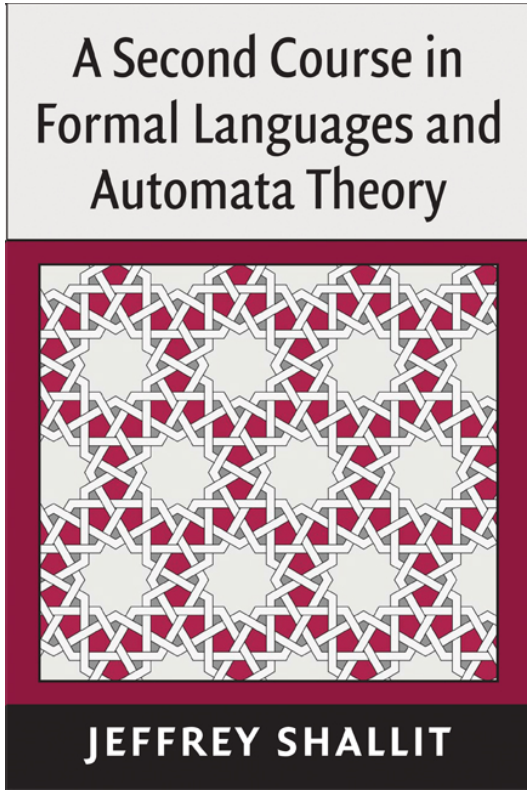
Open Problems in Automata Theory and Formal Languages

Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@cs.uwaterloo.ca
http://www.cs.uwaterloo.ca/~shallit

An Advertisement



Just out from Cambridge University Press! Order your copy today!

Separating Words with Automata

Motivation: a classical problem from the early days of automata theory:

Given two DFA's 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 string accepted by one but not the other?

- The cross-product construction gives an upper bound of $mn - 1$
- But an upper bound of $m + n - 2$ follows from the usual algorithm for minimizing automata
- Furthermore, this bound is best possible.
- For NFA's the bound is exponential in m and n

Separating Words with Automata

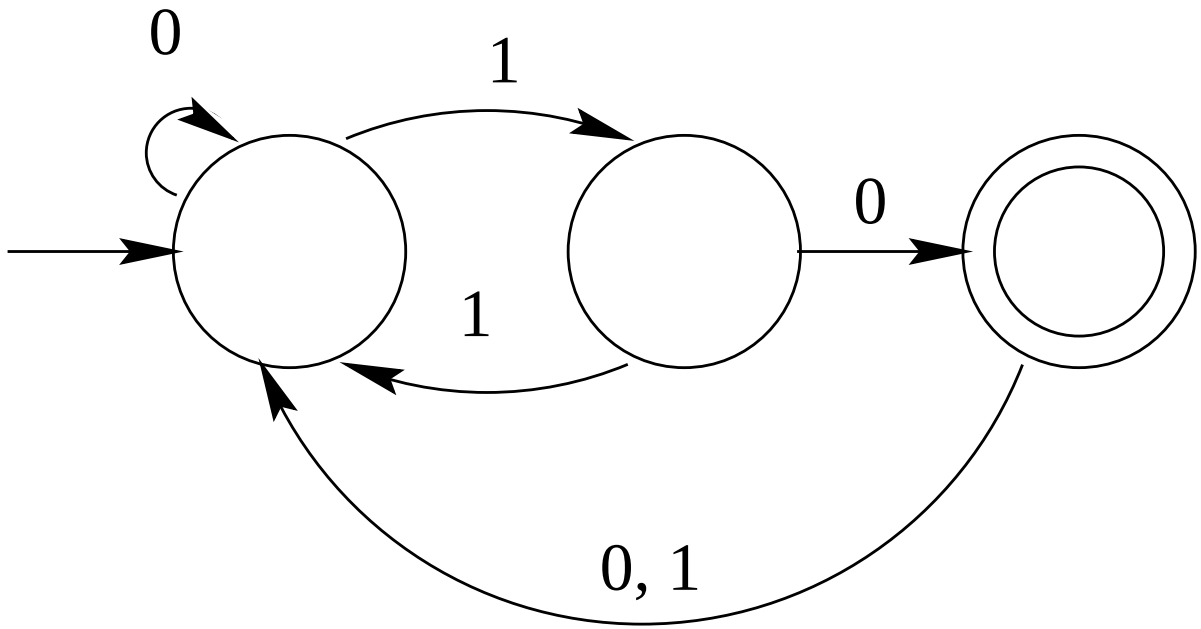
We now look at the inverse problem: given two distinct words, how big an automaton do we need to separate them?

