend its definition to a (partial) function from $Q \times \Sigma^+$ to $Q \times \{L, R\}$ as follows. Let $w \in \Sigma^+$. We write $\delta(q_i, w) = (q_j, L)$ to denote that the 2DFA, when started at the leftmost symbol of w at state q_i, eventually leaves w moving to its left while entering state q_j; we write $\delta(q_i, w) = (q_j, R)$ to denote that the 2DFA, when started at the leftmost symbol of w at state q_i, eventually leaves w moving to its right while entering state q_j. It is possible that $\delta(q_i, w)$ may be undefined when the 2DFA hangs or loops within w.

Similarly, with respect to the original given definition of δ, we extend the definition of δ to a (partial) function from $\Sigma^+ \times Q$ to $Q \times \{L, R\}$ as follows. We write $\delta(w, q_i) = (q_j, L)$ to denote that the 2DFA, when started at the rightmost symbol of w at state q_i, eventually leaves w moving to its left while entering state q_j; we write $\delta(w, q_i) = (q_j, R)$ to denote that the 2DFA, when started at the rightmost symbol of w at state q_i, eventually leaves w moving to its right while entering state q_j. Again, it is possible that $\delta(w, q_i)$ may be undefined.

Let $a \in \Sigma$. Using the previous notation, $\delta(q_i, a) = \delta(a, q_i)$.

2.2. Sweeping Automata

A sweeping automaton is a 2DFA which makes turns only at the endmarkers. To consider a 2DFA as a sweeping automaton, we require that the set of states is partitioned into left-going and right-going states. In other words, no state can be both left-going and right-going; that is, it cannot be the case that there exists a state q, and two non-endmarker symbols $a, a' \in \Sigma$ such that $\delta(q, a) = (q', L)$ and $\delta(q, a') = (q'', R)$; however, a left-going (right-going) state has to make a right-going (left-going) move at the left (right) endmarker. Also, a left-going (right-going) move cannot result in a state that is right-going (left-going). Initially, a sweeping automaton is required to start in a right-going state. For the special case when the input string is an empty string, any 2DFA can only sweep from one endmarker to another endmarker, and thus behaves like a sweeping automaton.

Let $w \in \Sigma^+$. For $Q' \subseteq Q$, let $\overrightarrow{\delta}(Q', w)$ denote $\{q_j \mid q_i \in Q',\ \delta(q_i, w) = (q_j, R)\}$ and $\overleftarrow{\delta}(w, Q')$ denote $\{q_j \mid q_i \in Q',\ \delta(w, q_i) = (q_j, L)\}$. We write $\overrightarrow{\delta}(w)$ to denote $\overrightarrow{\delta}(Q, w)$ and $\overleftarrow{\delta}(w)$ to denote $\overleftarrow{\delta}(w, Q)$. Even if some of the k components of $\overrightarrow{\delta}(w)$ are not

defined, we still consider $\overrightarrow{\delta}(w)$ to be defined. That is, $\overrightarrow{\delta}(w) = \overrightarrow{\delta}(w')$ if corresponding entries of the k-tuples are either both undefined or both defined and equal. The same can be said about $\overleftarrow{\delta}(w)$. Next, we write $\delta(w)$ to denote $(\overrightarrow{\delta}(w), \overleftarrow{\delta}(w))$. Thus, the sweeping automaton will behave similarly (with the same behavior when viewed externally) on strings w and w' if $\delta(w) = \delta(w')$.

We write $\eta(w) = (|\overrightarrow{\delta}(w)|, |\overleftarrow{\delta}(w)|)$. We define a partial ordering on ordered pairs of natural numbers such that $(p, q) \leq (i, j)$ iff $p \leq i$ and $q \leq j$. It is easy to see that $\eta(ww') \leq \eta(w)$ and $\eta(w'w) \leq \eta(w)$ for $w, w' \in \Sigma^+$.

We say that a live string w is *minimal* with respect to a sweeping automaton if, for all live strings w', $\eta(w') \leq \eta(w)$ implies $\eta(w') = \eta(w)$. Note that by definition a dead string cannot be minimal. It is easy to see that minimal strings exist when the sweeping automaton accepts a nonempty string.

