A NOTE ON SWEEPING AUTOMATA

Piotr Berman
Mathematical Institute of PAS
Nowowiejska 10/50, 00 653-Warszawa
Poland

1. Introduction

Last year at the Eleventh Annual ACM Symposium on Computing (May 1979) Michael Sipser presented his paper: "Lower Bounds on the Size of Sweeping Automata". In this paper he stated a new question concerning automata size arised during efforts at solving the L = ?NL problem. Namely, this problem is related to the minimal size of two-direction finite automata for certain languages in cases when they are deterministic and nondeterministic (2dfa and 2nfa respectively). For details see [1] and [2] .

In his paper M. Sipser defined sweeping automata, denoted sa, as such 2dfa which do not change the direction of motion except at the ends of the input tape. He proved the theorem that the relationship between the minimal size of sa and 2nfa for certain languages implies an aswer to the L = ?NL problem (in fact he claims that his proof is the same as that given by Lingas on a similar theorem in [1], and it really is). Moreover Sipser proved that for the series of regular languages B_n the minimal sizes of 1nfa and sa which recognize B_n are n and 2^n respectively. He also conjectured that there is a series of languages for which minimal sizes of 2dfa and sa are O(n) and 2^n respectively. The proof of this fact is the subject of this work.

2. Main result

DEFINITION. Sweeping automaton is such a 2dfa that changes the direction of motion from rightward to leftward only over the letter $\dashv$ and from leftward to rightward only over the letter $\vdash$.

Let $\sum$ be a finite alphabet, $\{\vdash, \dashv\} \cap \sum = \emptyset$, and A is a sa. Then the recognizing of $L \subset \sum^{*}$ by A means that A halts on the input $\vdash w \dashv$ in the accepting state iff $w \in L$.

THEOREM. There is a series of regular languages C_n and $c > 0$ that for any n the minimal sizes of 2dfa and sa recognizing C_n are less than cn, and at least 2^n respectively. Symbols of the alphabet of C_n are nonnegative integers not greater than 2^n:

$$\Sigma_n = \{0,\ 1,\ 2, \dots,\ 2^n\}$$
$$C_n = -(\Sigma_n^{*}\ 012\dots(2^n - 1)\ \Sigma_n^{*})$$

3. Proof of 2dfa's upper bound

C_n may be recognized by a (4n + 3)-state 2dfa which will check whether two consecutive letters represent two consecutive numbers by comparison of their binary digits. Say that s + 1 = t and digits of s and t are $\alpha_1, \dots, \alpha_n$ and $\beta_1, \dots \beta_n$ respectively. It is easy to see that then there is a number i such tat:

(i) $1 < i \leqslant n$

(ii) $\bigwedge_{1 \leqslant j < i} \alpha_j = \beta_j$

(iii) $\alpha_i = 0$ and $\beta_i = 1$

(iv) $\bigwedge_{i < j \leqslant n} \alpha_j = 1$ and $\beta_j = 0$

Obviously the existence of such i implies also that s + 1 = t.

Now it is easy to design a (4n + 3)-state 2dfa which recognizes C_n. First in the initial state it seeks the letter 0. If 0 is found the automaton subsequently checks (using 4n states) whether consecutive letters represent consecutive numbers. Another state is needed to make the transition from one checking to another. If the last letter for which this test will succeed is the $(2^n - 1)$ one the automaton will reject the word. If not it will be again in the initial state.

4. Proof of sa lower bound

The proof will be similar to Sipser's one in 3 , it differs mainly in an example of a series of languages. C_n has three properties of B_n that enable us to proof that it has the same lower bound for sa size, i.e. 2^n. These properties are the following:

1^o There is $x \in \sum_n$ that $w \in C_n$ and $v \in C_n$ iff $wxv \quad C_n$ (this x is 2^n).

2^o There is $t \in \sum^*$ that $t \notin C_n$, $|t| = 2^n$ and $uvw = t$ and $|v|>0$ implies $uw \in C_n$ (this t is $012\ldots(2^n - 1)$).

3^o If $u \notin C_n$ then for any v and w $vuw \notin C_n$.

Now we will describe an automata model which is used in the proof.

DEFINITION. A parallel finite automaton P is a string of 1dfa, called components, and an accepting set of state strings $F \subset Q_1 \times \ldots \times Q_k \times Q_1 \times \ldots \times Q_k$, where Q_i is the state set of A_i.