That is, given two words w and x of length $\leq n$, what is the smallest number of states in any DFA that accepts one but not the other?

Call this number $\mathrm{sep}(w, x)$.

A machine M **separates** the word w from the word x if M accepts w and rejects x, or vice versa.

For example, the machine below separates 1000 from 0010.



However, no 2-state DFA can separate these two strings. So $\mathrm{sep}(1000, 0010) = 3$.

Separating Words with Automata

Easy case: if the two words are of different lengths, then we can separate them with a DFA of size $O(\log n)$.

For by the prime number theorem, if $x \neq y$, and $x, y \leq n$ then there is a prime $p = O(\log n)$ such that $x \not\equiv y \pmod{p}$.

So we can accept one string and reject the other by using a cycle mod p, and the appropriate residue class.

Separating Words with Automata

A similar idea works if the strings have a different number of 1's, or if the 1's are in different positions, or if the number of occurrences of a short subword is different, etc.

Separating Words With Automata

- Let

$$S(n) := \max_{\substack{|w|=|x|=n \\ w \neq x}} \operatorname{sep}(w, x),$$

  the smallest number of states required to separate any two strings of length n.

- The separation problem was first studied by Goralcik and Koubek 1986, who proved $S(n) = o(n)$.

- In 1989 Robson who obtained the best known bound: $S(n) = O(n^{2/5}(\log n)^{3/5})$.

Separating Words with Automata

For equal-length strings, $S(n)$ doesn't depend on alphabet size (provided it is at least 2).

Suppose x, y are distinct strings of length n an alphabet Σ of size > 2.

Then they must differ in some position, say

$$\begin{aligned} x &= x' \; a \; x'' \\ y &= y' \; b \; y'' \end{aligned}$$

for $a \neq b$.

Map a to 0, b to 1 and assign all other letters arbitrarily to either 0 or 1. This gives two new distinct strings X and Y of the same length. If X and Y can be separated by an m-state DFA, then so can x and y, by renaming transitions to be over Σ instead of 0 and 1.

Separating Words With Automata

A weaker result:

Theorem (Robson, 1996). We can separate words by computing the parity of the number of 1's occurring in positions congruent to $i \pmod{j}$, for $i, j = O(\sqrt{n})$.

This gives the bound $S(n) = O(n^{1/2})$.

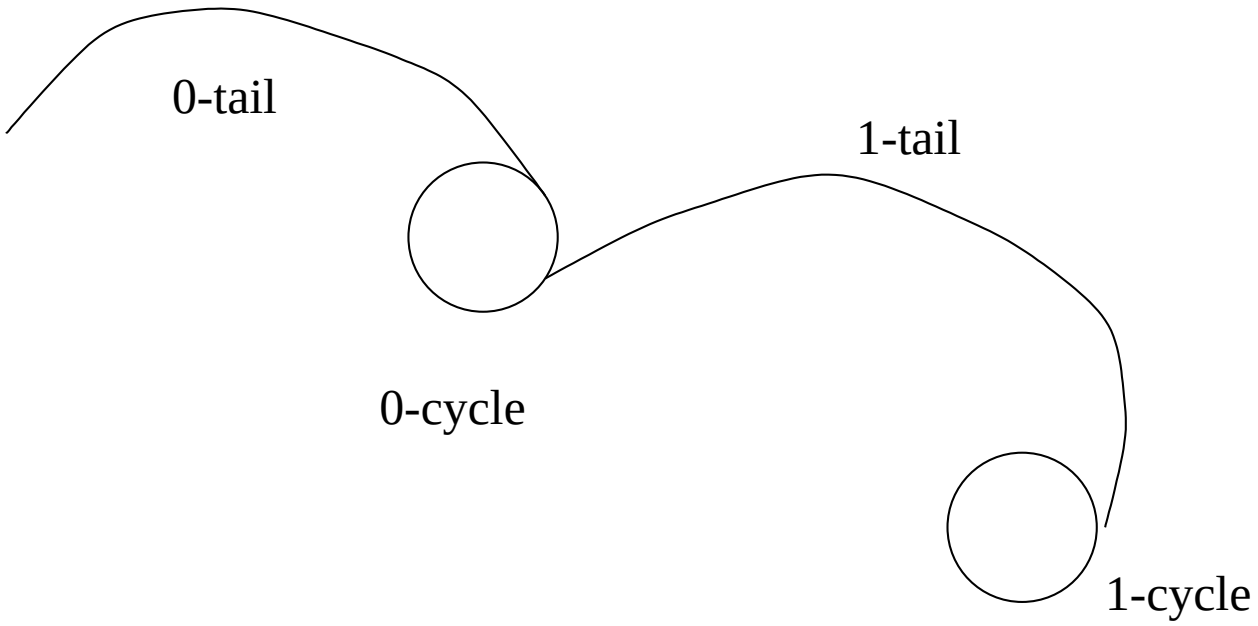
Open Problem 1: Improve Robson's bound of $O(n^{2/5}(\log n)^{3/5})$ on $S(n)$.