We observe that the statement $\forall x, y \in \Sigma^+$, $\overrightarrow{\delta}(Q', xy) = \overrightarrow{\delta}(\overrightarrow{\delta}(Q', x), y)$ is true for sweeping automata, but not for 2DFA.

3. Main Result

In this section, we are going to prove Theorem 1 which is the main result of this paper.

Theorem 1 *Let L be a language and let x be a string such that for all live strings u and v, uxv is live. Furthermore, let α_i and β_i, for $1 \leq i \leq m$, be strings such that*

- *$\alpha_i\beta_j$ is live if $i + j \geq m + 1$,*
- *$\alpha_i\beta_j$ is not live if $i + j = m$.*

Then any sweeping automaton accepting L has at least m states.

Consider a sweeping automaton $B = (Q, \Sigma, \vdash, \dashv, \delta, q_1, F)$ accepting L where $Q = \{q_1, q_2, \ldots, q_k\}$. We want to show that $k \geq m$.

Let z be a minimal string with respect to B and let $g = xzx$. We see that $g = ((\epsilon xz)x\epsilon)$ is live by the the property of x and the fact that the empty string ϵ is live since L is nonempty. As g is live and has a minimal string z as its substring, we deduce that g is minimal. Also, g possesses the same property that x has: Let u, v be two live strings. Consider $ugv = u(xzx)v = ((uxz)xv)$ which is live.

For any string y, it is true that gyg is live iff y is live. If y is not live, by the definition of a live string, gyg cannot be live. Suppose y is live. Then the word $gyg = (xzx)y(xzx) = ((\epsilon x((zxy)xz))x\epsilon)$ is again live. Since ϵ is live, $gg = g\epsilon g$ is live and hence minimal as g is minimal.

Lemma 1 *For $1 \leq i, j \leq m$, we have*

- *$g\alpha_i\beta_j g$ is minimal if $i + j \geq m + 1$,*
- *$g\alpha_i\beta_j g$ is not minimal if $i + j = m$.*

Proof. Suppose $i+j \geq m+1$. By the properties of α_i's and β_j's, $\alpha_i\beta_j$ is live. Recall that gyg is live iff y is live. Thus, $g\alpha_i\beta_j g$ is live and minimal since g is minimal. When $i+j=m$, we have $\alpha_i\beta_j$ is not live; hence $g\alpha_i\beta_j g$ is not live and not minimal. □

Lemma 2 *$\overrightarrow{\delta}(gg) = \overrightarrow{\delta}(g)$ and $\overleftarrow{\delta}(gg) = \overleftarrow{\delta}(g)$*

Proof. The lemma follows from the facts that g and gg are minimal, and that

$$\overrightarrow{\delta}(gg) = \overrightarrow{\delta}(Q, gg) \subseteq \overrightarrow{\delta}(Q, g) = \overrightarrow{\delta}(g) \text{ and}$$
$$\overleftarrow{\delta}(gg) = \overleftarrow{\delta}(gg, Q) \subseteq \overleftarrow{\delta}(g, Q) = \overleftarrow{\delta}(g).$$ □

Let $\overrightarrow{\delta}(g) = \{q_{r_1}, q_{r_2}, \ldots, q_{r_s}\}$ and $\overleftarrow{\delta}(g) = \{q_{l_1}, q_{l_2}, \ldots, q_{l_t}\}$, where $\eta(g) = (s,t)$. Note that s cannot be 0. Otherwise g cannot be live, since a string is only accepted while reading the right endmarker. On the other hand, it is possible that t may be 0.

Define a matrix $\overrightarrow{D}$ over the field of integers mod 2 with rows indexed by

$$\{q \mid q \in \overrightarrow{\delta}(g\alpha_i),\ 1 \leq i \leq m\}$$

and columns indexed by

$$\{(\beta_j g, q_{r_h}) \mid 1 \leq h \leq s,\ 1 \leq j \leq m\}$$

such that $\overrightarrow{D}[q, (\beta_j g, q_{r_h})] = 1$ if $\delta(q, \beta_j g) = (q_{r_h}, R)$, and 0 otherwise.

