Theoretical Computer Science 47 (1986) 149–158
North-Holland

FINITE AUTOMATA AND UNARY LANGUAGES

Marek CHROBAK
Institute of Informatics, Warsaw University, PKIN VIII p, 00-901 Warsaw, Poland

Communicated by G. Mirkowska
Received February 1986
Revised May 1986

Abstract. We prove that $O(e^{\sqrt{n \log n}})$ states are sufficient to simulate an n-state 1nfa recognizing a unary language by a 1dfa. The lower bound is the same. Similar tight bounds are shown for the simulation of a 2dfa by a 1dfa and a 1nfa. We also show that $O(n^2)$ states are sufficient and necessary to simulate an n-state 1nfa recognizing a unary language by a 2dfa.

1. Notation

By an *fa* we denote a finite automaton. Using appropriate prefixes we specify what kind of an fa we consider. The possible prefixes are formed of the symbols:

- 1: one-way,
- 2: two-way,
- d: deterministic,
- n: nondeterministic,
- a: alternating.

For example, a 2dfa is a two-way deterministic finite automaton. A *unary language* is a language over a one-symbol alphabet. A *unary fa* is an fa with a one-symbol input alphabet. Clearly, a unary fa recognizes a unary language. Also, if an fa recognizes a unary language, then we can make it unary by deleting unnecessary symbols and modifying the next-state function.

We will consider only unary fa's. In this case, input words can be identified with nonnegative integers and we will write x instead of a^x. Moreover, a unary 1nfa is simply a digraph whose vertices are states and whose edges correspond to the next-state function. Therefore, we can talk of vertices, edges, cycles, strongly connected components, etc. of a 1nfa. We will describe a 1nfa as a quadruple (Q, q_0, E, F), where Q, q_0, F are as usual and E is the set of edges corresponding to the next-state function. As usual, $L(A)$ is the language accepted by an fa A. A *sweeping 2nfa* is a 2nfa making reversals only at the endmarkers.

All logarithms are to base e. Sometimes we will write $\exp(x)$ instead of e^x.

By $\gcd(x_1, \ldots, x_k)$ we denote the greatest common divisor of $x_1, \ldots, x_k$, and by $\operatorname{lcm}(x_1, \ldots, x_k)$ their smallest common multiple. The following function will play a basic role in our investigations:

$$F(n) = \max\{\operatorname{lcm}(x_1, \ldots, x_k) \mid x_1 + \cdots + x_k = n\}.$$

0304-3975/86/$3.50

For $a_1, \ldots, a_k$ such that $\gcd(a_1, \ldots, a_k) = 1$ we denote by $G(a_1, \ldots, a_k)$ the greatest number b such that the Diophantine equation $a_1x_1 + \cdots + a_kx_k = b$ has no solution in natural numbers.

By $H(n)$ we denote the function $H(n) = e^{\sqrt{n \log n}}$.

2. Introduction

In this paper we investigate the following classical problem of the automata theory: given two classes of fa's C_1, C_2, how many states are necessary and sufficient to simulate n-state automata from C_1 by automata in C_2. It is well known that all 'reasonable' fa's, even 2afa's, recognize only regular sets [2, 5, 12, 21]. However, stronger fa's can describe a given language using less states. A fundamental theorem about one-way fa's is that 2^n states suffice to simulate any n-state 1nfa by a 1dfa and there are examples showing that this bound cannot be improved [7, 16, 18, 21, 29]. Other known bounds are:

- each n-state 2dfa can be simulated by a 1dfa with $O(n^n)$ states [1, 2, 21, 26];
- each n-state 2nfa can be simulated by a 1dfa with $O(2^{n^2})$ states [2, 21];
- each n-state 1afa can be simulated by a 1dfa with $O(2^{2^n})$ states [5].

It is also known that these bounds are asymptotically best possible [1, 2, 5, 18, 21].

