My Favorite Open Problems in Words

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

Quotes About Problems

"The value of a problem is not so much in coming up with the answer as in the ideas and attempted ideas it forces on the would-be solver."
– Israel Herstein

"If logic is the hygiene of the mathematician, it is not his source of food; the great problems furnish the daily bread on which he thrives."
– André Weil

"The tantalizing and compelling pursuit of mathematical problems offer mental absorption, peace of mind amid endless challenges, repose in activity, battle without conflict, 'refuge from the goading urgency of contingent happenings', and the sort of beauty changeless mountains present to senses tried by the present-day kaleidoscope of events."
– Morris Kline

Quotes About Problems

"It has been said that unsolved problems form the very life of mathematics; certainly they can illuminate and, in the best cases, crystallize and summarize the essence of the difficulties inherent in the various fields."

– Stanislaw Ulam

"As long as a branch of science offers an abundance of problems, so long it is alive; a lack of problems foreshadows extinction or the cessation of independent development."

– David Hilbert

"Mathematics often owes more to those who ask questions than to those who answer them."

– Richard Guy

Criteria for Selection

- Problems should be open
- Problems should be easy to state
- Problems should be capable of generalization

"A mathematical problem should be difficult in order to entice us, yet not completely inaccessible, lest it mock at our efforts. It should be to us a guide post on the mazy paths of hidden truths, and ultimately a reminder of our pleasure in the successful solution."

– David Hilbert

- The **difficulty** of a problem is the log base 2 of the number of days it has been open.

Finite Automata

A **deterministic finite automaton** (DFA) consists of

- a finite nonempty set of states Q
- a finite nonempty input alphabet Σ
- an initial state q_0
- a set of final states $F \subseteq Q$, and
- a *complete* transition function $\delta : Q \times \Sigma \to Q$ that, given a state and a letter, tells what state the machine moves to.

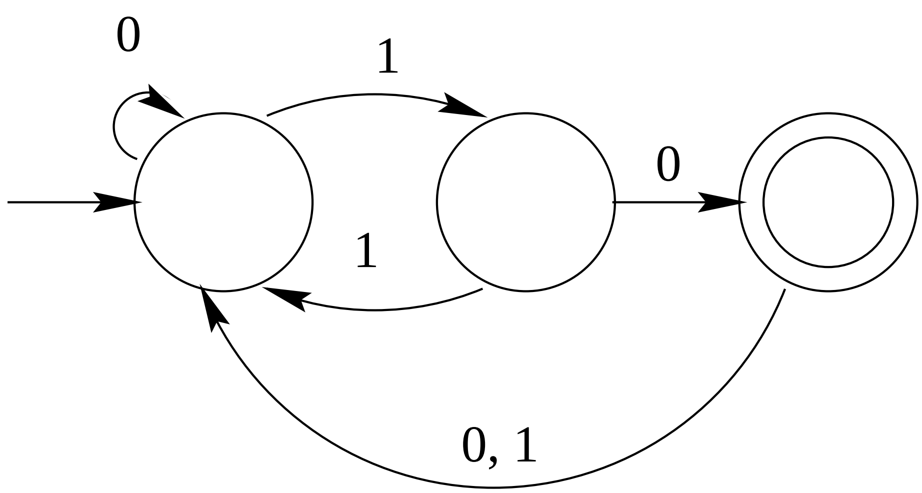
The domain transition function δ can be extended to $Q \times \Sigma^*$ in the usual manner, so that $\delta(q, x)$ tells what state the machine is in after reading x.

A word x is **accepted** if $\delta(q_0, x) \in F$ and **rejected** otherwise.

Separating Words With Automata

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.

Separating Words With Automata

- For words w, x define $\text{sep}(w, x)$ to be the smallest number of states in any DFA separating w from x, or ∞ if $w = x$.
- If $|w| \neq |x|$, can find a DFA with $O(\log n)$ states separating w from x, where $|w|, |x| \leq n$, since there is a prime $p = O(\log n)$ such that

$$|w| \bmod p \neq |x| \bmod p.$$

- Thus we may assume $|w| = |x| = n$.

Separating Words With Automata

- We are interested in a good upper bound on the separation number

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

- First studied by Goralcik and Koubek 1986, who proved $S(n) = o(n)$.
- In 1989 Robson who obtained the best known bound: $S(n) = O(n^{2/5}(\log n)^{3/5})$.

