ELSEVIER

Theoretical Computer Science 180 (1997) 181–201

Theoretical Computer Science

Automaticity II: Descriptional complexity in the unary case

Carl Pomerance [a,1], John Michael Robson [b,2], Jeffrey Shallit [c,*,3]

[a] *Department of Mathematics, University of Georgia, Athens, GA 30601-3024, USA*
[b] *Department of Computer Science, Australian National University, Canberra, ACT 0200, Australia*
[c] *Department of Computer Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1*

Received March 1995
Communicated by M. Nivat

Abstract

Let Σ and Δ be finite alphabets, and let f be a map from Σ^* to Δ. Then the deterministic automaticity of f, $A_f(n)$, is defined to be the size of the minimum finite-state machine that correctly computes f on all inputs of size $\leqslant n$. A similar definition applies to languages L. We denote the nondeterministic analogue (for languages L) of automaticity by $N_L(n)$.

In a previous paper, Shallit and Breitbart examined the properties of this measure of descriptional complexity in the case $|\Sigma| \geqslant 2$. In this paper, we continue the study of automaticity, focusing on the case where $|\Sigma| = 1$.

We prove that $A_f(n) \leqslant n + 1 - \lfloor \log_\ell n \rfloor$, where $\ell = |\Delta|$. We also prove that $A_f(n) > n - 2\log_\ell n - 2\log_\ell \log_\ell n$ for almost all functions f.

In the nondeterministic case, we show that there exists a c such that for almost all unary languages L, we have $N_L(n) > cn/\log n$ for all sufficiently large n. The proof is based on a new enumeration method for languages accepted by unary q-state NFAs.

If L is not a regular language, then it follows from a result of Karp that $\limsup_{n\to\infty} A_L(n)/n \geqslant \frac{1}{2}$. We conjecture that if $L \subseteq 0^*$, then this bound can be improved to $(\sqrt{5}-1)/2$.

Finally, we give some lower bounds for nondeterministic automaticity for nonregular languages.

1. Introduction

In a previous paper [33], the third author and Breitbart examined the notion of automaticity, a fundamental measure of descriptional complexity for functions and

* Corresponding author. E-mail: shallit@uwaterloo.ca.

[1] Research supported in part by NSF Grant DMS 920-6784. E-mail: carl@math.uga.edu.

[2] Present address: LaBRI, Université de Bordeaux I, 351, cours de la Libération, 33405 Talence Cedex, France; e-mail: robson@labri.u-bordeaux.fr.

[3] Research supported in part by a grant from NSERC; partial support under NSF Grant DCR 920-8639 and the Wisconsin Alumni Research Foundation.

PII S0304-3975(96)00189-2

languages defined over finite alphabets Σ. Roughly speaking, the automaticity $A_L(n)$ of a language L is the minimum number of states in any deterministic finite automaton that recognizes L on all strings of length $\leqslant n$.

Automaticity has been studied by many writers, including Karp [22], who proved a "gap" theorem showing that if a language L is not regular, then its deterministic automaticity $A_L(n)$ must infinitely often be at least linear in n. In addition to its evident intrinsic interest, the measure has proved useful in obtaining nontrivial lower bounds. For example, Dwork and Stockmeyer (and Kaneps and Freivalds, independently) used the measure to obtain lower bounds on computation by two-way probabilistic finite automata [13, 14, 20], and Kaneps and Freivalds used it to obtain lower bounds on the space complexity of probabilistic Turing machines [19].

Previous work focused on the case $k = |\Sigma| \geqslant 2$. To date, it appears that no one has made a systematic study of automaticity in the unary case, when $k = |\Sigma| = 1$. This is the case we examine in this paper. We prove upper and lower bounds on the automaticity of languages, and along the way obtain in Theorem 7 the best known upper bound on the number of distinct unary languages accepted by nondeterministic finite automata with q states. We give a conjectured improvement to Karp's bound in the case of unary languages, and show that our conjecture, if correct, is best possible. Finally, we examine the nondeterministic version of automaticity and prove upper and lower bounds.

2. Notation and definitions

We will use the following notation: $\Sigma^{\leqslant n} = \varepsilon + \Sigma + \Sigma^2 + \cdots + \Sigma^n$.

We will be concerned with finite automata that can compute functions. A *deterministic finite automaton* with *output* (DFAO) is a sextuple $M = (Q, \Sigma, \delta, q_0, \Delta, \tau)$, where Q is a finite nonempty set of states, Σ (the input alphabet) and Δ (the output alphabet) are finite nonempty sets, δ is the transition function mapping $Q \times \Sigma$ into Q, q_0 is the initial state, and τ is an output function mapping Q into Δ. We emphasize that δ is *complete*; i.e., it is defined for all members of $Q \times \Sigma$. The machine M computes a function g_M from Σ^* to Δ as follows: $g_M(w) = \tau(\delta(q_0, w))$.

In the case where $\Delta = \{0, 1\}$, this flavor of automaton coincides with the ordinary notion of automaton and acceptance/rejection. In this case we can associate a set of *final states* F such that $F = \{q \in Q : \tau(q) = 1\}$. The language accepted by M is then $L(M) = \{w \in \Sigma^* : \delta(q_0, w) \in F\}$.

By $|M|$ we will mean the "size" of the automaton M, which we define to be the cardinality of the set Q of states in M.

Let Σ and Δ be finite alphabets, and let f be a map from Σ^* to Δ. Then the (*deterministic*) *automaticity* of f is a function $A_f(n)$ defined as follows:

$$A_f(n) = \min\{|M| : M \in \text{DFAO and } \forall w \in \Sigma^{\leqslant n}\ f(w) = g_M(w)\}.$$

Roughly speaking, $A_f(n)$ counts the minimum number of states in any DFAO M that simulates f correctly on all strings of length $\leqslant n$; how M behaves on longer strings is unspecified. In general, there may be many different automata for which the number of states is a minimum.

If $L \subseteq \Sigma^*$ is a language, then we write $A_L(n)$ for the automaticity of the characteristic function $\chi_L(w)$, defined as follows:

$$\chi_L(w) = \begin{cases} 1 & \text{if } w \in L; \\ 0 & \text{otherwise.} \end{cases}$$

In this case,

$$A_L(n) = \min\{|M| : M \in \text{DFA and } L(M) \cap \Sigma^{\leqslant n} = L \cap \Sigma^{\leqslant n}\}.$$

There is also a *nondeterministic* analogue of automaticity $N_L(n)$, which we define only for languages L:

$$N_L(n) = \min\{|M| : M \in \text{NFA and } L(M) \cap \Sigma^{\leqslant n} = L \cap \Sigma^{\leqslant n}\}.$$

We note that our model of nondeterministic finite automaton is that defined in [18], and allows transitions only on single letters and the empty string ε.