The problem of simulating 1nfa's (or 2nfa's) by 2dfa's was stated by Sakoda and Sipser in [22] and still remains open. Berman and Lingas [2] combine bounds for the simulation of 2dfa's and 2nfa's by 1dfa's to show that, in general, $\Omega(n^2/\log n)$ states are necessary to simulate a 2nfa by a 2dfa. Also, as it was shown by Sipser [27], sweeping 2dfa's (that is, 2dfa's making reversals only at the endmarkers) require 2^n states to simulate 1nfa's. Sakoda and Sipser conjecture that this also holds for all 2dfa's. The problem has not only interest of its own. The following theorem, due to Berman and Lingas [2], relates it to the DLOG = NLOG problem: If DLOG = NLOG, then there is a polynomial p such that, for each integer m and each n-state 2nfa A, there is a $p(mn)$-state 2dfa B such that $L_{mn}(A) = L_{mn}(B)$, where $L_k(A)$ is the set of words in $L(A)$ of length at most k.

In this paper we consider the particular case of the above problems when the languages considered are unary. The problem was mentioned by Sipser [27]. Note that the proofs of lower bounds in the general case essentially use the fact that the alphabet consists of at least two letters. The proofs, based on the Myhill–Nerode theorem, are of information-theoretic nature: usually, it is shown that a 1dfa with too few states is not able to carry enough information through the input word. In case of unary languages, as we will show later, we face number-theoretic problems which, fortunately, are now quite satisfactorily solved. Using these number-theoretic methods we prove that

(1) each unary n-state 1nfa can be simulated by a 1dfa with $O(H(n))$ states and this bound is asymptotically optimal;

(2) each unary n-state 2dfa can be simulated by a 1dfa with $O(H(n))$ states and this bound is asymptotically optimal;

(3) each unary n-state 1nfa can be simulated by a 2dfa with $O(n^2)$ states and this bound is asymptotically optimal.

A weaker result similar to (1) was obtained by Liubicz in [15]. He proved that, in our notation, the upper bound in (1) is $O(nH(n))$. The proof in [15] is matrix-oriented, using some results about positive matrices. Our method is purely combinatorial. We transform each 1nfa to the normal form (Lemma 4.3), from which we derive (1) and (3). The normal-form lemma has interest of its own since it says that each unary 1nfa A can be substituted by an equivalent 1nfa A' making only one guess, and the size of A' is bounded by a small-degree polynomial of the size of A.

In (2), the lower bound applies also to 1nfa's instead of 1dfa's. The $\Omega(n^2)$ lower bound in (3) is the best lower bound for the simulation of 2nfa's by 2dfa's we know of. It strengthens the mentioned result of Berman and Lingas [2] for 2nfa's, not only because the function is of higher order but also because it concerns very simple 2nfa's, namely unary 1nfa's. It is amazing that the proof is so simple. Unfortunately, our technique cannot give better lower bounds.

Unary languages have already been studied in the theory of automata and complexity. For example, they can be used as witness languages for proving separation results about space complexity classes [8, 11, 24, 25], or for solving the '$k+1$ versus k heads'-problem for multihead automata [20]. Also, some known open problems, as, for example, the LBA problem, can be reduced to problems about unary languages [19]. In [8, 11], nonregular unary languages of space complexity $O(\log\log n)$ were constructed. Unary languages have also been studied in the theory of AFLs [9]. This research was partially motivated by the above results.

Except results using diagonalization, it seems almost a rule that when dealing with unary languages, one arrives at number-theoretic problems [8, 9, 10, 11]. For example, the proofs that the languages constructed in [8, 11] are of space complexity $O(\log\log n)$ use results from the analytic number theory.