Define a matrix $\overleftarrow{D}$ over the field of integers mod 2 with rows indexed by

$$\{(q_{l_k}, g\alpha_i) \mid 1 \leq k \leq t,\ 1 \leq i \leq m\}$$

and columns indexed by

$$\{q \mid q \in \overleftarrow{\delta}(\beta_j g),\ 1 \leq j \leq m\}$$

such that $\overleftarrow{D}[(q_{l_k}, g\alpha_i), q] = 1$ if $\delta(g\alpha_i, q) = (q_{l_k}, L)$, and 0 otherwise.

Obtain from $\overrightarrow{D}$ by elementary row operations a matrix $\overrightarrow{E}$ over the field of integers mod 2 with rows indexed by $\{g\alpha_i \mid 1 \leq i \leq m\}$ such that the row in $\overrightarrow{E}$ indexed by $g\alpha_i$ is obtained by adding those rows of $\overrightarrow{D}$ indexed by states in $\overrightarrow{\delta}(g\alpha_i)$.

Obtain from $\overleftarrow{D}$ by elementary column operations a matrix $\overleftarrow{E}$ over the field of integers mod 2 with columns indexed by $\{\beta_j g \mid 1 \leq j \leq m\}$ such that the column in $\overleftarrow{E}$ indexed by $\beta_j g$ is obtained by adding those columns of $\overleftarrow{D}$ indexed by states in $\overleftarrow{\delta}(\beta_j g)$.

Lemma 3 *Let $P(i,j)$ denote the statement that*

$$\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_h})] = 1 \text{ and } \overleftarrow{E}[(q_{l_k}, g\alpha_i), \beta_j g] = 1$$

for all $1 \leq h \leq s$ and $1 \leq k \leq t$. We have

- *if $i+j \geq m+1$, then $P(i,j)$ is true,*
- *if $i+j=m$, then $P(i,j)$ is false.*

Proof. The proof for the case when $t = 0$ is similar to that for the case when $t > 0$ but only simpler with $P(i, j)$ denoting $\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_h})] = 1$ for all $1 \leq h \leq s$. In the following proof, we suppose $t > 0$.

Suppose $i + j \geq m + 1$. By Lemma 1, $g\alpha_i\beta_j g$ is minimal. Since g is minimal, we deduce that $g\alpha_i$ is also minimal. Thus, $s = |\overrightarrow{\delta}(g)| = |\overrightarrow{\delta}(g\alpha_i)| = |\overrightarrow{\delta}(g\alpha_i\beta_j g)|$. Observe that $\overrightarrow{\delta}(\overrightarrow{\delta}(g\alpha_i), \beta_j g) = \overrightarrow{\delta}(g\alpha_i\beta_j g) = \overrightarrow{\delta}(g) = \{q_{r_1}, \ldots, q_{r_s}\}$. Let $1 \leq h \leq s$. Since $|\overrightarrow{\delta}(g\alpha_i)| = s$, there exists a *unique* $q' \in \overrightarrow{\delta}(g\alpha_i)$ such that

$$\delta(q', \beta_j g) = (q_{r_h}, R).$$

That is, for $q \in \overrightarrow{\delta}(g\alpha_i)$, $\overrightarrow{D}[q, (\beta_j g, q_{r_h})] = 1$ iff $q = q'$. Therefore,

$$\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_h})] = \sum\{\overrightarrow{D}[q, (\beta_j g, q_{r_h})] \mid q \in \overrightarrow{\delta}(Q, g\alpha_i)\} = 1.$$

Similarly, we can show that $\overleftarrow{E}[(q_{l_k}, g\alpha_i), \beta_j g] = 1$ for $1 \leq k \leq t$.