P recognizes L means that $w \in L$ iff $\langle A_1(w), \ldots, A_k(w), A_1(w^r), \ldots \ldots, A_k(w^r) \rangle \in F$, where $A_i(w)$ is the state reached by running A_i on w and where w^r is the reverse of w.

The lower bound proof is divided into two lemmas.

LEMMA 1. If for a language L there is a k-state sa recognizing L then there is a parallel automaton accepting L whose components all have k states.

For the proof see [3].

LEMMA 2. If a parallel automaton recognizes C_n then one of its components has at least $2^n - 1$ states.

In the proof of lemma 2 some notations are used. If q is a state of a 1dfa then q(s) denotes the state reached by starting at state q and applying an input word s. For S a set of words and R a set of states let R(S) denote $\{q(s) : q \in R \text{ and } s \in S\}$. $\# K$ denotes the cardinality of the set K. A word over $\sum_n$ will be called j-word if it belongs to C_n, otherwise it will be called s-word.

DEFINITION. Let A be a 1dfa with a state set Q_A. A j-word g is A-generic if $\# Q(g) \leqslant \# Q(t)$ for any j-word t. Say g is k-generic if it is A-generic for every 1dfa A with at most k states. Say g is k^r-generic if g is j-word and for any j-word t and k-state A holds $\# Q_A(g^r) \leqslant \leqslant \# Q_A(t^r)$.

SUBLEMMA. There are k-generic and k^r-generic words.

Proof. For every 1dfa A let $f_A : C_n \to N$, $f(w) = \# Q_A(w)$.
If $m = \min f(C_n)$, than any member of $f^{-1}(m)$ is A-generic. Next, if w is A-generic and vw and wu are j-words then vw and wu are also A-generic (because A is deterministic and $\# Q_A(s)$ cannot increase during scanning a word). Since the number of different k-state 1dfa's is finite, one can list them $A_1, A_2, \ldots, A_p$.
If w_i is A_i-generic for $i = 1, 2, \ldots, p$, then $w = w_1 x w_2 x \ldots x w_p$ is k-generic. The case of k^r-generic word is similar.

Let us assume that $P = \langle\langle A_1, \ldots, A_k\rangle, F\rangle$ recognizes C_n, Q_i is the state set of A_i, $m = \max(\# Q_1, \ldots, \# Q_k)$, g is m-generic, w is m^r-generic and s is the shortest s-word $01\ldots(2^n - 1)$.

Now some remarks. It is easy to see that if a word t is m-generic (m^r-generic) then $2^n t 2^n$ is also m-generic (m^r-generic). Thus we may assume that the words g and w are of the $2^n t 2^n$ form. For $i = 1, \ldots, k$ let $R_i = Q_i(g)$ and $S_i = S_i = Q_i(w^r)$. Three facts are to be established.

1) For any word v $\qquad R_i(vg) \subset R_i$

$\qquad\qquad S_i(v^r w^r) \subset S_i$

2) If vg is j-word then $\quad R_i(vg) = R_i$ and

if wv is j-word then $\quad S_i(v^r w^r) = S_i$

3) If v is s-word then exists such i that

$R_i(wvg) \neq R_i$ or $\qquad S_i(g^r v^r w^r) \neq S_i$

Statement 1) holds since $R_i(vg) = (R_i(v))(g) \subset Q_i(g) = R_i$

$$S_i(v^r w^r) = (S_i(v^r))(w^r) \subset Q_i(w^r) = S_i$$

Statement 2) holds since $Q_i(gvg) = R_i(vg) \subset R_i = Q_i(g)$ and $\# Q_i(gvg) \geqslant \# Q_i(g)$, because g is m-generic and gvg j-word (as the concatenation of two j-words with the first one ended by 2^n). The case of S_i is similar.

With regard to statement 3) let us assume by contradiction that for any i $R_i(wvg) = R_i$ and $S_i(g^r v^r w^r) = S_i$. Then $\pi(q) = q(wvg)$ is a permutation on R_i (for any i) and $\tau(q) = q(g^r v^r w^r)$ is a permutation on S_i.
Hence for any i $A_i(gw) = q_o^{\ i}(gw) = q_o^i(g(wvg)^{m!}w) = A_i(g(wvg)^{m!}w)$ and similarly $A_i((gw)^r) = A_i([g(wvg)^{m!}w]^r)$, what yields the contradiction because P cannot distinguish j-word gw from s-word $g(wvg)^{m!}w$.