Separating Words With Automata: Lower Bound

- Claim: $S(n) = \Omega(\log n)$.
- To see this, consider the two strings

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

Proof in pictures:



So no t-state machine can distinguish these strings.

Since $\operatorname{lcm}(1, 2, \ldots, t) = e^{t+o(t)}$ by the prime number theorem, the lower bound $S(n) = \Omega(\log n)$ follows.

Separating Words With Automata

Some data:

n	$S(n)$	n	$S(n)$
1	2	10	4
2	2	11	4
3	2	12	4
4	3	13	4
5	3	14	4
6	3	15	4
7	3	16	4
8	3	17	4
9	3	18	5

Variations on Separating Words

- Separation by context-free grammars; count number of productions
- Problem: right-hand sides can be arbitrarily complicated
- Solution: Use CFG's in Chomsky normal form (CNF), where all productions are of the form $A \to BC$ or $A \to a$.

Variations on Separating Words

- In 1999 Currie, Petersen, Robson and JOS proved:
  - If $|w| \neq |x|$ then there is a CFG in CNF with $O(\log \log n)$ productions separating w from x. Furthermore, this bound is optimal.
  - If $|w| = |x|$ there is a CFG in CNF with $O(\log n)$ productions separating w from x. There is a lower bound of $\Omega(\frac{\log n}{\log \log n})$.

Open Problem 2: Find matching upper and lower bounds in the case $|w| = |x|$.

More Variations on Separating Words

- Separation by NFA. Do NFA give more power?
  Yes,

$$\mathrm{sep}(0001, 0111) = 3$$

  but

$$\mathrm{nsep}(0001, 0111) = 2.$$

More Variations on Separating Words

Is

$$\mathrm{sep}(x, w)/\mathrm{nsep}(x, w)$$

unbounded?

Yes.