Suppose $i+j = m$. Suppose the contrary that $P(i, j)$ is true. By Lemma 1 and the fact that g is minimal, it suffices to show that $g\alpha_i\beta_j g$ is live; hence, a contradiction. By Lemma 2, we have

$$\begin{aligned}
\overrightarrow{\delta}(\{q_{r_1}, \ldots, q_{r_s}\}, g\alpha_i\beta_j g) &= \overrightarrow{\delta}(\overrightarrow{\delta}(\{q_{r_1}, \ldots, q_{r_s}\}, g), \alpha_i\beta_j g) \\
&= \overrightarrow{\delta}(\overrightarrow{\delta}(\overrightarrow{\delta}(g), g), \alpha_i\beta_j g) \\
&= \overrightarrow{\delta}(\overrightarrow{\delta}(gg), \alpha_i\beta_j g) \\
&= \overrightarrow{\delta}(\overrightarrow{\delta}(g), \alpha_i\beta_j g) \\
&= \overrightarrow{\delta}(g\alpha_i\beta_j g)
\end{aligned}$$

which is $\{q_{r_1}, \ldots, q_{r_s}\}$ by the assumption that $\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_h})] = 1$ for $1 \leq h \leq s$. Thus, there exists x such that $\delta(q_{r_h}, (g\alpha_i\beta_j g)^x) = (q_{r_h}, R)$ for all $1 \leq h \leq s$. Similarly, there exists y such that $\delta((g\alpha_i\beta_j g)^y, q_{l_k}) = (q_{l_k}, L)$ for all $1 \leq k \leq t$. Therefore, $\delta(g(g\alpha_i\beta_j g)^{xy} g) = \delta(gg)$. Since gg is live, $g(g\alpha_i\beta_j g)^{xy} g$ is live and hence $g\alpha_i\beta_j g$ is live. □

Lemma 4 *Let F be an $m \times m$ matrix over the field of integers mod 2 such that if $i + j \geq m + 1$ then $F[i, j] = 1$. Then* $\text{rank}(F) \geq 1 + |\{(i, j) \mid i + j = m,\ F[i, j] = 0\}|$.

Proof. See Figure 1 for a picture of F when $m = 15$. The entries labelled ? and $*$ consist of 0's and 1's. In the Appendix, we show that the matrix F has the full rank of m if all positions labelled ? hold the values 0's. As a consequence, the rows that have a zero entry in the position labelled ?, together with the row of all 1's, are linearly independent. We are done since the positions labelled ? are entries (i, j) such that $i + j = m$. □

Lemma 5 *Suppose $t > 0$. Then* $\text{rank}(\overrightarrow{E}) + \text{rank}(\overleftarrow{E}) \geq m + 1$.

$$\begin{bmatrix}
* & * & * & * & * & * & * & * & * & * & * & * & ? & 1 \\
* & * & * & * & * & * & * & * & * & * & * & ? & 1 & 1 \\
* & * & * & * & * & * & * & * & * & * & ? & 1 & 1 & 1 \\
* & * & * & * & * & * & * & * & * & ? & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & * & ? & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & ? & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & ? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
? & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Figure 1: Structure of F when m is 15

Proof. Recall that each column of $\overrightarrow{E}$ is indexed by $(\beta_j g, q_{r_h})$, where $1 \leq j \leq m$ and $1 \leq h \leq s$. By selecting from $\overrightarrow{E}$ one column for each group of s columns indexed by $\{(\beta_j g, q_{r_h}) \mid 1 \leq h \leq s\}$, we obtain a matrix $\overrightarrow{F}$ over the field of integers mod 2 with rows and columns indexed by i and j for $1 \leq i, j \leq m$. The selections are done as follows. If $j = m$, then we select any arbitrary column from the s columns of $\overrightarrow{E}$ indexed by $\{(\beta_m g, q_{r_h}) \mid 1 \leq h \leq s\}$. Otherwise, suppose $j < m$. Let $i = m - j$. There are two cases. The first case is when $\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_h})] = 1$ for all $1 \leq h \leq s$. We select again any arbitrary column from the s columns of $\overrightarrow{E}$ indexed by $\{(\beta_j g, q_{r_h}) \mid 1 \leq h \leq s\}$. The second case is when there exists $1 \leq h' \leq s$ such that $\overrightarrow{E}[g\alpha_i, (\beta_j g, q_{r_{h'}})] = 0$. We select the column indexed by $(\beta_j g, q_{r_{h'}})$ from $\overrightarrow{E}$ as the column indexed by j for $\overrightarrow{F}$. It follows from the properties of $\overrightarrow{E}$ as stated in Lemma 3 that if $i + j \geq m + 1$ then $\overrightarrow{F}[i, j] = 1$. That is, $\overrightarrow{F}$ has the structure of F as given in Lemma 4. Similarly, we can construct $\overleftarrow{F}$ from $\overleftarrow{E}$ which also has the structure of F. Let $\mathcal{P} = \{(i, j) \mid i + j = m, 1 \leq i, j \leq m\}$. By Lemma 3, if $(i, j) \in \mathcal{P}$, then either one or both of $\overrightarrow{F}[i, j]$ and $\overleftarrow{F}[i, j]$ is 0. Thus, by Lemma 4,