Let i be such that $R_i(wsg) \neq R_i$ or $S_i(g^r s^r w^r) \neq S_i$. To simplify the proof let us assume that $R_i(wsg) \neq R_i$ (we know that $R_i(wsg) \subset R_i$). By applying the more sophisticated reasoning of Sipser from [3] we may prove that $\# Q_i \geqslant 2^{n/2} - 1$.

We know that $f: R_i \rightarrow R_i$, $f(q) = q(wsg)$ is not one-one, hence there are $q_1, q_2 \in R_i$ that $q_1(wsg) = q_2(wsg)$ and $q_1 \neq q_2$. Then there are x, y such that $0 \leqslant x < y \leqslant 2^n - 1$ and $\langle q_1(w01\ldots x), q_2(w01\ldots y)\rangle =$ $= \langle q_1(w01\ldots y), q_2(w01\ldots y)\rangle$ or $\# Q_i \geqslant 2^{n/2} - 1$ (otherwise we have 2^n different members of a set which has at most $(2^{n/2} - 1)^2 < 2^n$ members). The latter is our thesis and first yields a contradition.

Indeed, then we have $q_1(w01\ldots x(y+1)\ldots(2^n - 1)g) = q_1(w01\ldots y(y+1)\ldots$ $\ldots(2^n - 1)g) = q_2(w01\ldots y(y+1)\ldots(2^n - 1)g) = q_2(w01\ldots x(y+1)\ldots$ $\ldots(2^n - 1)g)$ what means that for j-word $t = w01\ldots x(y+1)\ldots(2^n - 1)g$ holds $q_1(t) = q_2(t)$, so $R_i(t) \neq R_i$ what contradicts the statement 2).

With this weaker proof the Theorem holds for $\overline{C}_n = C_{2n}$ and $c = 8 + \varepsilon$, while the stronger method yields $c = 4 + \varepsilon$.

5. Final remarks

For any n the language C_n may be recognized by a (4n - 2)-state 1nfa*). Hence the upper bounds for sa recognizing C_n yields a relationship between 1nfa and sa which is similar to Sipser's one although weaker. Unfortunately these relationship do not imply any answer to L = ?NL question because the length of words that are used in the proof is exponential with respect to n. If somebody proved a similar relationship for a series of languages with a polynomial length of words then he would obtain a negative answer to the L = ?NL question.

---

*) The design of (4n - 2)-state 1nfa recognizer of C_n is based upon the following observation. For a string of n-digit binary numbers $s_1 \ldots s_p$ it holds

$$s_1 \ldots s_p = 01 \ldots (2^n - 1) \text{ iff}$$

(i) $s_1 = 0$ and $s_p = (2^n - 1)$

(ii) $s_i(n) = 1 - s_{i+1}(n)$ for $i = 0, 1, \ldots, p-1$

(iii) $s_i(j) = s_{i+1}(j)$ and $s_i(j+1) = s_{i+1}(j+1)$
or
$s_i(j+1) + 2s_i(j) + 1 = s_{i+1}(j+1) \mod 4$
for $i = 0, 1, \ldots, p-1$ and $j = 1, 2, \ldots, n-1$

(iv) $s_i \neq 0$ for $i = 2, 3, \ldots, p$

To accept a word w as a j-word it is necessary to know that this statement does not hold for any subword of w. This means that after each letter 0 there is another 0 or 2^n before any $(2^n - 1)$, or no $(2^n - 1)$ at all, or (and this case is most important) there is a place between this 0 and the nearest $(2^n - 1)$ that is in contradiction with point (ii) or (iii). Such place can be found nondeterministically using 4n - 4 states. If the recognizer does not find such place it will reject the whole word.

REFERENCES

[1] Berman P. and A. Lingas, On the complexity of regular languages in terms of finite automata, ICS PAS Report no 304, 1977, Warszawa.

[2] Sakoda W.J. and M. Sipser, Nondeterminism and the size of two-way finite automata, Tenth Annual STOC, 1978.

[3] Sipser M., Lower Bounds on the size of sweeping automata, Eleventh Annual ACM Symposium on Theory of Computing, May 1979.