3. Two number-theoretic problems

The problem of finding a good approximation for $F(n)$ is known as Landau's problem [13, 14, 28, 30]. The problem is usually stated in terms of permutation groups: what is the maximal order in S_n, the symmetric group on n symbols? (the order of P in S_n is the order of the cyclic subgroup generated by P). Landau [13, 14] has proved that

$$\lim_{n\to\infty} \log F(n)/\sqrt{n \log n} = 1.$$

The best known approximation is due to Szalay:

Theorem A ([28])

$$F(n) = \exp[(n \log n + \log\log n - 1 + (\log\log n - 2 + o(1))/\log n)^{1/2}].$$

For our purpose the bound in the corollary below will be sufficient.

Corollary A. $F(n) = O(H(n))$.

The second problem concerns linear Diophantine equations. First consider equations with two variables. The following well-known fact will be used in our proofs.

Fact A. *If* $\gcd(a, b) = 1$, *then the greatest number such that the equation* $ax + by = c$ *has no solution in natural numbers is* $(a-1)(b-1)-1$.

Frobenius stated the problem of generalizing this result, that is, of finding a good approximation for $G(a_1, \ldots, a_k)$. There are quite a lot of papers on Frobenius's problem, although the known approximations still seem far from the exact value. We will use the following result.

Theorem B ([3, 15]). *Let* $a_1 < \cdots < a_k$ *and* $\gcd(a_1, \ldots, a_k) = 1$. *Then we have* $G(a_1, \ldots, a_k) \leq (a_k - 1)(a_1 - 1)$.

Erdös and Graham [6] give a more accurate approximation, as well as some more references to the problem. The corollary below follows from Theorem B.

Corollary B. *Let* $a_1, \ldots, a_k$ *be natural numbers* $\leq n$. *Let* X *be the set of all* x*'s for which the Diophantine equation* $a_1x_1 + \cdots + a_kx_k = x$ *is solvable in natural numbers. Then the set of numbers in* X *greater than* n^2 *is an arithmetic progression with period* $\gcd(a_1, \ldots, a_k)$.

4. 1nfa versus 1dfa

In this section, we will present tight bounds for the simulation of unary 1nfa's by 1dfa's.

Definition 4.1. Let A be a unary 1nfa such that r of its vertices are in cycles and s of them are not (so $r+s$ is the number of A's states). Then we define $S(A) = (r, s)$.

Definition 4.2. A unary 1nfa $A = (Q, q_0, E, F)$ is in *normal form* if it has the following properties:

(a) $\quad Q = \{q_0, \ldots, q_m\} \cup C_1 \cup \cdots \cup C_k,$

where $C_i = \{p_{i,0}, p_{i,1}, \ldots, p_{i,y_i-1}\}$, for $i = 1, \ldots, k$,

(b) $$E = \{(q_i, q_{i+1}) \mid i = 1, \ldots, m-1\}$$
$$\cup \{(p_{i,j}, p_{i,j+1}) \mid i = 1, \ldots, k \text{ and } j = 0, \ldots, y_i - 1\}$$
$$\cup \{(q_m, p_{i,0}) \mid i = 1, \ldots, k\}$$

(the addition $j+1$ in the second component is mod y_i).

Informally, A consists of a path from q_0 to q_m and cycles $C_1, \ldots, C_k$ connected to q_m. An example of an automaton in normal form is shown in Fig. 1, for $m=4$, $k=3$, $y_1=4$, $y_2=5$, $y_3=3$.

Lemma 4.3. *For each unary n-state* 1*nfa A there is an equivalent* 1*nfa A′ in normal form such that* $S(A') \leqslant (n, \mathrm{O}(n^2))$.

Proof. Let $A=(Q, q_0, E, F)$. Without any loss of generality, we can assume that $F=\{q_F\}$.

A *superpath in A* is a subgraph of A denoted by

$$\alpha = P_1 D_1 P_2 D_2 \ldots P_t D_t P_{t+1},$$

where

(a) for $i=1, \ldots, t$, D_i is a strongly connected component of A;

(b) for $i=1, \ldots, t+1$, P_i is a path in A whose inner points do not belong to strongly connected components of A;

(c) the first vertex of P_1 is q_0, the last vertex of P_{t+1} is q_F;

(d) for $i=1, \ldots, t$, the last vertex of P_i belongs to D_i;

(e) for $i=2, \ldots, t+1$ the first vertex of P_i belongs to D_{i-1}.