$$\begin{aligned}
\operatorname{rank}(\overrightarrow{F}) + \operatorname{rank}(\overleftarrow{F}) &\geq 1 + |\{(i,j) \in \mathcal{P} \mid \overrightarrow{F}[i,j] = 0\}| + \\
&\quad 1 + |\{(i,j) \in \mathcal{P} \mid \overleftarrow{F}[i,j] = 0\}| \\
&\geq 2 + |\{(i,j) \in \mathcal{P} \mid \overrightarrow{F}[i,j] = 0 \text{ or } \overleftarrow{F}[i,j] = 0\}| \\
&= 2 + |\mathcal{P}| \\
&= 2 + (m-1) \\
&= m + 1.
\end{aligned}$$

Therefore, $\operatorname{rank}(\overrightarrow{E}) + \operatorname{rank}(\overleftarrow{E}) \geq m+1$ since $\overrightarrow{F}$ and $\overleftarrow{F}$ are obtained from $\overrightarrow{E}$ and $\overleftarrow{E}$

respectively by elementary operations. □

Lemma 6 *Suppose $t = 0$. Then* $\text{rank}(\overrightarrow{E}) \geq m$.

Proof. The proof is similar to that of the previous lemma. The difference is that we apply Lemma 4 only once to give $\text{rank}(\overrightarrow{E}) \geq \text{rank}(\overrightarrow{F}) \geq 1 + |\mathcal{P}| = 1 + (m-1) = m$. □

Finally, we are ready to prove Theorem 1.

Proof. **(Theorem 1)** There are two cases to consider. The first case is when $t > 0$. Since $\overrightarrow{E}$ is derived from $\overrightarrow{D}$ by elementary row operations, we have

$$\text{rank}(\overrightarrow{D}) \geq \text{rank}(\overrightarrow{E}).$$

Since the rows of $\overrightarrow{D}$ are indexed by states in $\bigcup_{1 \leq i \leq m} \overrightarrow{\delta}(g\alpha_i)$, we have

$$|\bigcup_{1 \leq i \leq m} \overrightarrow{\delta}(g\alpha_i)| \geq \text{rank}(\overrightarrow{D}) \geq \text{rank}(\overrightarrow{E}).$$

Similarly, $|\bigcup_{1 \leq j \leq m} \overleftarrow{\delta}(\beta_j g)| \geq \text{rank}(\overleftarrow{D}) \geq \text{rank}(\overleftarrow{E})$. As the left-going and right-going states are disjoint, the number of states is at least $\text{rank}(\overrightarrow{E}) + \text{rank}(\overleftarrow{E}) \geq m + 1$.

The second case is when $t = 0$. A simpler analysis shows that the number of states is at least $|\bigcup_{1 \leq i \leq m} \overrightarrow{\delta}(g\alpha_i)| \geq \text{rank}(\overrightarrow{E}) \geq m$ by Lemma 6. □

In Section 2, we required that, when a sweeping automaton wants to signal acceptance, it must enter a final state while moving right on the last symbol a_n before detecting the right endmarker $\dashv$. We observe that the proof of the main result does not rely on this specific requirement for acceptance. The result is still valid if we relax the acceptance criterion allowing the sweeping automaton to signal acceptance after it has detected the right endmarker.