Robson's Proof that $S(n) = O(n^{1/2})$

- Robson gave a simple proof that $S(n) = O(n^{1/2})$.
- We only consider automata of a very special form: the ones that count the parity of the number of 1's in positions congruent to i (mod k), for $0 \leq i < k$ and $1 \leq k \leq r$.
- For each such automaton we only need $O(r)$ states.
- Robson shows that we can take $r = \Theta(\sqrt{n})$.

Open Problem 1 (*Hardness = 12.5*):

Improve Robson's bound on $S(n)$.

Separating Words With Automata

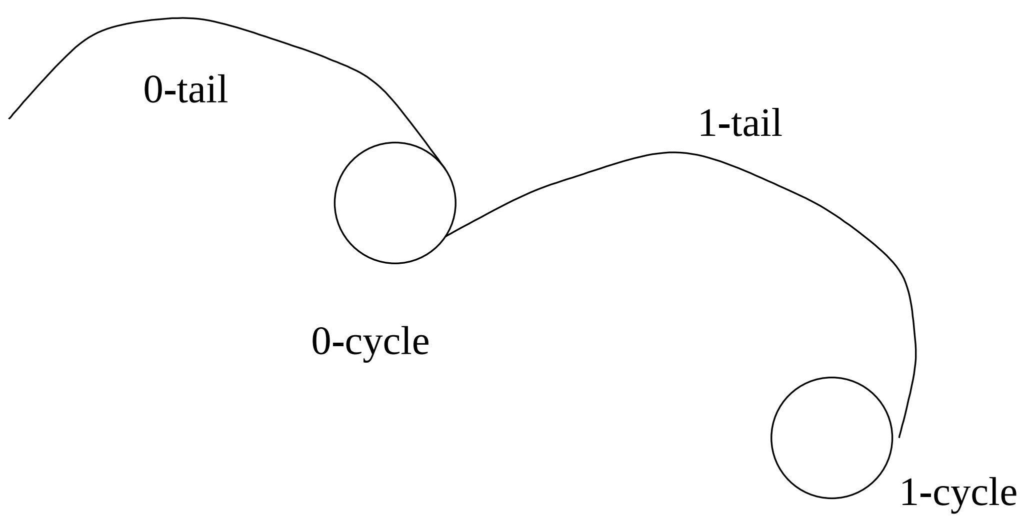
- Claim: $S(n) = \Omega(\log n)$.
- To see this, consider the two strings

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

and

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

Proof in pictures:



So no t-state machine can distinguish these strings.

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

Separating Words With Automata

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

Variations on Separating Words

- Separation by context-free grammars
- A **context-free grammar** is a quadruple $G = (V, \Sigma, P, S)$ where
  - V is a nonempty set of variables
  - Σ is a nonempty alphabet
  - P is a set of productions of the form $A \to \alpha$ for $\alpha \in (V \bigcup \Sigma)^*$
  - S is the start symbol
- The language generated by G is the set of all strings in Σ^* obtainable by starting with S and applying some sequence of productions, where a variable is replaced by a right-hand-side.
- 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 $|w| \not= |x|$ 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 (*Hardness = 11*): 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

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

Is

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

unbounded?

Yes.