Let L_α be the set of all lengths of paths from q_0 to q_F in α. Let Π be the set of all superpaths in A. Then $L(A)$ is the union of all sets L_α for α in Π.

For a strongly connected component D of A, let $\Pi(D)$ be the set of all superpaths α such that D is the last strongly connected component in α. Let Π_0 be the set of all superpaths which do not contain strongly connected components (that is, simple paths from q_0 to q_F). Then

$$L(A) = \bigcup_{\alpha \in \Pi_0} L_\alpha \cup \bigcup_{D} \bigcup_{\alpha \in \Pi(D)} L_\alpha,$$

where the second sum is taken over all strongly connected components D in A.

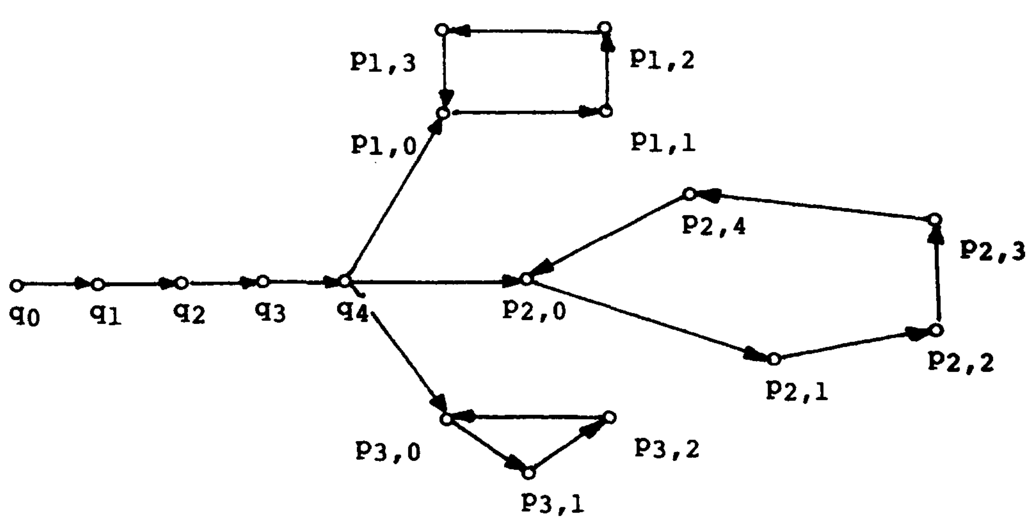

Fig. 1. A unary 1nfa in normal form.

Let us now fix a strongly connected component D and a superpath α in $\Pi(D)$, $\alpha = P_1D_1P_2D_2 \ldots P_tD_tP_{t+1}$, $D_t = D$. Let $\gcd(\alpha)$ be the greatest common divisor of the lengths of the cycles in α, and $\gcd(D)$ the greatest common divisor of the lengths of the cycles in D. Clearly, $\gcd(\alpha)$ divides $\gcd(D)$.

Let $x \in L_\alpha$ and let R be the path in α of length x. Then $x = x_0 + a_1x_1 + \cdots + a_px_p$, where x_0 is the length of the path obtained from R by deleting the cycles, and $a_0, \ldots, a_p$ are the lengths of all cycles in α. Let $m = n^2 + n$. By Corollary B we obtain that $L_\alpha = L_\alpha^1 \cup L_\alpha^2$, where L_α^1 is the subset of L_α containing numbers $\leqslant m$, and L_α^2 is an arithmetic progression with period $\gcd(\alpha)$. It is easy to see that we could substitute for α an 'equivalent' subgraph consisting of a single path and a cycle of length $\gcd(\alpha)$ attached to it. Instead we will use a cycle of length $\gcd(D)$, so, in this way, it can be used as a common cycle for all superpaths in $\Pi(D)$.

