Remarks on Separating Words

Erik D. Demaine[1], Sarah Eisenstat[1], Jeffrey Shallit[2], and David A. Wilson[1]

[1] MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, {edemaine,seisenst,dwilson}@mit.edu
[2] University of Waterloo, Waterloo, ON N2L 3G1, Canada, shallit@cs.uwaterloo.ca

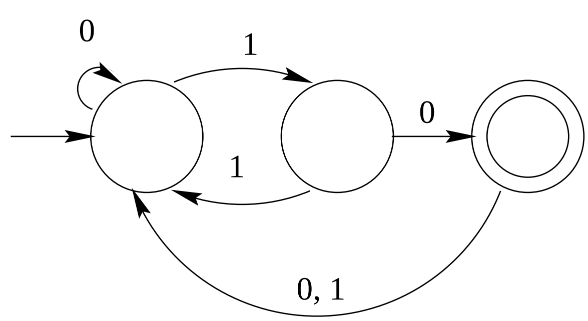
Abstract. The *separating words problem* asks for the size of the smallest DFA needed to distinguish between two words of length $\leq n$ (by accepting one and rejecting the other). In this paper we survey what is known and unknown about the problem, consider some variations, and prove several new results.

1 Introduction

Imagine a computing device with very limited powers. What is the simplest computational problem you could ask it to solve? It is not the addition of two numbers, nor sorting, nor string matching — it is telling two inputs apart: distinguishing them in some way.

Take as our computational model the deterministic finite automaton or DFA. As usual, it consists of a 5-tuple, $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite nonempty set of states, Σ is a nonempty input alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function (assumed to be *complete*, or defined on all members of its domain), $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states.

We say that a DFA M *separates* w and x if M accepts one but rejects the other. Given two distinct words w, x we let $\text{sep}(w, x)$ be the number of states in the smallest DFA accepting w and rejecting x. For example, the DFA below separates 0010 from 1000.



However, by a brief computation, we see that no 2-state DFA can separate these two words. So $\mathrm{sep}(1000, 0010) = 3$. Note that $\mathrm{sep}(w, x) = \mathrm{sep}(x, w)$, because the language of a DFA can be complemented by swapping the reject and accept states.

We let $S(n) = \max_{\substack{w \neq x \\ |w|,|x| \leq n}} \mathrm{sep}(w, x)$. The *separating words problem* is to determine good upper and lower bounds on $S(n)$. This problem was introduced 25 years ago by Goralčík and Koubek [5], who proved $S(n) = o(n)$. It was later studied by Robson [7, 8], who obtained the best upper bound so far: $S(n) = O(n^{2/5}(\log n)^{3/5})$.

As an additional motivation, the separating words problem can be viewed as an inverse of a classical problem from the early days of automata theory: given two DFAs accepting different languages, what length of word suffices to distinguish them? More precisely, given two DFAs M_1 and M_2, with m and n states, respectively, with $L(M_1) \neq L(M_2)$, what is a good bound on the length of the shortest word accepted by one but not the other? The usual cross-product construction quickly gives an upper bound of $mn - 1$ (make a DFA for $L(M_1) \cap \overline{L(M_2)}$). But the optimal upper bound of $m + n - 2$ follows from the usual algorithm for minimizing automata. Furthermore, this bound is best possible [9, Thm. 3.10.6]. For NFAs the bound is exponential in m and n [6].

From the following result, already proved by Goralčík and Koubek [5], we know that the challenging case of word separation comes from words of equal length:

Proposition 1. *Suppose* $|w|, |x| \leq n$ *and* $|w| \neq |x|$. *Then* $\mathrm{sep}(w, x) = O(\log n)$. *Furthermore, there is an infinite class of examples where* $\mathrm{sep}(w, x) = \Omega(\log n)$.

We use the following lemma [10]:

Lemma 1. *If* $0 \leq i, j \leq n$ *and* $i \neq j$, *then there is a prime* $p \leq 4.4 \log n$ *such that* $i \not\equiv j \pmod{p}$.