Consider once again the strings

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

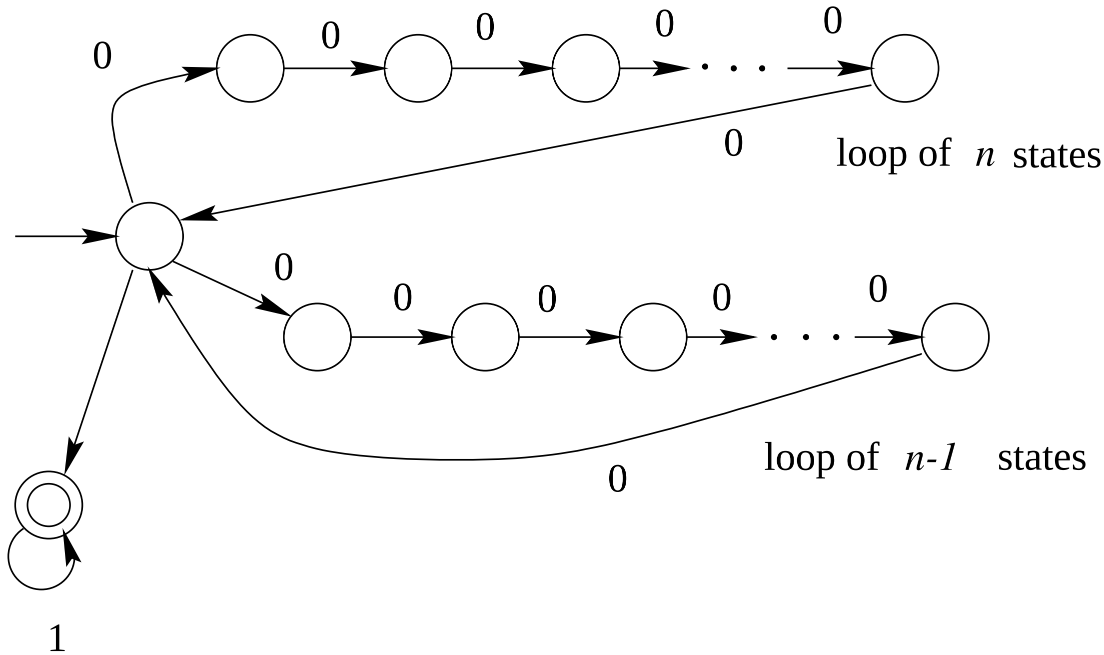
and

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

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

We know that any DFA separating these string smust have at least $t + 1$ 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 sufficiently large integer is in A, since $\gcd(n, n-1) = 1$. Hence w is accepted by M but x is not.

M has $2n = O(\sqrt{t})$ states.

More Variations on Separating Words

Open Problem 3 (*Hardness = 1*): Find good bounds on $\mathrm{nsep}(w, x)$ for $|w| = |x| = n$, as a function of n.

Open Problem 4 (*Hardness = 1*): 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 (*Hardness = 3*): Is

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

unbounded?

More Variations on Separating Words

- Instead of the worst case in separating two words, how about the average case?
- Define
$$\mathrm{SA}(n) = \frac{1}{2^n(2^n-1)} \sum_{\substack{|w|=|x|=n \\ w \neq x}} \mathrm{sep}(w,x).$$

An easy argument proves

Theorem. $\mathrm{SA}(n) = O(\log n)$.

Proof. Let x be fixed. Consider a DFA that accepts all strings whose length-$\lceil \log_2 n \rceil$ prefix agrees with that of x. This DFA separates x from all other strings except $2^{n-\lceil \log_2 n \rceil} \leq 2^n/n$ of them. Since these other strings can be separated using at most $n+1$ states, it follows that

$$\mathrm{SA}(n) \leq (1-\frac{1}{n})O(\log n)+(n+1)\frac{1}{n} = O(\log n).$$

More Variations on Separating Words

- In the limit, $31/36 = 86.1\%$ of all pairs of words can be distinguished by 2-state machines
- Empirically, 99.999% of all pairs of words can be distinguished by 2- or 3-state machines.
- As Julien Cassaigne noted, we can extend the argument of the previous page to prove $\mathrm{SA}(n) = O(1)$, as follows: a 3-state machine can be used to check the first letter, and hence distinguishes at least $1/2$ of all pairs. A 4-state machine can be used to check the first two letters, and hence distinguishes at least $3/4$ of all pairs, etc. Thus the expected number of states is

$$3/2 + 4/4 + 5/8 + 6/16 + \cdots ,$$

which is $O(1)$.