We will sometimes use the following terminology. We say that a function $f: \Sigma^* \to \Delta$ is an *nth-order approximation* to a function $g: \Sigma^* \to \Delta$ if $f(w) = g(w)$ for all w with $|w| \leqslant n$. Similarly, we say that a language $L \subseteq \Sigma^*$ is an *nth-order approximation* to a language $L' \subseteq \Sigma^*$ if we have $L \cap \Sigma^{\leqslant n} = L' \cap \Sigma^{\leqslant n}$.

The implied constant in the big-O bounds in this paper may depend on $k = |\Sigma|$ and $\ell = |\Delta|$, but not on n.

3. Properties of automaticity

In this section we recall from [33] some of the properties of deterministic and nondeterministic automaticity.

Theorem 1. *Let $\Sigma = \{0\}$, $f: \Sigma^* \to \Delta$ and $L \subseteq \Sigma^*$. Then:*

(1) *Karp's Theorem: If L is not a regular language, then $A_L(n) \geqslant (n+3)/2$ for infinitely many n.*

(2) *For each $w \in \Sigma^*$ with $|w| \leqslant n$, define $S_w(n) = f(w)f(w0)f(w00)\cdots f(0^{n-|w|})$. Let $\mathcal{S}(n)$ be the collection $\{S_w(n) : w \in \Sigma^{\leqslant n}\}$. For strings in $\mathcal{S}(n)$, define the partial order $\leqslant$ as follows: $x \leqslant y$ if x is a prefix of y. Then $A_f(n)$ equals the cardinality of the set of maximal elements (under $\leqslant$) of $\mathcal{S}(n)$.*

(3) *If $L \subseteq 0^*$, then there exists a constant c such that $A_L(n) \leqslant c e^{\sqrt{N_L(n) \log N_L(n)}}$.*

Proof. (1) See [33, Theorem 3] or [22].

(2) See [33, Theorem 7].

(3) The inequality $A_L(n) \leqslant c e^{\sqrt{N_L(n) \log N_L(n)}}$ for some constant c follows from results in [9]. (See also [26–29, 12].) □

4. Bounds on deterministic automaticity: The unary case

Some very interesting questions arise when one attempts to determine automaticity of functions over a 1-letter alphabet, say $\Sigma = \{0\}$.

The sequences $S_0(n), S_{00}(n)$, etc. introduced in Theorem 1 are nothing more than the suffixes of the sequence $S_\varepsilon(n)$. Thus there is a connection with string matching.

Let us introduce some notation. We say that the string x is a factor of a string y if there exist strings w, z such that $y = wxz$. If $\Sigma = \{0\}$, and $f : \Sigma^* \to \Delta$, we define $w = w(f) = f(\varepsilon)f(0)f(0^2)f(0^3)\cdots$. We call $w(f)$ the characteristic word of f.

Lemma 2. *Let $\Sigma = \{0\}$, $\ell = |\Delta| \geqslant 2$, and let $f : \Sigma^* \to \Delta$ be any function. Let $w = w(f) = w_0 w_1 w_2 \ldots$ be the characteristic word of f, so $w_i = f(0^i)$. Then $A_f(n) = n + 1 - t$, where t is the length of the longest (possibly empty) suffix of $w_0 w_1 \ldots w_n$ that is also a factor of $w_0 w_1 \ldots w_{n-1}$.*

Proof. If there is such a suffix of length t, then there exists $m < n$ such that

$$w_{m+1-t} \ldots w_m = w_{n+1-t} \ldots w_n.$$

Hence $S_{0^{n-k}}(n)$ is a prefix of $S_{0^{m-k}}(n)$ for $0 \leqslant k \leqslant t-1$. It follows that $A_L(n) \leqslant n+1-t$.

On the other hand, if $A_L(n) < n + 1 - t$, then $S_{0^{n-t}}(n)$ would be a prefix of $S_{0^i}(n)$ for some $i < n - t$, contradicting the maximality of t. □

It is easy to see that $A_f(n) \leqslant n + 1$; in fact, this bound can be attained for any *particular* value of n by setting $f(0^i) = 0$ for $0 \leqslant i < n$, and $f(0^n) = 1$. We have $S_\varepsilon = 0^n 1$, and none of the successive suffixes are prefixes of any other suffix.

A more interesting question is to ask about the behavior of $A_f(n)$ for any fixed f, as $n \to \infty$. We will prove the following:

Theorem 3. *Let $\Sigma = \{0\}$ and $\ell = |\Delta| \geqslant 2$. Then for any function $f : \Sigma^* \to \Delta$ the inequality $A_f(n) \leqslant n + 1 - \lfloor \log_\ell n \rfloor$ holds for infinitely many n.*

Proof. Define $n = n(m) = \ell^m + m - 1$. Note that $m = \lfloor \log_\ell n \rfloor$. Consider the string

$$S_\varepsilon = f(\varepsilon)f(0)f(0^2)\ldots f(0^n) = w_0 w_1 \ldots w_n.$$

Contained in the string S_ε are $\ell^m + 1$ (overlapping) factors of length m. Hence there must be some factor x that appears at least twice in S_ε. Choose x so that $n' = n'(m)$, the position at which the second occurrence (counting from the left) of x ends, is as small as possible.

Let the first occurrence of x be $w_k w_{k+1} \dots w_{k+m-1}$, and let the second occurrence be $w_{n'+1-m} \dots w_{n'}$. Then, by Lemma 2,

$$A_f(n') \leqslant n' + 1 - m \leqslant n + 1 - m = n + 1 - \lfloor \log_\ell n \rfloor.$$

To see that the inequality is true for infinitely many n', it remains to see that $n'(m)$ is strictly increasing. Suppose $n'(m+1) \leqslant n'(m)$. Then there would be a factor of length m whose second occurrence ends at a position $\leqslant n'(m+1) - 1$, contradicting the minimality of $n'(m)$. This completes the proof. □

Is it possible to explicitly construct an f for which $n - A_f(n) = \mathrm{O}(\log n)$? The answer is yes.

Theorem 4. *Let f be the function which assigns to $\varepsilon, 0, 0^2, \dots$ the values*

$$w = 2202120020121021120002001201020112100210121102\dots,$$

In other words, we concatenate all possible binary strings of length 0, 1, 2, etc., separated by 2's. Then $n - A_f(n) = \mathrm{O}(\log n)$.

Proof. Looking at the proof of the previous theorem, we see that what we are trying to do is construct an infinite sequence such that the longest factor that occurs twice in any prefix of length n is $\mathrm{O}(\log n)$.

Suppose we consider a prefix P of w of length n. Between any two occurrences of 2 in P, there is a string of 0's and 1's of length $\leqslant \log_2 n$. Hence any factor of length at least $2 + 2\log_2 n$ must contain two 2's. But then this cannot possibly match an earlier factor. It follows that all duplications must be of length $< 2 + 2\log_2 n$. □

We can also make this construction work with a binary alphabet by recoding: we replace each 0 by 0, each 1 by 10 and each 2 by 11. The same argument as before works, and we have now expanded the string by a factor of at most 2. Hence the longest duplication is of length $< 4 + 4\log_2 n$. It follows that for this f we have $n - A_f(n) = \mathrm{O}(\log n)$.