Proof. (of Proposition 1) Let's prove the upper bound. If $|w| \neq |x|$, then by Lemma 1 there exists a prime $p \leq 4.4 \log n$ such that $|w| \bmod p \neq |x| \bmod p$. Hence a simple cycle of p states serves to distinguish w from x.

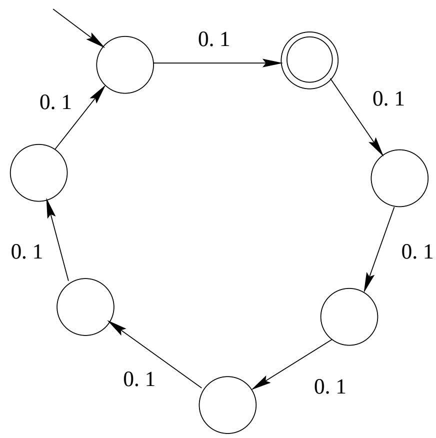
For the other direction, we first recall that a sequence $(p_i)_{i \geq 0}$ is said to be *ultimately periodic* if there exist integers $r \geq 0, s \geq 1$ such that $p_i = p_{r+i}$ for all $i \geq s$. In this case s is called the *preperiod* and r the *period*.

Now we claim that no DFA with n states can distinguish

$$0^{n-1} \quad \text{from} \quad 0^{n-1+\mathrm{lcm}(1,2,\ldots,n)}.$$

To see this, let $p_i = \delta(q_0, 0^i)$ for $i \geq 0$. Then p_i is ultimately periodic with period $\leq n$ and preperiod at most $n - 1$. Thus $p_{n-1} = p_{n-1+\mathrm{lcm}(1,2,\ldots,n)}$. Since, from the prime number theorem, we have $\mathrm{lcm}(1, 2, \ldots, n) = e^{n(1+o(1))}$, the $\Omega(\log n)$ lower bound follows. □

Example 1. Suppose $|w| = 22$ and $|x| = 52$. Then $|w| \equiv 1 \pmod 7$ and $|x| \equiv 3 \pmod 7$. So we can accept w and reject x with a DFA that uses a cycle of size 7, as follows:



In what follows, then, we only consider the case of equal-length words, and we redefine $S(n) = \max_{\substack{w \neq x \\ |w|=|x|=n}} \text{sep}(w, x)$. The goal of the paper is to survey what is known and unknown, and to examine some variations on the original problem. Our main new results are Theorems 2 and 3.

Notation: in what follows, if x is a word, we let $x[j]$ denote the j'th symbol of x (so that $x[1]$ is the first symbol).

2 Independence of alphabet size

As we have defined it, $S(n)$ could conceivably depend on the size of the alphabet Σ. Let $S_k(n)$ be the maximum number of states needed to separate two length-n words over an alphabet of size k. Then we might have a different value $S_k(n)$ depending on $k = |\Sigma|$. The following result shows this is not the case for $k \geq 2$. This result was stated in [5] without proof; we supply a proof here.

Proposition 2. *For all $k \geq 2$ we have $S_k(n) = S_2(n)$.*

Proof. Suppose x, y are distinct length-n words over an alphabet Σ of size $k > 2$. Then x and y must differ in some position, say for $a \neq b$,

$$x = x' \; a \; x'',$$
$$y = y' \; b \; y'',$$

for $|x'| = |y'|$.

Now map a to 0, b to 1 and map all other letters of Σ to 0. This gives two new distinct binary words X and Y of length n. If X and Y can be separated by an m-state DFA, then so can x and y, by renaming transitions of the DFA to be over $\Sigma \backslash b$ and $\{b\}$ instead of 0 and 1, respectively. Thus $S_k(n) \leq S_2(n)$. But clearly $S_2(n) \leq S_k(n)$, since every binary word can be considered as a word over the larger alphabet Σ. So $S_k(n) = S_2(n)$. □

3 Average case

One frustrating aspect of the separating words problem is that nearly all pairs of words can be easily separated. This means that bad examples cannot be easily produced by random search.