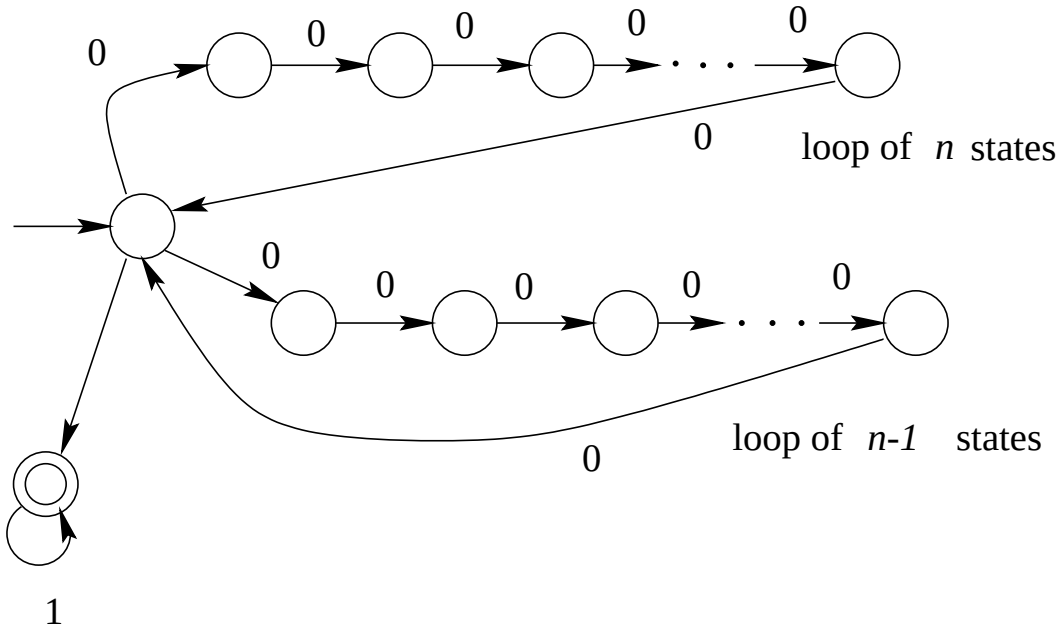
Consider once again the strings

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

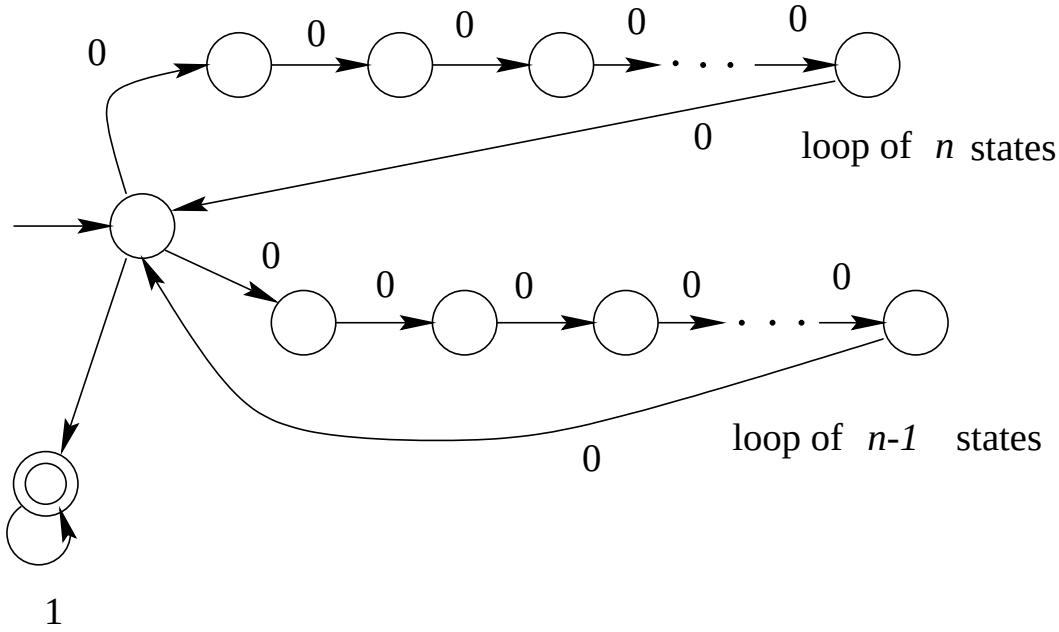
where $t = n^2 - 3n + 2$, $n \geq 4$.

We know from before that any DFA separating these strings 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 $t - 1 = n^2 - 3n + 1 \notin A$.

On the other hand, every integer $\geq t$ is in A. Hence
$w = 0^{t-1+\mathrm{lcm}(1,2,\ldots,t)}1^{t-1}$ is accepted by M but
$x = 0^{t-1}1^{t-1+\mathrm{lcm}(1,2,\ldots,t)}$ is not.

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

More Variations on Separating Words

Open Problem 3: Find good bounds on $\mathrm{nsep}(w, x)$ for $|w| = |x| = n$, as a function of n.

Open Problem 4: Find good bounds on $\mathrm{sep}(w, x)/\mathrm{nsep}(w, x)$.

More Variations on Separating Words

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

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

  but

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

Open Problem 5:
Is

$$\Big|\mathrm{sep}(x, w) - \mathrm{sep}(x^R, w^R)\Big|$$

unbounded?

More Variations on Separating Words

- Two words are *conjugates* if one is a cyclic shift of the other.
- Is the separating words problem any easier if restricted to pairs of conjugates?

Another Kind of Separation

Suppose you have regular languages R_1, R_2 with $R_1 \subseteq R_2$ and $R_2 - R_1$ infinite.

Then it is easy to see that there is a regular language R_3 such that $R_1 \subseteq R_3 \subseteq R_2$ such that $R_2 - R_3$ and $R_3 - R_1$ are both infinite.