4. Applications of Theorem 1

Sipser [12] introduced a technique, which was also used by Berman [1] and Micali [8] for proving lower bounds on the size of sweeping automata. Using Theorem 1, we give an alternate proof of Sipser's result ([1], [8]) as presented in the next lemma.

Lemma 7 (Sipser) *Let L be a language such that*

1. *If $w \in L$, then all substrings of w are in L.*
2. *There exists a string x such that for all $u, v \in L$, $uxv \in L$.*
3. *There exists a string $d \notin L$ and $|d| = 2^n$ such that the removal of any non-empty substring from d results in a string that is in L.*

Then any sweeping automaton accepting L has at least $2^n - 1$ states.

Proof. From condition 1, we know that all live strings are in the language L. Since all strings in L are live, the set of live strings is the same as the set of strings in L. We can therefore rewrite condition 2 to state that there exists x such that for all live strings u and v, uxv is live. Let $d = a_1a_2 \dots a_{2^n}$ where a_i is a letter. For $1 \leq i \leq 2^n - 1$, we define $\alpha_i = a_1a_2 \dots a_{2^n-i}$ and $\beta_i = a_{i+2}a_{i+3} \dots a_{2^n}$. Thus, $\alpha_i\beta_j = (a_1a_2 \dots a_{2^n-i})(a_{j+2}a_{j+3} \dots a_{2^n})$. Suppose $i + j \geq 2^n$. We deduce that $2^n - i + 1 \leq j + 1$. Thus, $\alpha_i\beta_j$ is live in L as it is obtained by removing the non-empty substring $a_{2^n-i+1} \dots a_{j+1}$ from d. When $i + j = 2^n - 1$ holds, then we have

$$\begin{aligned}\alpha_i\beta_j &= (a_1a_2 \dots a_{2^n-i})(a_{j+2}a_{j+3} \dots a_{2^n}) \\ &= (a_1a_2 \dots a_{2^n-i})(a_{2^n-i+1}a_{2^n-i+2} \dots a_{2^n}) \\ &= d\end{aligned}$$

which is not live as $d \notin L$. Therefore, by applying Theorem 1, any sweeping automaton for L requires at least $2^n - 1$ states. □

In [6], the next lemma is proved. Here we give a sketch of the proof using Theorem 1. For a detailed proof, see [6].

Lemma 8 (Leung) *Let $L = (0 + 0(1^*0)^{n-1})^*$. Any sweeping automaton for L has at least $2^n - 1$ states.*

Proof. (Sketch) Let $x = 0^n$. One can show that for all live strings u, v, uxv is live. In [5], it is shown that there exists w_P for each nonempty set $P \subseteq \{1, 2, \dots, n\}$ such that for any nonempty subsets $P, Q \subseteq \{1, 2, \dots, n\}$, w_Pw_Q is live iff $P \cap Q \neq \emptyset$. For each nonempty set $P \subseteq \{1, 2, \dots, n\}$, we can represent it as a positive integer i whose equivalent n-bit binary representation $b_nb_{n-1} \dots b_1$ is defined by $b_i = 1$ if $i \in P$, and $b_i = 0$ otherwise. We denote w_P equivalently as w_i where i is the positive integer that represents P.

Consider two positive integers i and j such that $i + j = 2^n - 1$. As the n-bit binary representations of i and j are complementary, the corresponding two subsets P and Q are disjoint. Therefore, w_iw_j is not live as w_Pw_Q is not live when $P \cap Q = \emptyset$. On the other hand, suppose $i + j \geq 2^n$. There must exist $1 \leq k \leq n$ such that the k-th bits of the binary representations i and j are both one. Therefore, $k \in P \cap Q \neq \emptyset$. Hence, w_iw_j is live as w_Pw_Q is live. By defining $\alpha_i = \beta_i = w_i$ and applying Theorem 1, we conclude that any sweeping automaton for L has at least $2^n - 1$ states. □

In general, it is easy to construct a language that requires an exponential number of states for a sweeping automaton to denote the language. We illustrate the steps using two examples.