Let $T_1, \ldots, T_k$ be all strongly connected components of A in some fixed order. Then A' will have the set of states as in Definition 4.2, where m and k are the numbers defined above and $y_i = \gcd(T_i)$, $i = 1, \ldots, k$. We only have to mark appropriate states as accepting. Clearly, $q_x \in F'$ iff $x \in L_\alpha^1$ for some superpath α (for $\alpha \in \Pi_0$ we set $L_\alpha^1 = L_\alpha$). Similarly, $p_{i,x} \in F'$ iff $x + m \in L_\alpha^2$ for some superpath $\alpha \in \Pi(T_i)$, $i = 1, \ldots, k$. □

Theorem 4.4. *For each unary n-state 1nfa A there is an equivalent 1dfa B with $O(H(n))$ states.*

Proof. For a given 1nfa A with n states, we first construct a 1nfa A' in normal form such that $S(A') = (r, s)$, for $r \leqslant n$ and $s = O(n^2)$, as in Lemma 4.3. Let $y_1, \ldots, y_k$ be the lengths of cycles $C_1, \ldots, C_k$ and $y = \operatorname{lcm}(y_1, \ldots, y_k)$. Then $B = (Q, q_0, E, F)$, where

$$Q = \{q_0, \ldots, q_{s-1}, q_s, \ldots, q_{y+s-1}\},$$
$$E = \{(q_i, q_{i+1}) \mid i = 0, 1, \ldots, y+s-2\} \cup \{(q_{y+s-1}, q_s)\}$$

and F is defined as follows. If q_i, $0 \leqslant i < s$, is an accepting state in A', then $q_i \in F$. Also, if some $p_{i,j}$ is accepting in A', then $q_{s+t} \in F$ for each t such that $t - j = cy_i$ for some integer c. It is straightforward to check that $L(B) = L(A)$. B has $y + s - 2 = O(F(n)) + O(n^2) = O(F(n))$, states. Using Corollary A we then obtain the theorem. □

Theorem 4.5. *For each n there is a unary n-state 1nfa A such that each 1dfa recognizing $L(A)$ requires $\Omega(H(n))$ states.*

Proof. We will show that $F(n-1)$ states are necessary. Let n be arbitrary but fixed and $x_1, \ldots, x_k$ be the numbers for which the maximum in the definition of $F(n-1)$ is attained. W.l.o.g. we can assume that $x_1 < \cdots < x_k$. From the properties of $F(n)$ it is also known that they are relatively prime in pairs. Let also

$$L = \{cx_i \mid i = 1, \ldots, k,\ c \in \mathbb{N}\}.$$

Consider a 1nfa $A=(Q, q_0, E, F)$ such that

$$Q=\{q_0\}\cup\{p_{i,j}\mid 1\leq i\leq k, 0\leq j<x_i\},$$

$$F=\{p_{i,0}\mid 1\leq i\leq k\},$$

$$E=\{(q_0, p_{i,1})\mid i=1,\ldots,k\}\cup\{(p_{i,j}, p_{i,j+1})\mid i=1,\ldots,k, j=0,\ldots,x_i-1\}$$

(the addition is mod x_i).

So A consists of the initial state and several cycles. Obviously, $L(A)=L$ and A has n states.

Now, let R be the Myhill–Nerode equivalence relation: $R(u, v)$ iff ($u+z\in L$ iff $v+z\in L$ for each z).

We will show that the index of R is at least $y=F(n-1)$. In order to do this we will prove that, for each two different $0\leq u, v<y$, it is *not* true that $R(u, v)$ holds. We must find a z such that exactly one of $u+z$, $v+z$ belongs to L.