This is a kind of topological separation property.

Another Kind of Separation

In 1980, Bucher asked:

Open Problem 6: Is the same true for context-free languages?

That is, given context-free languages L_1, L_2 with $L_1 \subseteq L_2$ and $L_2 - L_1$ infinite, need there be a context-free language L_3 such that $L_1 \subseteq L_3 \subseteq L_2$ such that $L_2 - L_3$ and $L_3 - L_1$ are both infinite?

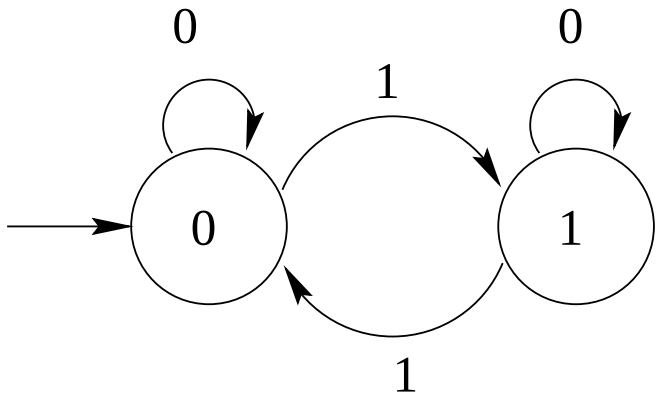
The Thue-Morse Sequence

The Thue-Morse sequence

$$\mathbf{t} = t_0 t_1 t_2 \cdots = 01101001100101101001011 0 \cdots$$

can be described in many ways

- by the recurrence $t_{2n} = t_n$ and $t_{2n+1} = 1 - t_n$
- as the fixed point of the map $0 \to 01$ and $1 \to 10$
- as the sequence generated by the following DFA, where one inputs n expressed in base 2



Runs in The Thue-Morse Sequence

An interesting sequence related to Thue-Morse is the associated sequence **d** of run lengths:

$$\overbrace{\mathbf{t}}^{\mathbf{d}=} = \overbrace{0}^{1}\overbrace{11}^{2}\overbrace{0}^{1}\overbrace{1}^{1}\overbrace{00}^{2}\overbrace{11}^{2}\overbrace{00}^{2}\overbrace{1}^{1}\cdots$$

$$\mathbf{d} = d(0)d(1)d(2)\cdots = 12112221121\cdots .$$

It is not difficult to prove that **d** is the fixed point of the morphism h that sends

$$\begin{aligned} 1 &\rightarrow 121 \\ 2 &\rightarrow 12221. \end{aligned}$$

Runs in The Thue-Morse Sequence

Subsequences of the sequence **d** have some really remarkable properties. For example:

$$\begin{aligned} d(16n+1) &= d(8n+1) \text{ for } 0 \leq n \leq 56 \\ d(32n) &= d(8n) \text{ for } 0 \leq n \leq 14562 \\ d(32n+21) &= d(8n+5) \text{ for } 0 \leq n \leq 29126 \\ d(64n+1) &= d(16n+1) \text{ for } 0 \leq n \leq 1864134, \end{aligned}$$

and the most amazing of all,

$$d(64n) = d(16n) \text{ for } 0 \leq n \leq 119304646.$$

Runs in The Thue-Morse Sequence

However, each of these "identities" fails at the next index.

The reason for this behavior is still somewhat puzzling, although it is certainly related to the fact that the incidence matrix of h has eigenvalues 2 and 1.

Runs in The Thue-Morse Sequence

I can prove

Theorem. For all integers $a > b > 0$, there exists an $n = O(4^{4^{4^a}})$ with $d(4^a n) \neq d(4^b n)$.

But this is almost certainly not best possible.

The proof idea is to use representations in terms of the sequence

$$1, 1, 3, 5, 11, 21, 43, \ldots, \frac{1}{3}(2^n - (-1)^n).$$

Open Problem 7: Determine a good lower bound on

$$\min\{n \ : \ d(4^a n) \neq d(4^b n)\}.$$

This suggests all sorts of similar questions. More generally, given a morphic sequence (generated as a fixed point of a morphism), for how many terms can distinct linear subsequences agree, as a function of the size of the morphism?