Let $L_1 = \{xy \mid x, y \in \{0, 1\}^n, i + j \geq 2^n$, where x and y are the binary representations of the integers i and j $\}$. A sweeping automaton can denote L_1 in $O(n^2)$ states by performing the addition logic in n passes over the input to inspect the two corresponding positions of each of the n-bits in x and y. Let $m = 2^n - 1$. It is clear that by identifying the x's with α's and the y's as β's, the second condition for Theorem 1 is satisfied. Let $L_2 = \{w \mid w$ is a substring of some string in $L_1\}$. Note that $\epsilon \in L_2$

and the set of live strings of L_2 is exactly L_2. Let $\$ \notin \{0,1\}$ be a new symbol. Let $L_3 = (L_2\$)^*$. We can see that for all live strings u and v of L_3, $u\$v$ is also live with respect to L_3. Indeed, it is also true that the set of live strings of L_3 is exactly L_3. Moreover, the second condition of Theorem 1 still holds for L_3. By Theorem 1, any sweeping automaton for L_3 has at least $2^n - 1$ states.

As another example, we can re-define $L_1 = \{xy \mid x, y \in \{0,1\}^n, x \neq y\}$. Following the same technique as in the previous paragraph, we define L_3. For $0 \leq i \leq 2^n - 1$, let α_i denote the n-bit binary string of value i, while β_i is defined as α_{m-i} where $m = 2^n - 1$. In applying Theorem 1, we need only to consider α_i, β_i for $1 \leq i \leq m$; that is, α_0 and β_0 are not considered. By Theorem 1, any sweeping automaton for L_3 has at least $2^n - 1$ states.

5. Open Question

In [11], Schmidt introduced a technique, which we summarize in the following lemma, for proving lower bounds for the size of unambiguous finite automata.

Lemma 9 *Let L be a language. Suppose there exist strings α_i and β_i for $1 \leq i \leq m$. Let A be an $m \times m$ matrix such that $A[i,j] = 1$ if $\alpha_i\beta_j \in L$, and $A[i,j] = 0$ otherwise. Then any unambiguous finite automaton for L has at least* rank$[A]$ *states.*

Specifically, Schmidt's lemma can be used to prove the following lemma.

Lemma 10 *Let L be a language and let α_i and β_i, for $1 \leq i \leq m$, be strings such that*

- *$\alpha_i\beta_j \in L$ if $i + j \geq m + 1$,*
- *$\alpha_i\beta_j \notin L$ if $i + j = m$.*

Then any unambiguous finite automaton for L has at least m states.

Proof. By Lemma 9 and the fact that the rank of A is m. □

It is interesting to notice the similarity between the last lemma and the statement of Theorem 1. One wonders if Theorem 1 can be generalized to a formulation that looks like Schmidt's technique as summarized in Lemma 9, which leads to the following open question:

Let L be a language and let x be a string such that for all live strings u and v, uxv is live. Suppose there exist strings α_i and β_i for $1 \leq i \leq m$. Let A be an $m \times m$ matrix such that $A[i,j] = 1$ if $\alpha_i\beta_j$ is live, and $A[i,j] = 0$ otherwise. Is it true that any sweeping automaton for L has at least rank$[A]$ states?

References

[1] P. BERMAN, A note on sweeping automata. In: *Proc. 7th Int. Conf. on Automata, Languages, and Programming* (*ICALP*). LNCS 85, Springer-Verlag, 1980, 91–97.

[2] J. Hromkovič, G. Schnitger, Nondeterminism versus determinism for two-way finite automata: generalizations of Sipser's separation. In: *Proc. 30th Int. Conf. on Automata, Languages, and Programming* (*ICALP*). LNCS 2719, Springer-Verlag, 2003, 439–451.

[3] C. A. Kapoutsis, Small sweeping 2NFAs are not closed under complement. In: *Proc. 33rd Int. Conf. on Automata, Languages, and Programming* (*ICALP*). LNCS 4051, Springer-Verlag, 2006, 144–156.