If $2\leq i\leq k$, or $i=1$ and $x_1\neq 2$, then we define

$$z_i=\begin{cases}0 & u_i\neq 0 \text{ and } v_i\neq 0,\\ 1 & u_i=0, 0\leq v_i<x_i-1 \text{ or } v_i=0, 0\leq u_i<x_i-1,\\ 2 & u_i=0,\ v_i=x_i-1 \text{ or } v_i=0,\ u_i=x_i-1,\end{cases}$$

where $u_i=u \bmod x_i$, $v_i=v \bmod x_i$. If $x_1=2$, then

$$z_1=\begin{cases}0 & u_1\neq v_1 \text{ or } u_1=v_1=1,\\ 1 & u_1=v_1=0.\end{cases}$$

By the Chinese Remainder Theorem there is exactly one z such that, for $i=1,2,\ldots,k$, $z \bmod x_i=z_i$. Then, by the definition of z_i, we have that $(u+z)\bmod x_i\neq 0$ and $(v+z)\bmod x_i\neq 0$ for $i=2,3,\ldots,k$.

Suppose first that $x_1=2$ and $u_1\neq v_1$. Then, either $u_1=0$ and $v_1=1$, or $u_1=1$ and $v_1=0$. In both cases we have that exactly one of $u+z$, $v+z$ belongs to L.

Otherwise, let j be the smallest number such that $u_j\neq v_j$. W.l.o.g. we can assume that $(u+z)\bmod x_j>(v+z)\bmod x_j$. Let $t_i=0$ for $i\neq j$ and $t_j=x_j-(u+z)\bmod x_j$. Then again by the Chinese Remainder Theorem, there is exactly one t such that $t \bmod x_i=t_i$ for $i=1,\ldots,k$. Then $u+z+t\in L$, but $v+z+t\notin L$. □

5. 2dfa versus 1dfa

Theorem 5.1. *Each unary n-state 2dfa can be simulated by a 1dfa with* $O(H(n))$ *states.*

Proof. It is easy to show that any unary 2dfa can be substituted by an equivalent sweeping 2dfa without increasing the number of its states. So let $A=(Q, q_0, \delta, F)$ be a sweeping unary 2dfa with n states. A 1dfa B simulating A on input x first checks if x is a word of length $\leq n$ accepted by A. If not, A must make a cycle on each pass on x. Let $y_1, \ldots, y_k$ be the lengths of all of A's cycles. Clearly, $y_1 + \cdots + y_k \leq n$, because no state can be in two different cycles. For two numbers $v > u > n$ such that $v - u = \operatorname{lcm}(y_1, \ldots, y_k)$, A accepts u iff A accepts v. Therefore, the cycle of length y suffices to simulate A on words longer than n. Now the theorem follows from Corollary A. □

Theorem 5.2. *For each n there is a unary n-state 2dfa A such that each 1dfa recognizing $L(A)$ requires $\Omega(H(n))$ states.*

Proof. The theorem follows from the fact that the language L from Theorem 4.5 can be recognized by a 2dfa A with n states. A simply makes k passes over an input x computing $x \bmod x_i$ in the ith pass, $i = 1, 2, \ldots, k$. □

6. 2dfa versus 1nfa

First we strengthen Theorem 5.2 by showing that the lower bound even holds for 1nfa's.

Theorem 6.1. *For each n there is a unary n-state 2dfa A such that each 1nfa recognizing $L(A)$ requires $\Omega(H(n))$ states.*

Proof. Let $x_1, \ldots, x_k$ be the numbers for which the maximum in the definition of $F(n)$ is attained. Let also $L = \{cF(n) \mid c \in \mathbb{N}^+\}$. There is a 2dfa A with n states recognizing L. A behaves similarly to the automaton from the proof of Theorem 5.2, except that it accepts x iff $x \bmod x_i = 0$ for each $i = 1, 2, \ldots, k$. The shortest word in L is $F(n)$. Consider a 1nfa B recognizing L. B must have a simple path of length at least $F(n)$ between the starting state and a final state, because otherwise it would accept a word shorter than $F(n)$. This proves the theorem. □