Avoidability

- One of the beautiful properties of the Thue-Morse word is that it avoids overlaps
- An **overlap** is a subword (factor) of the form $axaxa$, where a is a single letter, and x is a word
- We'd like to generalize this
- One possible generalization involves Hankel determinants

Avoidability

- An $n \times n$ **Hankel determinant** of a sequence $(a_i)_{i \geq 0}$ is a determinant of a matrix of the form

$$\begin{bmatrix} a_k & a_{k+1} & \cdots & a_{k+n-1} \\ a_{k+1} & a_{k+2} & \cdots & a_{k+n} \\ a_{k+2} & a_{k+3} & \cdots & a_{k+n+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k+n-1} & a_{k+n} & \cdots & a_{k+2n-2} \end{bmatrix}.$$

- If a sequence of real numbers has an overlap

$$\overbrace{a_k}^{a}\overbrace{a_{k+1}\cdots a_{k+n-2}}^{x}\overbrace{a_{k+n-1}}^{a}\overbrace{a_{k+n}\cdots a_{k+2n-3}}^{x}\overbrace{a_{k+2n-2}}^{a}$$

then the corresponding $n \times n$ Hankel matrix has the first and last rows both equal to axa, and so the determinant is 0.

Avoidability

- Is there a sequence on **two** real numbers for which all the Hankel determinants (of all orders) are nonzero?
- No - a simple backtracking argument proves that the longest such sequence is of length 14. For example,

$$1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2.$$

- How about sequences over **three** numbers?

Open Problem 8: Is there a sequence on three real numbers for which all the Hankel determinants are nonzero?

Avoidability

Indeed, a backtracking algorithm easily finds such a sequence on $\{1, 2, 3\}$ with 200 terms.

What is interesting is that this algorithm never had to backtrack!

The sequence begins

$$1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 3, 1, 1, 2, 1, ...$$

Avoidability

Furthermore, the following morphism seems to generate such a sequence on 4 symbols:

$1 \to 12,\ 2 \to 23,\ 3 \to 14,\ 4 \to 32.$

This generates the sequence

$$1223231423141232\cdots .$$

Open Problem 9: Does this sequence have all nonzero Hankel determinants? We have checked up to length 800.

A universality problem

Classical problem: given M, a machine of some type (e.g., DFA, NFA, PDA), decide if $L(M) = \Sigma^*$.

- Unsolvable, if M is a PDA;
- PSPACE-complete, if M is an NFA;
- Solvable in polynomial time if M is a DFA.

A universality problem

A simple variation:

Given M, does there exist an integer $n \geq 0$ such that $\Sigma^n \subseteq L(M)$?

- Still unsolvable for PDA's
- NP-complete for DFA's
- PSPACE-hard for NFA's - but is it in PSPACE? This is **Open Problem 10**.

A universality problem

It would follow that this problem is in PSPACE if we knew that if $\Sigma^r \subseteq L(M)$ for some n, then there always exists a "small" such r (e.g., $r \leq exp(p(n))$ for some polynomial p).

To see this, on input the DFA, we just examine every length l up to the bound r and nondeterministically guess a string of length l (symbol-by-symbol) that *fails* to be accepted. All this can be done in NPSPACE and hence PSPACE by Savitch's theorem.

A related problem

Here is a related problem.

Open Problem 11: what is the complexity of the following problem: given a finite language L, is $\overline{L^*}$ infinite?

If L is represented by a regular expression, this problem is NP-hard.

Another related problem

Open Problem 12: What is the complexity of the following problem: given a finite list of words L over an alphabet Σ, is $\mathrm{Fact}(L^*) = \Sigma^*$?

Here by "Fact" we mean the set of all factors (contiguous subwords).

Similarly, we'd like to find good bounds on the length of the shortest word in $\Sigma^* - \mathrm{Fact}(S^*)$, given that $\mathrm{Fact}(L^*) \neq \Sigma^*$. This is **Open Problem 13**. Currently examples of quadratic length are known, but the best upper bound is doubly-exponential in the length of the longest word in S.

Problems on state complexity

Given a regular language L, we can measure its *state complexity*, the number of states in the smallest DFA accepting L.

We write it as $\mathrm{sc}(L)$.

Maslov (1970) initiated the study of state complexity in the Soviet Union, and since then there have been dozens of papers.

Many papers have focused on state complexity of various transformations.