[4] C. A. Kapoutsis, R. Kralovic, T. Momke, An exponential gap between Las Vegas and deterministic sweeping finite automata. In: *Proc. Int. Symp. on Stochastic Algorithms: Foundations and Applications*. 2007, 130–141.

[5] H. Leung, Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata. *SIAM Journal on Computing* **27** (1998), 1073–1082.

[6] H. Leung, Tight lower bounds on the size of sweeping automata. *Journal of Computer and System Sciences* **63** (2001), 384–393.

[7] A. Meyer, M. Fischer, Economy of description by automata, grammars, and formal systems. In: *Proc. 12th SWAT Symposium*. 1971, 188–191.

[8] S. Micali, Two-way deterministic finite automata are exponentially more succinct than sweeping automata. *Information Processing Letters* **12** (1981), 103–105.

[9] F. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Transactions on Computers* **20** (1971), 1211–1214.

[10] W. J. Sakoda, M. Sipser, Nondeterminism and the size of two way finite automata. In: *Proc. 10th ACM Symp. on Theory of Computing*. 1978, 275–286.

[11] E. Schmidt, *Succinctness of descriptions of context-free, regular, and finite languages*. Ph.D. Thesis, Cornell University, Ithaca, N.Y., 1978.

[12] M. Sipser, Lower bounds on the size of sweeping automata. *Journal of Computer and System Sciences* **21** (1980), 195–202.

Appendix

Let F' be an $m \times m$ matrix over the field of integers mod 2 such that if $i + j \geq m + 1$ then $F'[i, j] = 1$, and if $i + j = m + 1$ then $F'[i, j] = 0$. As before, an entry labelled $*$ can hold 0 or 1. We want to show that F' has the full rank m. A picture for F' is given in Figure 2.

By adding the last row of F' to each of the previous rows, we obtain a new matrix F'' which picture is illustrated in Figure 3.

In matrix F'', the entries labelled $*$'s consist of the following index pairs:

$$(1,1),(1,2),\ldots,(1,m-2),(2,1),(2,2),\ldots,(2,m-3),$$
$$(3,1),(3,2),\ldots,(3,m-4),\ldots,(m-4,1),(m-4,2),(m-3,1).$$

$$\begin{bmatrix}
* & * & * & * & * & * & * & * & * & * & * & * & * & 0 & 1 \\
* & * & * & * & * & * & * & * & * & * & * & * & 0 & 1 & 1 \\
* & * & * & * & * & * & * & * & * & * & * & 0 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & * & * & * & 0 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & * & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
* & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Figure 2: Structure of F' when m is 15

$$\begin{bmatrix}
* & * & * & * & * & * & * & * & * & * & * & * & * & 1 & 0 \\
* & * & * & * & * & * & * & * & * & * & * & * & 1 & 0 & 0 \\
* & * & * & * & * & * & * & * & * & * & * & 1 & 0 & 0 & 0 \\
* & * & * & * & * & * & * & * & * & * & 1 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Figure 3: Structure of F'' when m is 15

In a series of steps that involve elementary row transformations, we can turn all $*$-labelled entries to 0's in the order of

$$(1, m-2), (1, m-3), \ldots, (1, 1), (2, m-3), (2, m-4), \ldots, (2, 1),$$
$$(3, m-4), (3, m-5), \ldots, (3, 1), \ldots, (m-4, 2), (m-4, 1), (m-3, 1).$$

Initially, if the entry $(1, m-2)$ is not already of the value 0, we can turn it to 0 by adding row 2 to row 1. Next, if the current entry $(1, m-3)$ is not already of the value 0, we can turn it to 0 by adding row 3 to row 1. The technique can be repeated to turn all $*$-labelled entries to hold 0's. Therefore, we obtain a new matrix F''' of

the form as illustrated in the Figure 4.

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Figure 4: Structure of F''' when m is 15

By elementary row transformations, we can turn the last row of F''' to hold 0's except for the very last entry which has the value 1. The resulting matrix F'''', which has the full rank, is illustrated in Figure 5.

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Figure 5: Structure of F'''' when m is 15

(Received: October 20, 2008; revised: March 6, 2009)