A construction improving the 4 to 2 was given independently by Condon et al. [11].

We now prove the following "almost all" result. We assume a uniform probability model, where the probability of $f(w) = d$ for $d \in \Delta$ is $1/|\Delta|$.

Theorem 5. *Suppose $\Sigma = \{0\}$ and $|\Delta| = \ell \geqslant 2$. Then for almost all functions $f : \Sigma^* \to \Delta$ we have $A_f(n) > n - 2\log_\ell n - 2\log_\ell \log_\ell n$ for all sufficiently large n.*

Proof. Let us first estimate the number of distinct unary automata with outputs in Δ that have j states. It suffices to consider only those automata whose transition diagram is topologically connected. It is easy to see that the graph of such an automaton

must consist of j states connected consecutively, followed by the highest numbered state connected back to some previous state. Thus, topologically speaking, there are j possibilities. Since each state can have a different output associated with it, there are ℓ^j different possible output functions. This gives us an upper bound of $j\ell^j$ for the number of distinct connected automata with j states.

Since for any positive integer q we have

$$\sum_{1\leqslant j\leqslant q} j\ell^j = \frac{\ell^{q+1}(q\ell - q - 1) + \ell}{(\ell-1)^2} \leqslant \frac{\ell^{q+1}(q+1)}{\ell - 1},$$

it follows that the number of functions from $\Sigma^{\leqslant n}$ to Δ that are given by DFAO's with $\leqslant q$ states is bounded above by $\ell^{q+1}(q+1)/(\ell-1)$. Now set $q = c(n)$, where

$$c(n) = n - 2\log_\ell n - 2\log_\ell \log_\ell n.$$

Then, since the total number of functions from $\Sigma^{\leqslant n}$ to Δ is ℓ^{n+1}, the probability that a randomly chosen function f satisfies $A_f(n) \leqslant c(n)$ is bounded above by

$$\frac{n + 1 - 2\log_\ell n - 2\log_\ell \log_\ell n}{(\ell-1)n^2(\log_\ell n)^2} = \mathrm{O}\left(\frac{1}{n(\log n)^2}\right).$$

Since $\sum_{n\geqslant 2} \frac{1}{n(\log n)^2}$ converges, by the Borel–Cantelli lemma [15, p. 188], we must have $A_f(n) > c(n)$ for all sufficiently large n. □

For languages, we immediately get the following corollary:

Corollary 6. *For almost all languages $L \subseteq 0^*$, we have*

$$A_L(n) > n - 2\log_2 n - 2\log_2 \log_2 n$$

for all sufficiently large n.

5. An upper bound on the number of distinct unary NFA languages

In this section, we digress briefly to prove an upper bound on the number of distinct unary languages accepted by NFAs with q states. This bound will be used in the next section.

Theorem 7. *There are $\mathrm{O}(q/\log q)^q$ distinct unary languages accepted by NFAs with q states.*

The basic idea of the proof is to find a decomposition for such languages that can be completely described by a small number of parameters. The proof depends on a number of lemmas.

We assume that $L \subseteq a^*$ is a unary language. We say that L is c-monotonic if, for all $n \geqslant 0$, we have $a^n \in L$ implies $a^{n+c} \in L$. We also say that L is c-periodic after N

if, for all $n \geqslant N$, we have $a^n \in L$ iff $a^{n+c} \in L$. As a consequence of these definitions, it follows that if L is c-monotonic, then there exists a constant N such that L is c-periodic after N. (For if L is c-monotonic, for each residue class $j \pmod c$ consider the first occurrence, if it exists, of a^i where $i \equiv j \pmod c$). On the other hand, if L is c-periodic after N, it may not necessarily be c-monotonic.

Lemma 8. (a) *Let L_1, L_2 be c-monotonic unary languages. Then so is $L_1 \cup L_2$.*

(b) *Let L_1, L_2 be unary languages that are c-periodic after N. Then so is $L_1 \cup L_2$.*

Proof. Clear. □

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a unary NFA, i.e., an NFA where $\Sigma = \{a\}$. We call a sequence $(p_0, p_1, \ldots, p_r)$ of states of Q an accepting path for the string $w = a^r$ if $p_0 = q_0$, $p_r \in F$, and $p_i \in \delta(p_{i-1}, a)$ for $1 \leqslant i \leqslant r$.

If $M = (Q, \Sigma, \delta, q_0, F)$ is a unary NFA, then by $G(M)$ we mean the underlying digraph of M, given by (V, E), where $V = Q$ and

$$E = \{(p, p') : p \in Q,\ p' \in \delta(p, a)\}.$$

Also define $L(M, s)$ to be the set of all strings $w \in L$ having an accepting path that contains the state s.

Lemma 9. *Let M be a unary NFA with q states such that $G(M)$ has a directed cycle of length c. Let s be any state contained in a directed cycle of length c. Then $L(M, s)$ is c-monotonic.*

Proof. Let $w = a^n$ be a string in $L(M, s)$, let $(p_0, p_1, \ldots, p_n)$ be an accepting path for w, and let $p_i = s$ be a state contained in a cycle C of length c. Then we can create an accepting path for a^{n+c} by arriving at p_i, going around the states of C, returning to p_i, and then continuing to p_n. □

Lemma 10. *Let M, s, and $L(M, s)$ be as in Lemma 9. Then $L(M, s)$ is c-periodic after $(c+1)(q-1)$.*

Proof. Suppose $w = a^\ell \in L = L(M)$, with $\ell \geqslant (c+1)q - 1$, and suppose there exists an accepting path for w containing a state s, where s lies in a cycle of length c in $G(M)$. We will show how to produce an accepting path that contains s for $a^{\ell - kc}$, for some integer $k \geqslant 1$. The result will then follow from Lemma 9.

Let the accepting path for w be $\mathscr{P} = (p_0, p_1, \ldots, p_\ell)$, and let i be the smallest index such that $p_i = s$. Divide the accepting path into the prefix $P = (p_0, p_1, \ldots, p_i = s)$ and the suffix $S = (p_i = s, p_{i+1}, \ldots, p_\ell)$. Note that P and S together contain $i+1+\ell-i+1 = \ell + 2 > (c+1)q$ states. Let p' be any state that occurs most frequently in P, and s' be any state that occurs most frequently in S. The total number of occurrences of both p' and s' in P and S is $\geqslant c + 2$. If any two of the occurrences of p' or two of the

occurrences of s' in $\mathscr{P}$ are separated by a subpath $\mathscr{P}'$ of length $k \equiv 0 \pmod{c}$, then we obtain an accepting path for $a^{\ell-kc}$ by simply cutting out $\mathscr{P}'$.