Theorem 6.2. *Each unary n-state 1nfa A can be simulated by a 2dfa B with $O(n^2)$ states.*

Proof. For a given 1nfa A with n states we construct a 1nfa A' in normal form such that $S(A') = (r, s)$, for $r \leq n$ and $s = O(n^2)$, as in Lemma 4.3. A 2dfa B simulating A' first checks if an input x is $<s$. If so, B accepts iff q_x is an accepting state of A' (the notation is from Lemma 4.3). Otherwise, B makes k passes over the input and the length of the cycle in the jth pass is y_j. So, in the jth pass, B computes $t = (x - s) \bmod y_j$ and accepts x iff $p_{j,t}$ is an accepting state of A. This completes the proof, because B has $r + s = O(n^2)$ states. □

Theorem 6.3. *For each n there is a unary n-state 1nfa A such that each 2dfa recognizing $L(A)$ requires $\Omega(n^2)$ states.*

Proof. Let $L = \{x \mid x = nx_1 + (n-1)x_2 \text{ for } x_1, x_2 \in \mathbb{N}\}$. L can be recognized by a 1nfa A with n states. $A = (Q, q_0, E, F)$ is defined as follows: $Q = \{q_0, \ldots, q_{n-1}\}$, $E = \{(q_i, q_{i+1}) \mid i = 0, \ldots, n-1\} \cup \{(q_1, q_3)\}$ (the addition is mod n) and $F = \{q_0\}$. Let $m = \max(\mathbb{N} - L)$. By Fact A, $m = O(n^2)$. Consider a 2dfa B recognizing L and its computation on m. Suppose that, in all passes on m, B enters a cycle and let $y_1, \ldots, y_k$ be the lengths of these cycles. Then B would reject also $m' = m + \operatorname{lcm}(y_1, \ldots, y_k)$, which contradicts the fact that $m' \in L$. Therefore, there is a pass of B on m without a cycle and the theorem follows. □

7. Final remarks

Informally speaking we have shown that 1nfa's and 2dfa's are hard to simulate by 1dfa's, even if we consider only unary languages. Also, for unary languages, two-way motion is more powerful, in a sense, than guessing, because we can simulate unary 1nfa's by 2dfa's increasing the number of states only polynomially, which is not possible the other way round. Comparing it with the upper bound for 1dfa's one can say that nondeterminism does not help if we want to simulate 2dfa's by one-way fa's.

The following problems are still open:

(1) (The Sakoda–Sipser problem for unary languages.) Does there exist a polynomial p such that each unary n-state 2nfa can be simulated by a $p(n)$-state 2dfa? Actually, the problem of Sakoda and Sipser concerns also 1nfa's, but as we have shown in Section 6, in this case the answer is positive.

(2) What is the relationship between unary 1afa's (or 2afa's) and other fa's? It is easy to show some lower and upper bounds for 1afa's with only universal states. The author believes that unary 1afa's and 2dfa's are polynomially equivalent.

Acknowledgment

I would like to thank W. Rytter and B. Chlebus for many discussions on the problems presented in this paper. W. Rytter has also discovered references [7, 15, 16, 29]. I am especially grateful to Prof. A. Schinzel for pointing out references [6, 28].

References

[1] B. Barnes, A two-way automaton with fewer states than any equivalent one-way automaton, *IEEE Trans. on Comp.* C-20(4) (1974) 474–475.

[2] P. Berman and A. Lingas, On the complexity of regular languages in terms of finite automata, Rept. 304, Polish Academy of Sciences (1977).

[3] A. Brauer, On a problem of partitions, *Amer. J. Math.* 64 (1942) 299-312.