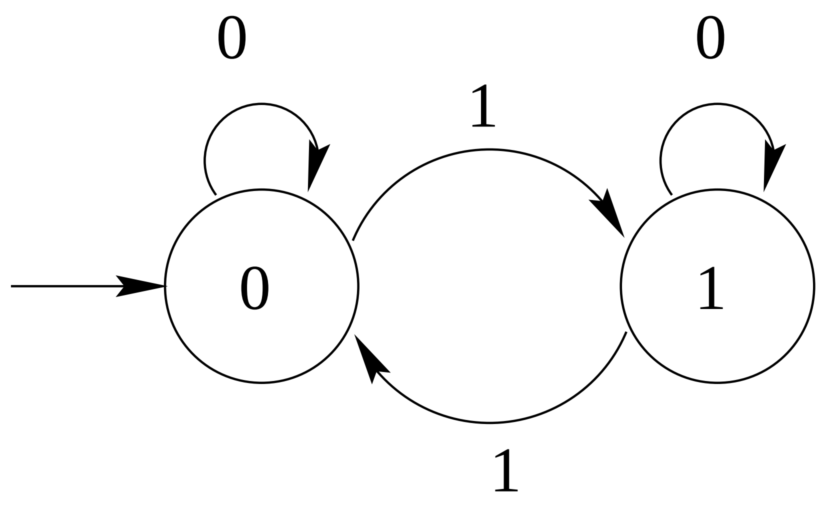
The Thue-Morse Sequence

The Thue-Morse sequence

$\mathbf{t} = t_0 t_1 t_2 \cdots = 011010011001011010010110 \cdots$

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 $0 \rightarrow 01$ and $1 \rightarrow 10$
- 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 $\mathbf{d}$ 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 $\mathbf{d}$ 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 $\mathbf{d}$ have some really remarkable properties. For example:

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

and the most amazing of all,

$$d(64n) = d(16n) \text{ for } 0 \le n \le 119304646,$$

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 6 (*Hardness = 11.5*):
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 subsequences agree, as a function of the size of the morphism?

Another Thue-Morse Problem

- *Subword complexity* is the function that counts the number of distinct length-n subwords (or "factors") in an infinite word.
- Brlek computed the exact subword complexity of the Thue-Morse sequence; it is linear.
- In fact, the subword complexity of any linearly-indexed subsequence is also linear.
- But how about the subsequence $(t_{n^2})_{n \geq 0}$?

> **Open Problem 7** (*Hardness = 9*):
> Is the subword complexity function of $(t_{n^2})_{n \geq 0}$ equal to 2^n?

Another problem is

> **Open Problem 8** (*Hardness = 13*):
> Do there exist infinitely many primes p such that $t_p = 0$?

Another Thue-Morse Problem

- The **multigrades problem** is the following: given n, find disjoint sets of integers S and U such that
$$\sum_{s \in S} s^i = \sum_{u \in U} u^i$$
for $i = 0, 1, 2, \ldots, n-1$.

- Prouhet (1851) discovered that we could take
$$S = \{0 \leq k < 2^n \; : \; t_k = 0\}$$
and
$$U = \{0 \leq k < 2^n \; : \; t_k = 1\}.$$

- For example
$$0^i + 3^i + 5^i + 6^i = 1^i + 2^i + 4^i + 7^i$$
for $i = 0, 1, 2$.

- However, there are other solutions using a partition of the set $\{0, 1, \ldots, 2^n - 1\}$.

Another Thue-Morse Problem

For example:

$\{0, 1, 6, 7, 9, 12, 14, 15, 16, 17, 18, 21, 24, 27, 30, 31, 32$
$33, 36, 39, 42, 45, 46, 47, 48, 49, 51, 54, 56, 57, 62, 63\}$

and

$\{2, 3, 4, 5, 8, 10, 11, 13, 19, 20, 22, 23, 25, 26, 28, 29, 34,$
$35, 37, 38, 40, 41, 43, 44, 50, 52, 53, 55, 58, 59, 60, 61\}$

Open Problem 9 (*Hardness = 9*):
Is the Thue-Morse partition given above the unique partition that solves the multigrade problem and for which $|\sum_{s\in S} s^n - \sum_{u\in U} u^n|$ is minimized?

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 three numbers?

Open Problem 10 (*Hardness* $=$ *9*):
Is there a sequence on three real numbers for which all the Hankel determinants are nonzero?

Indeed, a backtracking algorithm has found such a sequence on $\{1, 2, 3\}$ with 172 terms. What is interesting is that this algorithm never had to backtrack! The sequence begins

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

Furthermore, the following morphism seems to generate such a sequence on 4 symbols:
$1 \to 12$, $2 \to 23$, $3 \to 14$, $4 \to 32$.

Open Problem 11 (*Hardness = 6*):
Does the sequence generated as the fixed point of $1 \to 12$, $2 \to 23$, $3 \to 14$, $4 \to 32$ have all nonzero Hankel determinants?

More on Fixed Points

- There are many problems on fixed points of morphisms that are still open.
- For example, how hard is it to deduce whether a given finite word can be generated as the fixed point of a "small" morphism?

Open Problem 12 (*Hardness = 12*):
What is the computational complexity of the following problem?

Given an alphabet Σ, a string $w \in \Sigma^*$, and an integer n, do there exist an alphabet Δ, a morphism $h : \Delta^* \to \Delta^*$, an integer k, a letter $a \in \Delta$, and a coding $\tau : \Delta \to \Sigma$ such that w is a prefix of $\tau(h^k(a))$ and $|h| + |\tau| < n$?