State complexity of transformations

Let L_1, L_2 be regular languages with state complexity m and n, respectively.

Transformation	Complexity
$L_1 \cap L_2$	mn
$L_1 L_2$	$m \cdot 2^n - 2^{n-1}$
L_1^*	$3 \cdot 2^{n-2}$
L_1^R	2^n

A detour into topology

Consider a set S of real numbers, and let $\mathrm{Cl}(S)$ denote the "closure" under the usual topology (S together with all its accumulation points), and let $\overline{S}$ be the usual complement (with respect to $\mathbb{R}$).

Now consider all sets obtainable by starting from S and applying closure and complement any number of times, in any order.

Kuratowski's "14-theorem" says that this process gives at most 14 distinct sets.

For example,

$$S = [0,1) \cup (1,2] \cup \{3\} \cup [4,5] \cup (\mathbb{Q} \cap (6,7)).$$

State complexity

In analogy with Kuratowski's "14 theorem", one can prove the following:

Theorem. Start with any language L and apply the transformations $L \longrightarrow L^*$ and $L \longrightarrow \overline{L}$ in any order, any number of times. Then at most 14 distinct languages are generated.

We'd like to know the state complexities of these transformation, obtaining good upper and lower bounds.

All cases have been solved, except one: **Open Problem 14**: $L \longrightarrow (\overline{L^*})^*$. What is the state complexity of this transformation? It is potentially doubly-exponential, but is that achievable?

State complexity

Another open problem (**Open Problem 15**): what is the state complexity of the transformation that maps a string to its "boundary", that is, to $L^* \cap (\overline{L})^*$? It is potentially as bad as something like 4^n, but is that achievable?

One More for Dessert: Pierce Expansions

Let $a > b > 0$ be integers. Define $b_0 = b$ and $b_{i+1} = a \bmod b_i$ for $i \geq 0$.

Let $P(a, b)$ be the least index n such that $b_n = 0$.

One More for Dessert: Pierce Expansions

An example with $a = 35$, $b = 22$:

$$\begin{aligned} b_0 &= 22 \\ b_1 &= 35 \bmod 22 = 13 \\ b_2 &= 35 \bmod 13 = 9 \\ b_3 &= 35 \bmod 9 = 8 \\ b_4 &= 35 \bmod 8 = 3 \\ b_5 &= 35 \bmod 3 = 2 \\ b_6 &= 35 \bmod 2 = 1 \\ b_7 &= 35 \bmod 1 = 0. \end{aligned}$$

So $P(35, 22) = 7$.

One More for Dessert: Pierce Expansions

Open Problem 16 : Find good estimates for how big $P(a, b)$ can be, as a function of a.

The problem is interesting because it is related to the so-called "Pierce Expansion" of b/a:

$$\frac{22}{35} = \frac{1}{1}(1 - \frac{1}{2}(1 - \frac{1}{3}(1 - \frac{1}{4}(1 - \frac{1}{11}(1 - \frac{1}{17}(1 - \frac{1}{35})))))).$$

This is called a *Pierce expansion*.

One More for Dessert: Pierce Expansions

It is known that $P(a, b) = O(a^{1/3})$. However, the true behavior is probably $O((\log a)^2)$. There is a lower bound of $\Omega(\log a)$, which can be obtained by choosing

$$a = \operatorname{lcm}(1, 2, \ldots, n) - 1$$

$$b = n.$$

For Further Reading

- J. M. Robson, Separating words with machines and groups, *RAIRO Info. Theor. Appl.* **30** (1996), 81–86.
- J. Currie, H. Petersen, J. M. Robson, and J. Shallit, Separating words with small grammars, *J. Autom. Lang. Combin.* **4** (1999), 101–110.
- N. Rampersad, J. Shallit, Z. Xu, The computational complexity of universality problems for prefixes, suffixes, factors, and subwords of regular languages, `http://arxiv.org/abs/0907.0159`.
- G. Allouche, J.-P. Allouche, and J. Shallit, Kolams indiens, dessins sur le sables aux îles Vanuatu, courbe de Sierpiński, et morphismes de monoïde, *Annales de l'Institut Fourier* **56** (2006), 2115–2130.
- J. A. Brzozowski, E. Grant, J. Shallit, Closures in formal languages and Kuratowski's theorem, in *Developments in Language Theory*, 2009, pp. 125–144.