Proposition 3. *Consider a pair of words (w, x) selected uniformly from the set of all pairs of unequal words of length n over an alphabet of size k. Then the expected number of states needed to separate w from x is $O(1)$.*

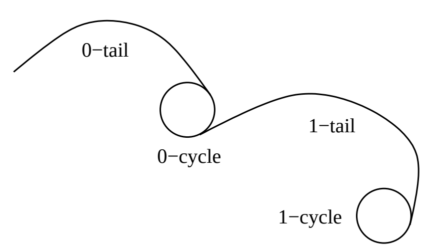
Proof. With probability $1-1/k$, two randomly-chosen words will differ in the first position, which can be detected by an automaton with 3 states. With probability $(1/k)(1-1/k)$ the words will agree in the first position, but differ in the second, etc. Hence the expected number of states needed to distinguish two randomly-chosen words is bounded by $\sum_{i \geq 1}(i+2)(1/k)^{i-1}(1-1/k) = (3k-2)/(k-1) \leq 4$. □

4 Lower bounds for words of equal length

First of all, there is a lower bound analogous to that in Proposition 1 for words of *equal* length. This does not appear to have been known previously.

Theorem 1. *No DFA of at most n states can separate the equal-length binary words $w = 0^{n-1}1^{n-1+\text{lcm}(1,2,\ldots,n)}$ and $x = 0^{n-1+\text{lcm}(1,2,\ldots,n)}1^{n-1}$.*

Proof. In pictures, we have



More formally, let M be any DFA with n states, let q be any state, and let a be any letter. Let $p_i = \delta(q, a^i)$ for $i \geq 0$. Then p_i is ultimately periodic with period $\leq n$ and preperiod ("tail") at most $n-1$. Thus $p_{n-1} = p_{n-1+\mathrm{lcm}(1,2,\ldots,n)}$.

It follows that after processing 0^{n-1} and $0^{n-1+\mathrm{lcm}(1,2,\ldots,n)}$, M must be in the same state. Similarly, after processing

$$0^{n-1}1^{n-1+\mathrm{lcm}(1,2,\ldots,n)} \quad \text{and} \quad 0^{n-1+\mathrm{lcm}(1,2,\ldots,n)}1^{n-1},$$

M must be in the same state. So no n-state machine can separate w from x. □

We now prove a series of very simple results showing that if w and x differ in some "easy-to-detect" way, then $\mathrm{sep}(w, x)$ is small.

4.1 Differences near the beginning or end of words

Proposition 4. *Suppose w and x are words that differ in some symbol that occurs d positions from the start. Then* $\mathrm{sep}(w, x) \leq d + 2$.

Proof. Let t be a prefix of length d of w. Then t is not a prefix of x. We can accept the language $t\Sigma^*$ using $d + 2$ states; such an automaton accepts w and rejects x. □

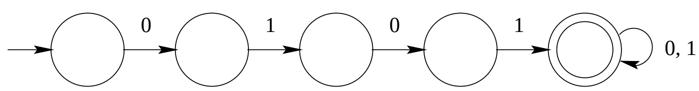
For example, to separate

01010011101100110000

from

01001111101011100101

we can build a DFA to recognize words that begin with 0101:



(Transitions to a dead state are omitted.)

Proposition 5. *Suppose w and x differ in some symbol that occurs d positions from the end. Then* $\mathrm{sep}(w, x) \leq d + 1$.