Otherwise, assume that no two occurrences of p' or s' are separated by a subpath of length $\equiv 0 \pmod{c}$. Call this Assumption A. We will shorten the path as follows: we cut out both the section between some occurrence of p' in P and the last occurrence of p' in P, shortening $\mathscr{P}$ by d, and the section between the first occurrence of s' in S, and some later occurrence of s' in S, shortening $\mathscr{P}$ by e. Now consider the $\geqslant c$ possible values $d \pmod{c}$ and $-e \pmod{c}$. Since by Assumption A, no two of the possible choices for d are equal $\pmod{c}$, the choices for d must be distributed in the *nonzero* residue classes $\pmod{c}$. The same thing holds for $-e$. Since there are at least c choices for d and $-e$, by the pigeonhole principle, there must be at least one pair $(d, -e)$ for which $d \equiv -e \pmod{c}$; hence $d + e \equiv 0 \pmod{c}$. By cutting out both corresponding sections, we obtain an accepting path for $a^{\ell-kc}$ for some k. □

Next, we prove a lemma about directed graphs. We say that a graph G is of girth c if every directed cycle is of length $\geqslant c$. If G is acyclic, its girth is defined to be infinite.

Lemma 11. *Let G be a digraph on q vertices of girth $> 2q/3$. Then there exists at least one vertex v that lies in every cycle.*

Proof. Any two directed cycles in G must have $> 2q/3 + 2q/3 - q = q/3$ vertices in common. Hence, any three directed cycles must have $> 2q/3 + q/3 - q > 0$ vertices in common. The result now follows from a theorem of Kosaraju [24]; also see [1, 34]. □

The next lemma introduces the decomposition we will use to count the number of languages accepted by a unary NFA with q states.

Lemma 12. *Let M be a unary NFA with q states. Then there exists an integer $r \geqslant 0$, a strictly increasing sequence $c_1 < c_2 < \cdots < c_r$, languages $L_1, L_2, \ldots, L_r$, and an NFA M_{r+1} such that*

$$L(M) = \left(\bigcup_{1 \leqslant i \leqslant r} L_i \right) \cup L(M_{r+1}) \tag{1}$$

and, for $1 \leqslant i \leqslant r$, the language L_i is c_i-monotonic and c_i-periodic after $(c_r + 1)(q - 1)$. Furthermore, if M_{r+1} has q' states, then the girth of $G(M_{r+1})$ is $> 2q'/3$, and if $r \geqslant 1$, then $q' \geqslant c_r/2$. Finally, $q' = q - (c_1 + c_2 + \cdots + c_r)$.

Proof. We describe a recursive procedure for computing the decomposition of $L(M_i)$. Let M_i have n_i states, and let c_i be the girth of $G(M_i)$. If $c_i > 2n_i/3$, we terminate the procedure. Otherwise, we write

$$L(M_i) = L_i \cup L(M_{i+1}),$$

where $L_i = \{w \in L(M_i)$: there exists an accepting path for w that contains a state in some cycle of length $c_i\}$, and M_{i+1} is obtained from M_i by removing all states in all cycles of length c_i. Note that we can take M_{i+1} to have exactly $n_i - c_i$ states, some of which may be inaccessible. Clearly, this procedure terminates, since at each step we remove a positive number of states. It follows that $q' = q - (c_1 + c_2 + \cdots + c_r)$.

If we write $M = M_1$, this gives us the decomposition

$$L(M_1) = L_1 \cup L_2 \cup \cdots \cup L_r \cup L(M_{r+1}),$$

where $c_1 < c_2 < \cdots < c_r$. The termination condition gives us $c_{r+1} > 2q'/3$. Furthermore, $c_r \leqslant 2n_r/3$. Since $q' = n_r - c_r$, we have $q' \geqslant 3c_r/2 - c_r \geqslant c_r/2$. The fact that L_i is c_i-monotonic and c_i-periodic after $(c_r + 1)(q - 1)$ follows from Lemmas 8–10. □

We are now ready to prove Theorem 7. The idea is to count the number of languages accepted by an NFA with q states by parameterizing the decomposition given in Lemma 12.

Proof of Theorem 7. We can completely specify any language accepted by an NFA with q-states by providing:

(1) the integers $c_1, c_2, \ldots, c_r$;

(2) for each pair (i,j) with $1 \leqslant i \leqslant r$ and $0 \leqslant j < c_i$, whether or not there exists an $n \geqslant 0$ with $n \equiv j \pmod{c_i}$ and $a^n \in L_i$;

(3) for each pair (i,j), with $1 \leqslant i \leqslant r$ and $0 \leqslant j < c_i$, the cardinality of

$$L_{i,j} = \left\{ a^n \in L_i - \left(\bigcup_{1 \leqslant t < i} L_t \right) : n < (c_r + 1)(q - 1) \text{ and } n \equiv j \pmod{c_i} \right\};$$

(4) the residual language $L(M_{r+1})$.

First, let us argue that these specifications suffice. From Lemma 12, we know that in the decomposition (1), each L_i is c_i-periodic after $(c_r + 1)(q - 1)$. It follows that L_i is completely determined by specifying c_i, the congruence classes $(\bmod c_i)$ of lengths of strings that are eventually covered by members of L_i, and the strings of length $< (c_r + 1)(q - 1)$. However, since each L_i is also c_i-monotonic, it is not necessary to actually specify all the strings of length $< (c_r + 1)(q - 1)$ in L_i. It suffices to specify, for each $j < c_i$, the shortest such string with length congruent to $j \pmod{c_i}$. And if this string is contained in L_t, for $t < i$, it need not be mentioned; thus it actually suffices to give the shortest such string s not contained in $\bigcup_{1 \leqslant t < i} L_t$. But then s is completely determined by the cardinality of $L_{i,j}$.

We now bound the number of possibilities in each of these parts as follows:

(1) Since $c_1 < c_2 < \cdots < c_r \leqslant q$, it suffices to specify a subset of cardinality r of $\{1, 2, \ldots, q\}$. Hence there are at most 2^q possibilities.

(2) The total number of possibilities is $2^{c_1 + c_2 + \cdots + c_r} = 2^{q - q'}$.

(3) The number of ways of choosing n nonnegative integers whose sum is $\leqslant m$ is

$$\binom{m+n}{n} < (m+n)^n / n!.$$

The number of possibilities here can be enumerated by counting the number of ways of choosing $c_1 + c_2 + \cdots + c_r = q - q'$ nonnegative integers whose sum is at most $(c_r + 1)(q - 1)$. This gives us the upper bound

$$B = ((c_r + 1)(q - 1) + q - q')^{q-q'}/(q - q')!. \tag{2}$$

Now we know from Lemma 12 that $c_r \leqslant 2q'$, so, by Stirling's approximation,

$$B = \mathrm{O}(qq'/(q - q'))^{q-q'}.$$

If $q' > 2q/3$, then $B = \mathrm{O}(q^{2/3})^q = \mathrm{O}(q/\log q)^q$. If $q' < 2q/3$, then $B = \mathrm{O}(q')^{q-q'}$. Now, by logarithmic differentiation with respect to q', it is easy to see that B is maximized by choosing $q' = \mathrm{O}(q/\log q)$, giving the bound $B = \mathrm{O}(q/\log q)^q$.