More on Fixed Points

Given an infinite word $\mathbf{w} = a_1 a_2 a_3 \cdots$ we can consider the language of prefixes

$$L_{\mathbf{w}} = \{\epsilon, a_1, a_1 a_2, a_1 a_2 a_3, \ldots\}.$$

A beautiful theorem of Berstel says that if $\mathbf{w}$ is the fixed point of a morphism, then $L_{\mathbf{w}}$ is a co-CFL.

Open Problem 13 (*Hardness = 14*):
Is $L_{\mathbf{w}}$ a co-CFL when $\mathbf{w}$ is the *image* of the fixed point of a morphism?

The Frobenius Problem

The **depth** of a state in an automaton is the length of the shortest path reaching that state from the start state.

The **radius** of an automaton is the maximum depth over all states.

The **radius** of a regular language L is defined to be the minimum radius over all automata accepting that language.

Formally,

$$\mathrm{rad}(L) = \min_{\substack{L(M)=L \\ M \in \mathrm{DFA}}} \max_{q \in Q} \min_{\substack{x \in \Sigma^* \\ q=\delta(q_0,x)}} |x|.$$

Radius

Similarly we can define the nondeterministic radius as follows:

$$\mathrm{nrad}(L) = \min_{\substack{L(M)=L \\ M \in \mathrm{NFA}}} \max_{q \in Q} \min_{\substack{x \in \Sigma^* \\ q \in \delta(q_0, x)}} |x|.$$

Keith Ellul proved that the radius of a regular language is the radius of its minimal automaton.

He also proved that the ratio $\mathrm{rad}(L)/\mathrm{nrad}(L)$ can be arbitrarily large.

However, we still do not know how to compute $\mathrm{nrad}(L)$:

Open Problem 14 (*Hardness = 10*):
Given a DFA accepting L, is $\mathrm{nrad}(L)$ computable?

Back to Automata

- The Frobenius problem asks: given a set of positive integers

$$\{a_1, a_2, \ldots, a_k\}$$

with

$$a_1 < a_2 < \cdots < a_k$$

and

$$\gcd(a_1, a_2, \ldots, a_k) = 1,$$

find the largest integer $G(a_1, a_2, \ldots, a_k)$ which is *not* a non-negative integer linear combination of the a_i.

- Example: the "chicken nuggets problem": $G(6, 9, 20) = 43$.
- Simple closed form known for $k = 2$
- Fast, simple algorithm for $k = 3$
- An impractical polynomial-time algorithm for fixed k
- NP-hard in the general case

The Frobenius Problem and State Complexity

- The relationship between the Frobenius number and state complexity is as follows: if
$$\gcd(a_1, a_2, \ldots, a_k) = 1,$$
then
$$\mathrm{sc}((0^{a_1}+0^{a_2}+\cdots+0^{a_k})^*) = G(a_1, a_2, \ldots, a_k)+2.$$
- Many bounds for G known; one of the simplest is
$$G(a_1, \ldots, a_k) < a_k^2.$$
- This bound implies that the state complexity of $(0^{a_1}, 0^{a_2}, \ldots, 0^{a_k})^*$ is $O(a_k^2)$.
- Furthermore, this bound is tight, since $G(n-1, n) = n^2 - 3n + 1$.
- What about state complexity for larger alphabets?
- This can be considered a version of the "noncommutative Frobenius problem".

The Non-Commutative Frobenius Problem

Open Problem 15 (*Hardness = 10*):
Find a tight upper bound for

$$\mathrm{sc}((w_1 + w_2 + \cdots + w_k)^*)$$

in terms of

$$n = |w_1| + \cdots + |w_k|.$$

- An upper bound is
$$\mathrm{sc}((w_1 + w_2 + \cdots + w_k)^*) \leq 2^{n-k+1}.$$
- For a long time it was thought that $O(n^2)$ was the "true" bound.
- However, recently I found a class of examples that achieves a state complexity of $2^{\Omega(\sqrt{n})}$.

The Non-Commutative Frobenius Problem

Let t be an integer ≥ 2, and define strings as follows:

$$y := 01^{t-1}0$$
$$x_i := 1^{t-i-1}01^{i+1}, \quad 0 \leq i \leq t-2 \ .$$

Let $S_t := \{0, x_0, x_1, \ldots, x_{t-2}, y\}$.