Proof. Let the DFA M be the usual pattern-recognizing automaton for the length-d suffix s of w, ending in an accepting state if the suffix is recognized. Then M accepts w but rejects x. States of M correspond to prefixes of s, and $\delta(t, a) =$ the longest suffix of ta that is a prefix of s. □

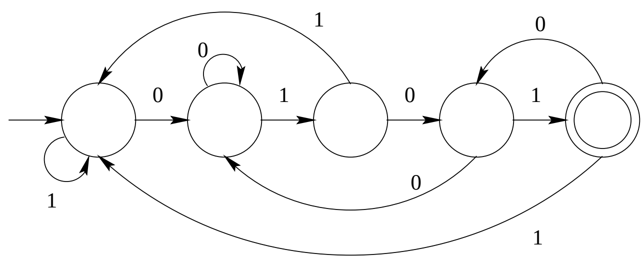
For example, to separate

$$1111101001100101010 1$$

from

$$1111101101001010110 1$$

we can build a DFA to recognize those words that end in 0101:



4.2 Fingerprints

Define $|w|_a$ as the number of occurrences of the symbol a in the word w.

Proposition 6. *If $|w|, |x| \leq n$ and $|w|_a \neq |x|_a$ for some symbol a, then* $\operatorname{sep}(w, x) = O(\log n)$.

Proof. By the prime number theorem, if $|w|, |x| = n$, and w and x have k and m occurrences of a respectively ($k \neq m$), then there is a prime $p = O(\log n)$ such that $k \not\equiv m \pmod{p}$. So we can separate w from x just by counting the number of a's, modulo p. □

Analogously, we have the following result.

Proposition 7. *If there is a pattern of length d occurring a different number of times in w and x, with $|w|, |x| \leq n$, then* $\operatorname{sep}(w, x) = O(d \log n)$.

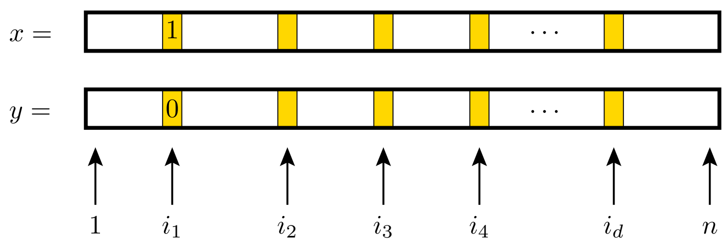
4.3 Pairs with low Hamming distance

The previous results have shown that if w and x have differing "fingerprints", then they are easy to separate. By contrast, the next result shows that if w and x are very similar, then they are also easy to separate.

The *Hamming distance* $H(w, x)$ between two equal-length words w and x is defined to be the number of positions where they differ.

Theorem 2. *Let w and x be words of length n. If $H(w, x) \leq d$, then* $\operatorname{sep}(w, x) = O(d \log n)$.

Proof. Without loss of generality, assume x and y are binary words, and x has a 1 in some position where y has a 0. Consider the following picture:



Let $i_1 < i_2 < \ldots < i_d$ be the positions where x and y differ. Now consider $N = (i_2 - i_1)(i_3 - i_1) \cdots (i_d - i_1)$. Then $N < n^{d-1}$. By the prime number theorem, there exists some prime $p = O(\log N) = O(d \log n)$ such that N is not divisible by p. So $i_j \not\equiv i_1 \pmod{p}$ for $2 \leq j \leq d$.

Define $a_{p,k}(x) = \left(\sum_{j \equiv k \pmod{p}} x[j] \right) \bmod 2$. This value can be calculated by a DFA consisting of two connected rings of p states each. We use such a DFA calculating a_{p,i_1}. Since p is not a factor of N, none of the positions $i_2, i_3, \ldots, i_d$ are included in the count a_{p,i_1}, and the two words x and y agree in all other positions. So x contains exactly one more 1 in these positions than y does, and hence we can separate the two words using $O(d \log n)$ states. □

5 Special classes of words

5.1 Reversals

It is natural to think that pairs of words that are related might be easier to separate than arbitrary words; for example, it might be easy to separate a word from its reversal. No better upper bound is known for this special case. However, we still have a lower bound of $\Omega(\log n)$ for this restricted problem:

Proposition 8. *There exists a class of words w for which* $\mathrm{sep}(w, w^R) = \Omega(\log n)$ *where $n = |w|$.*

Proof. Consider separating

$$w = 0^{t-1} 1 0^{t-1+\mathrm{lcm}(1,2,\ldots t)}$$

from

$$w^R = 0^{t-1+\mathrm{lcm}(1,2,\ldots t)} 1 0^{t-1}.$$

Then, as before, no DFA with $\leq t$ states can separate w from w^R. □

Must $\text{sep}(w^R, x^R) = \text{sep}(w, x)$? No, for $w = 1000$, $x = 0010$, we have

$$\text{sep}(w, x) = 3$$

but

$$\text{sep}(w^R, x^R) = 2.$$

Open Problem 1 *Is* $\left|\text{sep}(x, w) - \text{sep}(x^R, w^R)\right|$ *unbounded?*

5.2 Conjugates

Two words w, w' are *conjugates* if one is a cyclic shift of the other. For example, the English words `enlist` and `listen` are conjugates. Is the separating words problem any easier if restricted to pairs of conjugates?

Proposition 9. *There exist a infinite class of pairs of words* w, x *such that* w, x *are conjugates, and* $\text{sep}(w, x) = \Omega(\log n)$ *for* $|w| = |x| = n$.

Proof. Consider again

$$w = 0^{t-1} 1 0^{t-1+\text{lcm}(1,2,\ldots t)} 1$$

and

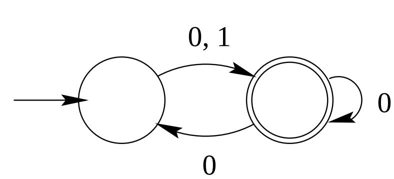
$$w' = 0^{t-1+\text{lcm}(1,2,\ldots t)} 1 0^{t-1} 1.$$

□

6 Nondeterministic separation

We can define $\text{nsep}(w, x)$ in analogy with sep: the number of states in the smallest NFA accepting w but rejecting x. There do not seem to be any published results about this measure.

Now there is an asymmetry in the inputs: $\text{nsep}(w, x)$ need not equal $\text{nsep}(x, w)$. For example, the following 2-state NFA accepts $w = 000100$ and rejects $x = 010000$, so $\text{nsep}(w, x) \leq 2$.



However, an easy computation shows that there is no 2-state NFA accepting x and rejecting w, so $\text{nsep}(x, w) \geq 3$.

Open Problem 2 *Is* $|\text{nsep}(x, w) - \text{nsep}(w, x)|$ *unbounded?*

A natural question is whether NFAs give more separation power than DFAs. Indeed they do, since $\text{sep}(0001, 0111) = 3$ but $\text{nsep}(0001, 0111) = 2$. However, a more interesting question is the *extent* to which nondeterminism helps with separation — for example, whether it contributes only a constant factor or there is any asymptotic improvement in the number of states required.