(4) If $G(M_{r+1})$ is acyclic, then $L(M_{r+1})$ can be specified completely by specifying all the strings it accepts, and there are at most $2^{q'}$ possibilities. Otherwise the girth of $G(M_{r+1})$ is finite and exceeds $2q'/3$, so by Lemma 11, there is a vertex v (i.e., a state of M_{r+1}) that lies in every cycle. Now $L(M_{r+1})$ can be specified completely by describing $A = L \cap \Sigma^{<q'}$ and $B = L \cap \Sigma^{\geqslant q'}$. There are $2^{q'}$ possibilities for A. Let w be a string in B, and consider the sequence of states encountered in an accepting path $(p_0, \ldots, p_f)$ for w in M_{r+1}. By the pigeonhole principle, some state $p = p_i$ must be repeated. This corresponds to a cycle in $G(M_{r+1})$, which must contain v. Now consider the portion $\mathscr{P}$ of the accepting path from v to p_f. Either $\mathscr{P}$ is of length $< q'$, or again, by the pigeonhole principle, some state must be repeated. Let p' be the first repeated state. Since v is in every cycle, we must have $p' = v$. Continuing in this fashion, we see that every accepting path of length $\geqslant q'$ can be split into three parts: (i) an initial portion of length $< q'$, (ii) a concatenation of cycles (possibly 0) beginning and ending at v, and (iii) a tail of length $< q'$. These accepting paths are completely specified by providing (i) the list of lengths of acyclic paths from p_0 to v, which is a subset of $[0, q')$, (ii) the set of possible cycle lengths, which is a subset of $(2q'/3, q']$, and (iii) the lengths of acyclic paths from v to any final state, which is a subset of $[0, q')$. It follows that there are at most $2^{q'} \cdot 2^{q'/3} \cdot 2^{q'} = 2^{7q'/3}$ possibilities for B. Multiplying this by the $2^{q'}$ possibilities for A gives a total of at most $2^{10q'/3}$ languages accepted by an NFA with underlying graph having finite girth $\geqslant 2q'/3$. Thus the total number of possibilities for $L(M_{r+1})$ is $2^{10q'/3} + 2^{q'}$.

By multiplying all four of these bounds together, we see that the number of distinct languages accepted by unary NFAs with q states is $\mathrm{O}(q/\log q)^q$. □

6. Bounds on nondeterministic automaticity: The unary case

In this section we give upper and lower bounds on nondeterministic automaticity when $L \subseteq 0^*$.

Corollary 13. *Suppose $L \subseteq 0^*$. Then there exists a constant c such that for almost all L we have $N_L(n) > cn/\log n$ for all sufficiently large n.*

Proof. The result follows immediately from the Borel–Cantelli lemma and Theorem 7 of the previous section. □

It is also easy to prove the following upper bound:

Theorem 14. *Let* $L \subseteq 0^*$. *Then* $N_L(n) \leqslant n + 1 - \lfloor \log_2 n \rfloor$ *for infinitely many* n.

Proof. This follows immediately from Theorems 3 and 1(3). □

Can one find an explicit example of a unary language with linear nondeterministic automaticity? The answer is yes, as the following theorem shows.

Theorem 15. *Define* $L_5 = \{0^{2^i} : i \geqslant 0\}$. *Then* $N_L(n) \geqslant n/4$ *for all* $n \geqslant 0$.

Proof. In [17], the following useful result is proved: suppose there exist r pairs of strings (x_i, w_i), $1 \leqslant i \leqslant r$, such that $x_i w_i \in L$ for all i, and $x_i w_j \notin L$ for all $i \neq j$, and $|x_i w_j| \leqslant n$ for all i, j, then $N_L(n) \geqslant r$.

We now apply this result to L_5 to show that, for $k \geqslant 2$,

$$N_{L_5}(n) \geqslant \begin{cases} 2^{k-2} + n - 2^k + 1 & \text{if } 2^k \leqslant n < 2^k + 2^{k-2}, \\ 2^{k-1} & \text{if } 2^k + 2^{k-2} \leqslant n < 2^{k+1}, \end{cases}$$

from which the desired result follows immediately.

To see this, first assume $2^k \leqslant n < 2^k + 2^{k-2}$. Then let $x_i = 0^{2^{k-2}+i-1}$ for $1 \leqslant i \leqslant 2^{k-2} + n - 2^k + 1$. Also define

$$w_i = \begin{cases} 0^{2^{k-2}-i+1} & \text{for } 1 \leqslant i \leqslant 2^{k-2}, \\ 0^{2^k - 2^{k-2}-i+1} & \text{for } 2^{k-2} < i \leqslant 2^{k-2} + n - 2^k + 1. \end{cases}$$

It is now easy to verify that

$$x_i w_i = \begin{cases} 0^{2^{k-1}} & \text{for } 1 \leqslant i \leqslant 2^{k-2}, \\ 0^{2^k} & \text{for } 2^{k-2} < i \leqslant 2^{k-2} + n - 2^k + 1, \end{cases}$$

and that $|x_i w_j|$ is not a power of 2 for $i \neq j$. □

7. Automaticity lower bounds for nonregular languages in the unary case

Recall that Karp's theorem (Theorem 1(1)) says that if L is not regular, then $A_L(n) \geqslant (n+3)/2$ for infinitely many n. The proof does not depend on k (the size of the input alphabet), and hence is true if $k = |\Sigma| = 1$. It follows that if $L \subseteq 0^*$ is not regular, then

$$\limsup_{n \to \infty} \frac{A_L(n)}{n} \geqslant \frac{1}{2}.$$

However, this lower bound of $\frac{1}{2}$ is apparently not attainable in the unary case. We make the following

Conjecture 16. If $L \subseteq 0^*$ is not regular, then

$$\limsup_{n\to\infty} \frac{A_L(n)}{n} \geqslant \frac{\sqrt{5}-1}{2} \doteq 0.61803.$$

Using Lemma 2, we can rephrase this conjecture in a purely combinatorial fashion:

Conjecture 16′. Let $w = w_0w_1w_2\ldots$ be an infinite word over a finite alphabet that is not ultimately periodic. Define $s_w(n)$ to be the length of the longest suffix of $w_0w_1\ldots w_n$ that is also a factor of $w_0w_1\ldots w_{n-1}$. Then

$$\liminf_{n\to\infty} \frac{s_w(n)}{n} \leqslant \frac{3-\sqrt{5}}{2} \doteq 0.38197.$$

Allouche has kindly informed us that Conjecture 16′ is related to a similar one of Rauzy [30, Section 5.2]. This last paper also mentions that Rauzy's conjecture has been proved by Rauzy in the case of the so-called Sturmian words. See [2] for more details. Also see [16].