Thus, for example,

$$S_6 := \{0, 1111101, 1111011, 1110111,$$
$$1101111, 1011111, 0111110\}.$$

Theorem. S_t^* has state complexity $3t2^{t-2} + 2^{t-1}$.

The proof of this theorem is rather complicated, so we just sketch a proof of the following slightly weaker result:

Theorem. $\mathrm{sc}(S_t^*) \geq 2^{t-2}$.

The Non-Commutative Frobenius Problem

Proof. First, we create an NFA M_t with $3t-1$ states that accepts S_t^*. This NFA has states

$$Q = \{p_0, p_1, \ldots, p_t, q_1, q_2, \ldots, q_{t-1}, r_1, r_2, \ldots, r_{t-1}\}$$

with only one final state $F = \{p_0\}$.

For example, here is the NFA M_6.

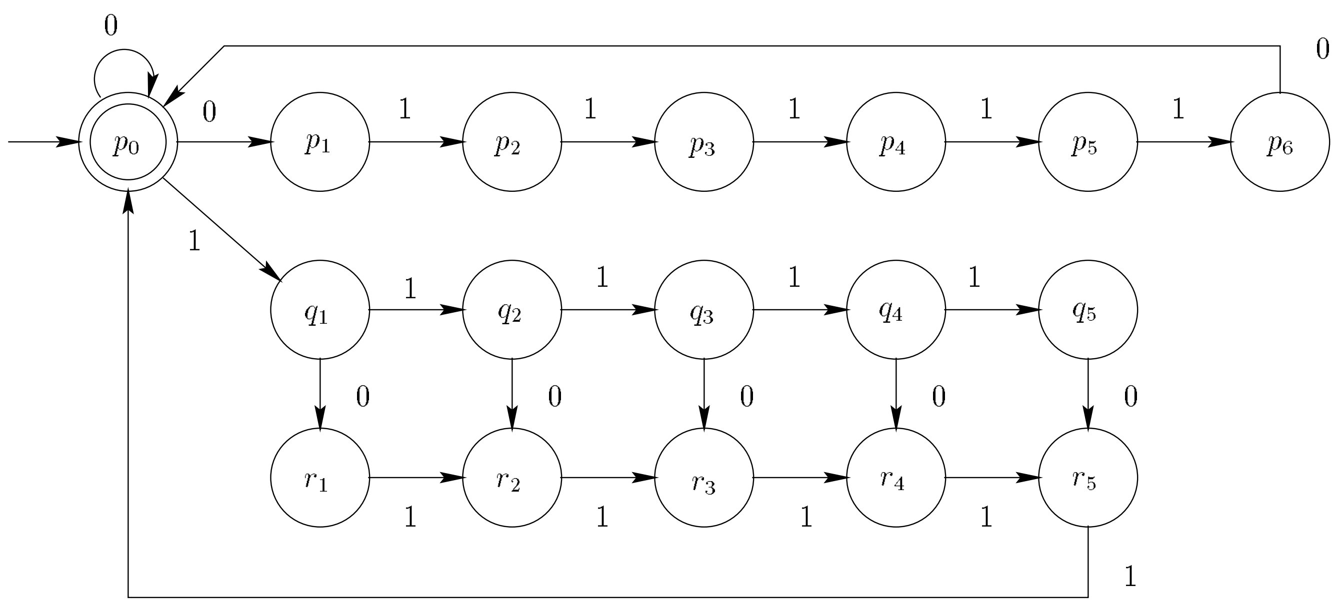


Figure 1: The NFA M_6

The Non-Commutative Frobenius Problem

Let T be any subset of $\{r_1, r_2, \ldots, r_{t-2}\}$, and write $T = \{r_{i_1}, r_{i_2}, \ldots, r_{i_j}\}$ for j indices

$$1 \leq i_1 < i_2 < \cdots < i_j \leq t-2.$$

We claim that the 2^{t-2} strings

$$y \; x_{t-2}y \; x_{t-3}x_{t-2}y \; x_{t-4}x_{t-3}x_{t-2}y$$

$$\cdots \; x_1 x_2 \cdots x_{t-2} y \; x_{i_1} x_{i_2} \cdots x_{i_j} y$$

where

$$1 \leq i_1 < i_2 < \cdots < i_j \leq t-2$$

are pairwise inequivalent under the Myhill-Nerode equivalence relation.