Theorem 3. *The quantity* $\text{sep}(w, x)/\text{nsep}(w, x)$ *is unbounded.*

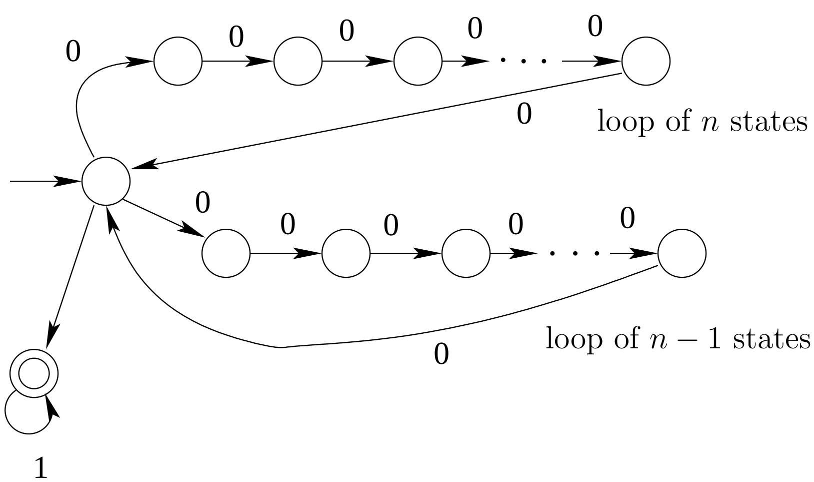
Proof. Consider once again the words

$$w = 0^{t-1+\text{lcm}(1,2,\ldots,t)}1^{t-1} \quad \text{and} \quad x = 0^{t-1}1^{t-1+\text{lcm}(1,2,\ldots,t)}$$

where $t = n^2 - 3n + 2$, $n \geq 4$.

We know from Theorem 1 that any DFA separating these words must have at least $t + 1 = n^2 - 3n + 3$ states.

Now consider the following NFA M:



The language accepted by this NFA is $\{0^a : a \in A\}1^*$, where A is the set of all integers representable by a non-negative integer linear combination of n and $n - 1$. But $t - 1 = n^2 - 3n + 1 \notin A$, as can be seen by computing $t - 1$ modulo $n - 1$ and modulo n. On the other hand, every integer $\geq t$ is in A. Hence $w = 0^{t-1+\text{lcm}(1,2,\ldots,t)}1^{t-1}$ is accepted by M but $x = 0^{t-1}1^{t-1+\text{lcm}(1,2,\ldots,t)}$ is not.

Now M has $2n = \Theta(\sqrt{t})$ states, so $\text{sep}(x, w)/\text{nsep}(x, w) = \Omega(\sqrt{t}) = \Omega(\sqrt{\log |x|})$, which is unbounded. □

Open Problem 3 *Find better bounds on* $\text{sep}(w, x)/\text{nsep}(w, x)$*.*

We can also get an $\Omega(\log n)$ lower bound for nondeterministic separation.

Theorem 4. *No NFA of n states can separate*

$$0^{n^2-1}1^{n^2-1+\mathrm{lcm}(1,2,\ldots,n)}$$

from

$$0^{n^2-1+\mathrm{lcm}(1,2,\ldots,n)}1^{n^2-1}.$$

Proof. A result of Chrobak [1], as corrected by To [11], states that every unary n-state NFA is equivalent to one consisting of a "tail" of at most $O(n^2)$ states, followed by a single nondeterministic state that leads to a set of cycles, each of which has at most n states. The size of the tail was proved to be at most $n^2 - 2$ by Geffert [3].

Now we use the same argument as for DFAs above. □

Open Problem 4 *Find better bounds on* $\mathrm{nsep}(w, x)$ *for* $|w| = |x| = n$*, as a function of n.*

Theorem 5. *We have* $\mathrm{nsep}(w, x) = \mathrm{nsep}(w^R, x^R)$*.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA with the smallest number of states accepting w and rejecting x. Now create a new NFA M' with initial state equal to any single state in $\delta(q_0, w) \cap F$ and final state q_0, and all other transitions of M reversed. Then M' accepts w^R. But M' rejects x^R. For if M' accepted x^R then M would also accept x, since the input string and transitions are reversed. □

7 Separation by 2DPDA's

In [2], the authors showed that words can be separated with small context-free grammars (and hence small PDA's). In this section we observe

Proposition 10. *Two distinct words of length n can be separated by a 2DPDA of size $O(\log n)$.*

Proof. Recall that a 2DPDA is a deterministic pushdown automaton, with endmarkers surrounding the input, and two-way access to the input tape. Given distinct strings w, x of length n, they must differ in some position p with $1 \leq p \leq n$. Using $O(\log p)$ states, we can reach position p on the input tape and accept if (say) the corresponding character equals $w[p]$, and reject otherwise.