We do not know how to prove Conjecture 16. However, we can prove that *if* the conjecture is true, then the constant $(\sqrt{5}-1)/2$ is best possible. In fact, this bound is achieved for an L related to f, the famous Fibonacci word [6, 7].

One possible definition of f is as follows: define $h_1 = 1$, $h_2 = 0$, and $h_n = h_{n-1}h_{n-2}$. Thus, for example, $h_3 = 01$, $h_4 = 010$, $h_5 = 01001$, etc. Clearly $h_n \leqslant h_{n+1}$ for all $n \geqslant 2$, and hence it is meaningful to define $f = \lim_{n\to\infty} h_n$. We write the individual bits of f as $f_0, f_1, \ldots$, and we have

$$f = f_0f_1f_2\ldots = 0100101001001\ldots$$

Notice that $|h_i| = F_i$, where F_i is the i'th Fibonacci number, defined by $F_0 = 0$, $F_1 = 1$, and $F_i = F_{i-1} + F_{i-2}$ for $i \geqslant 2$.

There is a remarkable description of f in terms of Fibonacci representations. It is well known (see, for example, [25, 36]) that every integer $n \geqslant 0$ can be expressed uniquely as

$$n = \sum_{i\geqslant 1} a_i(n)F_{i+1},$$

where $a_i = a_i(n) \in \{0, 1\}$, and $a_ia_{i+1} = 0$ for all $i \geqslant 1$. We can think of the Fibonacci representation of n as an infinite string $a_1a_2a_3\ldots$ where only finitely many of the a_i's are equal to 1; we write $n_{(F)} = a_1a_2a_3\ldots$. Also, we define $\mathrm{fval}(a_1a_2\ldots a_k) = \sum_{1\leqslant i\leqslant k} a_kF_{k+1}$.

We have the following well-known theorem [23, Exercise 1.2.8.36].

Theorem 17. *Let n be a nonnegative integer. Then $f_n = a_1(n)$.*

We define the unary Fibonacci language, L_F, as follows:

$$L_F = \{0^i : f_i = 0\} = \{\varepsilon, 0^2, 0^3, 0^5, 0^7, 0^8, \ldots\}.$$

It is known that f is not ultimately periodic (this follows, for example, from Karhumäki's result [21] that f is fourth-power-free), and hence L_F is not regular. We will prove that if $L = L_F$, then $\limsup_{n\to\infty} A_L(n)/n = (\sqrt{5}-1)/2$. The proof depends on a lemma of independent interest, which gives the number of matches between two shifts of the Fibonacci word.

First, we introduce some notation. If $w = w_0 w_1 w_2 \ldots$ is an infinite word, then by $n\downarrow w$ we mean the infinite word $w_n w_{n+1} \ldots$ By $w_{a\ldots b}$ we mean the word $w_a w_{a+1} \ldots w_b$. If $v = v_1 v_2 v_3 \ldots$ and $w = w_1 w_2 w_3 \ldots$ are words (finite or infinite), then by $d(v,w)$ we mean the least index i for which $v_i \neq w_i$. If no such index exists, then we write $d(v,w) = \infty$. We define $m(v,w) = d(v,w) - 1$; thus $m(v,w)$ counts the length of the longest matching prefix of v and w.

Lemma 18. *Let r,s be nonnegative integers with $r \neq s$. Suppose $d(r_{(F)}, s_{(F)}) = k$. Then $m(r\downarrow f, s\downarrow f) = F_{k+2} - (t+2)$, where $t = \mathrm{fval}(a_1(r)a_2(r)\ldots a_{k-1}(r))$.*

Proof. Notice that the formula is actually symmetric in r and s, since by definition $a_i(r) = a_i(s)$ for $1 \leqslant i \leqslant k-1$.

Without loss of generality, let us assume that $a_k(r) = 1$ and $a_k(s) = 0$. If $k = 1$, then $t = 0$, and hence $m(r\downarrow f, s\downarrow f) = F_3 - 2 = 0$. Hence, let us assume $k \geqslant 2$. As r and s are successively incremented, their Fibonacci representations coincide on bits 1 through $k-1$, up to and including $r + F_k - (t+1)$ and $s + F_k - (t+1)$. Then, at $r' = r + F_k - t$, $s' = s + F_k - t$, we have $a_{1\ldots k-1}(r) = 0^{k-1}$, $a_{1\ldots k-1}(s) = 0^{k-2}1$, and $d(r'_{(F)}, s'_{(F)}) = k-1$.

Now, as r' and s' are successively incremented, their Fibonacci representations coincide on bits 1 through $k-2$, up to and including $r' + F_{k-1} - 1$ and $s' + F_{k-1} - 1$. Then, at $r'' = r' + F_{k-1}$, $s'' = s' + F_{k-1}$, we have $a_{1\ldots k-2}(r'') = 0^{k-3}1$, $a_{1\ldots k-2}(s'') = 0^{k-2}$, and $d(r''_{(F)}, s''_{(F)}) = k-2$.

In the same manner as the previous paragraph, as r'' and s'' are successively incremented, their Fibonacci representations coincide on bits 1 through $k-3$, up to and including $r'' + F_{k-2} - 1$ and $s'' + F_{k-2} - 1$. Then, at $r''' = r'' + F_{k-2}$, $s''' = s'' + F_{k-2}$, we have $a_{1\ldots k-3}(r''') = 0^{k-3}$ and $a_{1\ldots k-3}(s''') = 0^{k-4}(1)$.

This continues until the pair $(r^{(k-2)}, s^{(k-2)})$, for which $d(r^{(k-2)}_{(F)}, s^{(k-2)}_{(F)}) = 2$. Finally, we see that if $r^{(k-1)} = r^{(k-2)} + 1$ and $s^{(k-1)} = s^{(k-1)} + 1$, then $d(r^{(k-1)}_{(F)}, s^{(k-1)}_{(F)}) = 1$, and hence $a_1(r^{(k-1)}) \neq a_1(s^{(k-1)})$. We see that

$$\begin{aligned} r'' - r' &= F_{k-1}, \\ r''' - r'' &= F_{k-2}, \\ &\vdots \\ r^{(k-1)} - r^{(k-2)} &= F_2 = 1, \end{aligned}$$

and so $r^{(k-1)} - r' = \sum_{2 \leqslant j \leqslant k-1} F_j = F_{k+1} - 2$.