[4] A. Brauer and J.E. Shockley, On a problem of Frobenius, *J. Reine Angew. Math.* **211** (1962) 215-220.
[5] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer, Alternation, *J. Assoc. Comput. Mach.* **28** (1981) 114-133.
[6] P. Erdös and R.L. Graham, On a linear diophantine problem of Frobenius, *Acta Arithm.* **21** (1972) 399-408.
[7] U.L. Ershov, On a conjecture of W.U. Uspenskii, *Algebra i logika* (seminar) **1** (1962) 45-48 (in Russian).
[8] A.R. Friedman and R.E. Ladner, Space bounds for processing contentless inputs, *J. Comput. System Sci.* **11** (1975) 118-128.
[9] J. Goldstine, Some independent families of one-letter languages, *J. Comput. System Sci.* **10** (1975) 351-369.
[10] E.M. Gurari and O.H. Ibarra, Simple counter machines and number-theoretic problems, *J. Comput System Sci.* **19** (1979) 145-162.
[11] J. Hartmanis and L. Berman, On tape bounds for single-letter language processing, *Theoret. Comput. Sci.* **3** (1973) 213-224.
[12] R.E. Ladner, R.J. Lipton and L.J. Stockmeyer, Alternating pushdown automata, *Conf. Proc. IEEE 19th Ann. Symp. on Foundations of Computer Science* (1978) 92-106.
[13] E. Landau, *Handbuch der Lehre von der Verteilung der Primzahlen I* (Teubner, Leipzig/Berlin, 1909) 222-229.
[14] E. Landau, Uber die Maximalordung der Permutationen gegebenen Grades, *Archiv. der Math. und Phys.* **3** (1903) 92-103.
[15] U.I. Liubicz, Bounds for the optimal determinization of nondeterministic autonomic automata, *Sibirskii Matemat. Journal* **2** (1964) 337-355 (in Russian).
[16] O.B. Lupanov, A comparison of two types of finite automata, *Problemy Kibernetiki* **9** (1963) 321-326 (in Russian).
[17] N.S. Mendelsohn, A linear diophantine equation with applications to nonnegative matrices, *Ann. NY Acad. Sci.* **175** (1970) 287-294.
[18] A.R. Meyer and M.J. Fischer, Economy of description by automata, grammars, and formal systems, *Proc. 12th IEEE Symp. on Switching and Automata Theory* (1971) 188-191.
[19] B. Monien, The LBA-problem and the deterministic tape complexity of two-way, one-counter languages over a one-letter alphabet, *Acta Inform.* **8** (1977) 371-382.
[20] B. Monien, Two-way multihead automata over a one-letter alphabet, *RAIRO Inform. Théor.* **14** (1980) 67-82.
[21] M. Rabin and D. Scott, Finite automata and their decision problems, *IBM J. Res. Dev.* **3** (1959) 198-200.
[22] W.J. Sakoda and M. Sipser, Nondeterminism and the size of two-way finite automata, *Proc. 10th ACM Symp. on Theory of Computing* (1978) 275-286.
[23] M. Schmidt, Succintness of description of context-free, regular and finite languages, Ph.D. Thesis, Cornell University, 1978.
[24] J.I. Seiferas, Relating refined space complexity classes, *J. Comput. System Sci.* **14** (1977) 100-129.
[25] J.I. Seiferas, Techniques for separating space complexity classes, *J. Comput. System Sci.* **14** (1977) 73-99.
[26] J. Shepardson, The reduction of two-way automata to one-way automata, *IBM J. Res. Dev.* **3** (1959) 114-125.
[27] M. Sipser, Lower bounds on the size of sweeping automata, *J. Comput. System Sci.* **21** (1980) 195-202.
[28] M. Szalay, On the maximal order in S_n and S_n^*, *Acta Arithm.* **37** (1980) 321-331.
[29] B.A. Trakhtenbort and A.M. Barzdin, *Finite Automata* (Nauka, Moscow, 1970) (in Russian).
[30] P. Turan, Combinatorics, partitions, group theory, in: B. Serge, ed., *Colloq. Internat. sulle Teorie Combinatorie* (Academia Nazionale dei Lincei, Rome, 1976) 181-200.