Here is how to access position p of the input. We show how to go from scanning position i to position $2i$ using a constant number of states: we move left on the input, pushing two symbols per move on the stack, until the left endmarker is reached. Now we move right, popping one symbol per move, until the initial stack symbol is reached. Using this as a subroutine, and applying it to the binary expansion of p, we can, using $O(\log p)$ states, reach position p of the input. □

8 Permutation automata

We conclude by relating the separating words problem to a natural problem of algebra.

Instead of arbitrary automata, we could restrict our attention to automata where each letter induces a permutation of the states ("permutation automata"), as suggested by Robson [8]. He obtained an $O(n^{1/2})$ upper bound in this case.

For an n-state automaton, the action of each letter can be viewed as an element of $\mathcal{S}_n$, the symmetric group on n elements.

Turning the problem around, then, we could ask: what is the shortest pair of distinct equal-length binary words w, x, such that for all morphisms $\sigma : \{0,1\}^* \to \mathcal{S}_n$ we have $\sigma(w) = \sigma(x)$? Although one might suspect that the answer is $\text{lcm}(1, 2, \ldots, n)$, for $n = 4$, there is a shorter pair (of length 11): 00000011011 and 11011000000.

Now if $\sigma(w) = \sigma(x)$ for all σ, then (if we define $\sigma(x^{-1}) = \sigma(x)^{-1}$) we have that $\sigma(wx^{-1}) =$ the identity permutation for all σ.

Call any nonempty word y over the letters $0, 1, 0^{-1}, 1^{-1}$ an *identical relation* if $\sigma(y) =$ the identity for all morphisms σ. We say y is *nontrivial* if y contains no occurrences of 00^{-1} and 11^{-1}.

What is the length ℓ of the shortest nontrivial identical relation over $\mathcal{S}_n$? Recently Gimadeev and Vyalyi [4] proved $\ell = 2^{O(\sqrt{n}\log n)}$.

9 Acknowledgments

Thanks to Martin Demaine and Anna Lubiw, who participated in the open problem session at MIT where these problems were discussed.

We thank the DCFS 2011 referees for their helpful suggestions.

References

1. M. Chrobak. Finite automata and unary languages. *Theoret. Comput. Sci.*, 47:149–158, 1986. Erratum in 302:497–498, 2003.
2. J. Currie, H. Petersen, J. M. Robson, and J. Shallit. Separating words with small grammars. *J. Automata, Languages, and Combinatorics*, 4:101–110, 1999.
3. V. Geffert. Magic numbers in the state hierarchy of finite automata. *Inform. Comput.*, 205:1652–1670, 2007.
4. R. A. Gimadeev and M. N. Vyalyi. Identical relations in symmetric groups and separating words with reversible automata. In F. M. Ablayev and E. W. Mayr, editors, *5th Int'l Computer Science Symposium in Russia, CSR 2010*, volume 6072 of *Lecture Notes in Computer Science*, pages 144–155. Springer-Verlag, 2010.
5. P. Goralčík and V. Koubek. On discerning words by automata. In L. Kott, editor, *Proc. 13th Int'l Conf. on Automata, Languages, and Programming (ICALP)*, volume 226 of *Lecture Notes in Computer Science*, pages 116–122. Springer-Verlag, 1986.
6. A. Nozaki. Equivalence problem of non-deterministic finite automata. *J. Comput. System Sci.*, 18:8–17, 1979.

7. J. M. Robson. Separating strings with small automata. *Inform. Process. Lett.*, 30:209–214, 1989.
8. J. M. Robson. Separating words with machines and groups. *RAIRO Inform. Théor. App.*, 30:81–86, 1996.
9. J. Shallit. *A Second Course in Formal Languages and Automata Theory.* Cambridge University Press, 2009.
10. J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. *J. Comput. System Sci.*, 53:10–25, 1996.
11. A. W. To. Unary finite automata vs. arithmetic progressions. *Inform. Process. Lett.*, 109:1010–1014, 2009.