Adding this to $r' - r = F_k - t$, we see that the strings $r\downarrow f$ and $s\downarrow f$ differ for the first time at position $F_k - t + F_{k+1} - 2 = F_{k+2} - (t+2)$. This completes the proof of the Lemma. □

Corollary 19. *Let* $d(r_{(F)}, s_{(F)}) = k$. *Then*

$$F_{k+1} - 1 \leqslant m(r\downarrow f, s\downarrow f) \leqslant F_{k+2} - 2,$$
$$F_{k+1} \leqslant d(r\downarrow f, s\downarrow f) \leqslant F_{k+2} - 1.$$

As the referee points out, related results can be found in [10].

Theorem 20. *Let* $L = L_F$, *the unary Fibonacci language. Suppose* $F_n - 2 \leqslant k \leqslant F_{n+1} - 3$. *Then* $A_L(k) = F_{n-1}$.

Proof. First we show that $A_L(k) \geqslant F_{n-1}$. Since $A_L(k)$ is an increasing function of k, it suffices to prove this for $k = F_n - 2$.

For this language L, and this value of k, we have $S_\varepsilon = f_{0\ldots(F_n-2)}$, and $S_{0^i} = f_{i\ldots(F_n-2)}$ for $0 \leqslant i \leqslant F_n - 2$. Define $x_i = f_{i\ldots(F_n-2)}$. We then partition the collection $\{S_{0^i} : 0 \leqslant i < F_{n-1}\}$ as follows:

$$D_1 = \{x_i : 0 \leqslant i \leqslant F_{n-2}\},$$
$$D_2 = \{x_i : F_{n-2} < i < F_{n-1}\}.$$

We will show that $D_1 \cup D_2$ consists of F_{n-1} strings that are pairwise incomparable under the prefix ordering. From this the result will follow.

First, we show that all the elements of D_1 are mutually incomparable under the $\leqslant$ ordering. This follows because each string in D_1 is as long or longer than $x_{F_{n-2}}$, which is of length $(F_n - 2) - F_{n-2} + 1 = F_{n-1} - 1$. But according to Corollary 19, $d(i\downarrow f, j\downarrow f) \leqslant F_{\ell+2} - 1$, where $\ell = d(i_{(F)}, j_{(F)})$. Since $i, j \leqslant F_{n-2}$, it follows that $\ell \leqslant n-3$. Hence we have $d(i\downarrow f, j\downarrow f) \leqslant F_{n-1} - 1$, which means that the two strings x_i and x_j differ in a position which is, at worst, their rightmost position. Thus, x_i and x_j are incomparable.

Next, we show that all the elements of D_2 are mutually incomparable under $\leqslant$. Let i, j be distinct integers such that $F_{n-2} < i, j < F_{n-1}$. Then i and j both have a 1-bit corresponding to the summand F_{n-2} in their Fibonacci representation. Thus $a_{n-3}(i) = a_{n-3}(j) = 1$. Since Fibonacci representations do not contain consecutive 1's, it follows that $a_{n-4}(i) = a_{n-4}(j) = 0$. Hence $d(i_{(F)}, j_{(F)}) \leqslant n - 5$. It follows from Corollary 19 that $d(i\downarrow f, j\downarrow f) \leqslant F_{n-3} - 1$. But $|x_i|, |x_j| \geqslant (F_n - 2) - (F_{n-1} - 1) + 1 = F_{n-2}$. Hence the two strings x_i and x_j differ, and so are incomparable.

Finally, we show that all the elements of D_2 are not comparable to elements of D_1. Let $0 \leqslant i \leqslant F_{n-2}$, and $F_{n-2} < j < F_{n-1}$. If $d(i\downarrow f, j\downarrow f) \leqslant n - 4$, then this follows as in the previous paragraph. Since $d(i_{(F)}, j_{(F)}) < n - 2$, the remaining case is when $d(i_{(F)}, j_{(F)}) = n - 3$. This can only occur when $i = a$ and $j = a + F_{n-2}$. In

this case, Lemma 18 shows that $d(i\downarrow f, j\downarrow f) = F_{n-1} - (a+1)$. On the other hand, the length of x_j (the shorter of the two strings) is $(F_n-2)-(a+F_{n-2})+1 = F_{n-1}-(a+1)$, exactly as long as is necessary to distinguish x_i from x_j.

Thus we have shown $A_L(k) \geqslant F_{n-1}$.

It remains to show that for $F_n - 2 \leqslant k \leqslant F_{n+1} - 3$, we have $A_L(k) \leqslant F_{n-1}$. Again, since $A_L(k)$ is increasing, it suffices to show this for $k = F_{n+1} - 3$. Let $y_i = f_{i\ldots F_{n+1}-3}$. As above, we partition the collection $\{S_{0^i} : 0 \leqslant i \leqslant F_{n+1} - 3\}$ as follows:

$$D = \{y_i : 0 \leqslant i < F_{n-1}\},$$
$$N = \{y_i : F_{n-1} \leqslant i \leqslant F_{n+1} - 3\}.$$

We will show that every string in N is a prefix of some longer string in $D \cup N$. Actually, it suffices to show that $y_{F_{n-1}}$ is a prefix of y_0, for it would then follow that $y_{F_{n-1}+I}$ is a prefix of y_i for $1 \leqslant i \leqslant F_n - 3$. But from Lemma 18, we know that $m(0\downarrow f, F_{n-1}\downarrow f) = F_n - 2$. But the length of $y_{F_{n-1}}$ is $F_n - 2$, so $y_{F_{n-1}}$ is a prefix of y_0. □

Corollary 21. *We have*

$$\limsup_{k\to\infty} \frac{A_L(k)}{k} = (\sqrt{5}-1)/2.$$

Proof. Let $F_n - 2 \leqslant k \leqslant F_{n+1} - 3$. Then

$$\frac{A_L(k)}{k} \leqslant \frac{F_{n-1}}{F_n - 2}.$$

Hence

$$\limsup_{k\to\infty} \frac{A_L(k)}{k} \leqslant \lim_{n\to\infty} \frac{F_{n-1}}{F_n - 2} = \frac{\sqrt{5}-1}{2}.$$

On the other hand, when $k = F_n - 2$, then

$$\frac{A_L(k)}{k} = \frac{F_{n-1}}{F_n - 2},$$

and so

$$\limsup_{k\to\infty} \frac{A_L(k)}{k} = \frac{\sqrt{5}-1}{2}. \qquad \square$$

In the last theorem of this section, we prove a result somewhat stronger than the claim in Conjecture 16, under a somewhat stronger hypothesis.

Theorem 22. *Let $a_1 < a_2 < a_3 < \cdots$ be a strictly increasing sequence of nonnegative integers, and define*

$$L = \{0^{a_1}, 0^{a_1+a_2}, 0^{a_1+a_2+a_3}, \ldots\}.$$

Then

$$\limsup_{n\to\infty} \frac{A_L(n)}{n} \geqslant \frac{3}{4}.$$

The constant $\frac{3}{4}$ *cannot be replaced by any larger number.*

Proof. It is readily verified using Lemma 2 that for $w = w(L)$ we have $s_w(\sum_{1\leqslant i\leqslant k} a_i) = a_{k-1}$. It follows that

$$\limsup_{n\to\infty} \frac{A_L(n)}{n} \geqslant 1 - \liminf_{n\to\infty} \frac{a_{n-1}}{\sum_{1\leqslant i\leqslant n} a_i}.$$

Now it can be shown (see [8]) that for all sequences of positive real numbers $a_1, a_2, \ldots$ we have

$$\liminf_{n\to\infty} \frac{a_{n-1}}{\sum_{1\leqslant i\leqslant n} a_i} \leqslant \frac{1}{4}.$$

From this, the first result follows.