The Non-Commutative Frobenius Problem

To show this, we first argue that any subset of states of the form $T' := \{p_0, r_{t-1}\} \cup T$, where T is as in the previous paragraph, is reachable from p_0. I claim that the following path reaches T':

$$\{p_0\} \xrightarrow{y} \{p_0, r_{t-1}\} \xrightarrow{x_{t-2}y} \{p_0, r_{t-2}, r_{t-1}\}$$
$$\xrightarrow{x_{t-3}x_{t-2}y} \{p_0, r_{t-3}, r_{t-2}, r_{t-1}\}$$
$$\xrightarrow{x_{t-4}x_{t-3}x_{t-2}y} \cdots$$
$$\xrightarrow{x_1 x_2 \cdots x_{t-2}y} \{p_0, r_1, r_2, \ldots, r_{t-1}\}$$
$$\xrightarrow{x_{i_1} x_{i_2} \cdots x_{i_j} y} \{p_0, r_{i_1}, r_{i_2}, \ldots, r_{i_j}, r_{t-1}\}.$$

The Non-Commutative Frobenius Problem

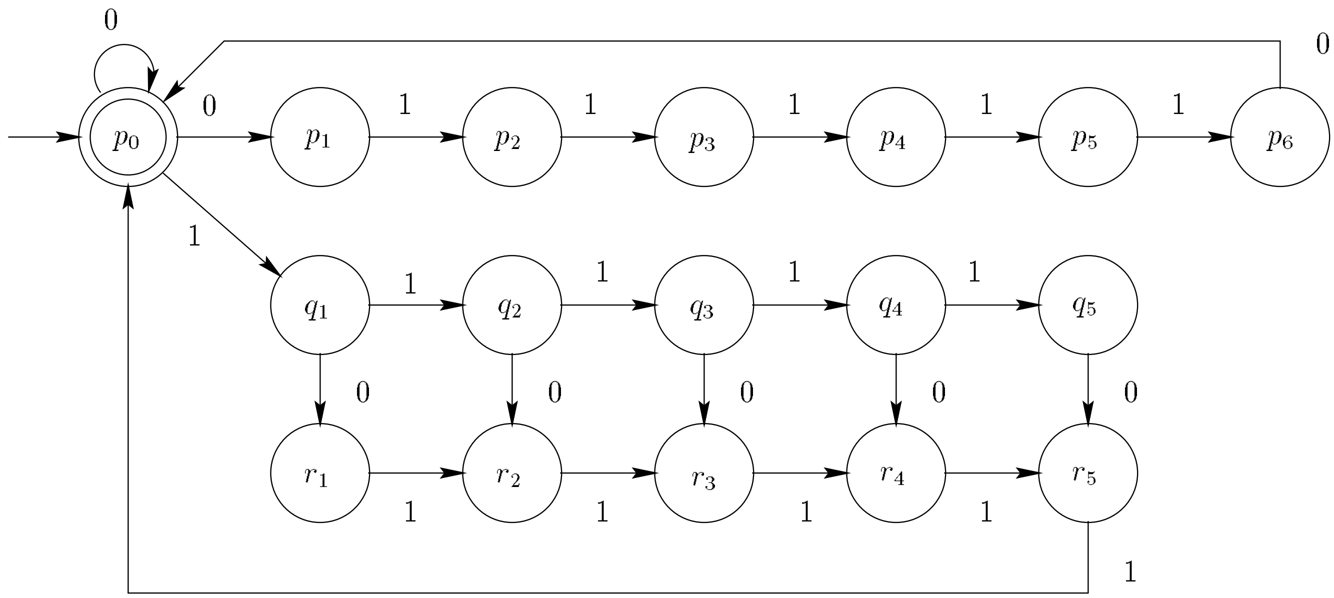


Figure 2: The NFA M_6

Finally, we argue that each of these subsets of states is inequivalent. This is because given two distinct such subsets, say T' and T'', there must be an r_i, $1 \leq i \leq t-2$, that is contained in one (say T') but not the other. Then reading the string 1^{t-i} takes T' to p_0, but not T''.

Since the alphabetic length of the strings in S_t is $n = t^2+t+1$, it now follows that this example has state complexity $2^{\Omega(\sqrt{n})}$.

A Final Problem

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 $b_n = 0$.

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 $P(35, 22) = 7$.

The problem is to estimate how big $P(a, b)$ can be, as a function of a.

Pierce Expansions

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*.

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

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

Open Problem 16 (*Hardness = 13*):
Find good upper and lower bounds on $P(a,b)$.