To see that the constant $\frac{3}{4}$ is best possible, consider $L = \{0^{2^i} : i \geqslant 0\}$. For this L we have $A_L(2^n) = 1 + 3\cdot 2^{n-2}$ for $n \geqslant 3$. □

8. Lower bounds for nondeterministic automaticity for nonregular languages

In this section we are interested in obtaining lower bounds, similar to that given in Karp's theorem, for the nondeterministic automaticity of nonregular languages.

Theorem 23. *There exists a constant* c' *(which does not depend on* L*) such that if* $L \subseteq 0^*$ *is not regular, then* $N_L(n) \geqslant c'(\log n)^2/(\log\log n)$ *infinitely often.*

Proof. Suppose, to the contrary, that L is nonregular and $N_L(n) < c'(\log n)^2/(\log\log n)$ for all n sufficiently large. Then from Theorem 1(3), we have $A_L(n) < cn^{\sqrt{c'/2}}$ for all n sufficiently large. (Here c is the constant in Theorem 1(3)). By choosing c' sufficiently small, we get a contradiction with Theorem 1(1). □

We now give a "natural" unary language with nondeterministic automaticity close to the lower bound in Theorem 23.

Theorem 24. *Let* $L_1 = \{0^{n^2} : n \text{ odd}, \geqslant 1\}$. *Then* $L_2 = \overline{L_1} = 0^* - L_1$ *is not a regular language. Assuming Conjecture* 25 *below, we have* $N_{L_2}(n) = O((\log n)^2(\log\log n))$. *Assuming the Extended Riemann Hypothesis (ERH), we have* $N_{L_2}(n) = O((\log n)^4/(\log\log n))$.

The proof depends on the observation that if a number congruent to 1 (mod 8) "looks like a square" modulo all "small" primes, then it is in fact a square.

More precisely, for r a positive integer $\equiv 1 \pmod 8$ that is not a square, define $h(r)$ to be the least odd prime p such that the Jacobi symbol $(r/p) = -1$. Also define

$$J(m) = \max_{\substack{1 < r \leqslant m \\ r \text{ not a square} \\ r \equiv 1 \pmod 8}} h(r).$$

Then the ERH implies that $J(m) < 3(\log m)^2$; see [4, 35].

The "reasonable conjecture" is the following:

Conjecture 25. We have $J(m) = O((\log m)(\log\log m))$.

A simple probabilistic model gives this better bound (and more), and it is also supported by the available numerical evidence; see, for example, [5].

Proof of Theorem 24. We construct an NFA M such that $L(M) \cap \Sigma^{\leqslant n} = L_2 \cap \Sigma^{\leqslant n}$ as follows: we "guess" an odd prime p and on input 0^j, compute $j \pmod p$ with a cyclic counter. If $(j/p) = -1$ (which depends only on $j \pmod p$), then j cannot be a square, and so we accept. We do this for all odd primes $p < J(n)$. We also have a nondeterministic transition from the initial state to a counter (mod 8), and accept if $j \not\equiv 1 \pmod 8$.

The number of states needed is therefore $9 + \sum_{2 < p \leqslant J(n)} p$, which is $O((\log n)^2(\log\log n))$ assuming Conjecture 25, or $O((\log n)^4/(\log\log n))$ assuming ERH. □

We can also give an example of a nonregular unary language with poly-logarithmic nondeterministic automaticity where the bound does not depend on unproved hypotheses. First, we prove a simple lemma:

Lemma 26. *Define* $\vartheta(x) = \sum_{p \leqslant x} \log p$. *Then* $\vartheta(x) > 0.23x$ *for* $x \geqslant 2$.

Proof. Rosser and Schoenfeld [31, Theorem 10] proved that $\vartheta(x) > 0.84x$ for $x \geqslant 101$. The stated inequality can now easily be verified for $2 \leqslant x < 101$. □

Theorem 27. *Define* $L_3 = \{0^n : n \geqslant 1$ *and the least positive integer not dividing* n *is not a power of* $2\}$. *Then* $N_{L_3}(n) = O((\log n)^3/(\log\log n))$.

Proof. The language L_3 is not regular, since it is proved in [3] that $\overline{L_3}$ is not regular.

Let n be a fixed integer >0; we show how to construct an NFA accepting an nth-order approximation to the language L_3. The construction of our NFA is based on the following two observations:

(i) if $0^n \in L_3$, then there exists a prime power p^k, with $p \geqslant 3$, $k \geqslant 1$ and $p^k \leqslant 4.4 \log n$ such that $n \not\equiv 0 \pmod{p^k}$ and $n \equiv 0 \pmod{2^s}$ where $s \geqslant 0$ is an integer with $2^s < p^k < 2^{s+1}$;

(ii) if there exists a prime power p^k ($p \geqslant 3$, $k \geqslant 1$) such that $n \not\equiv 0 \pmod{p^k}$ and $n \equiv 0 \pmod{2^s}$ with $s \geqslant 1$ and $2^s < p^k < 2^{s+1}$, then $0^n \in L_3$.

Proof of (i): let $0^n \in L_3$, and let t be the least integer not dividing n. Then t is not a power of 2. Clearly t is a prime power. Furthermore, we claim that $t \leqslant 4.4 \log n$. Suppose not; then n is divisible by all the integers $\leqslant 4.4 \log n$. Hence

$$n \geqslant \operatorname*{lcm}_{1 \leqslant k \leqslant 4.4 \log n} k = e^{\psi(4.4 \log n)} \geqslant e^{\vartheta(4.4 \log n)} > n,$$

a contradiction. (Here $\psi(x) = \sum_{p^k \leqslant x} \log p$, and we have used Lemma 26.)

We have $n \not\equiv 0 \pmod{t}$. Also $n \equiv 0 \pmod{2^s}$ for $2^s < t$; for otherwise the least integer not dividing n would be a power of 2.

Proof of (ii): suppose $n \not\equiv 0 \pmod{p^k}$ ($p \geqslant 3$, $k \geqslant 1$) and $n \equiv 0 \pmod{2^s}$ for $s \geqslant 1$ with $2^s < p^k < 2^{s+1}$. Let t be the least integer not dividing n. Then $t \leqslant p^k$. However, since $n \equiv 0 \pmod{2^s}$ for all s with $2^s \leqslant t$, t is not a power of 2. Hence $0^n \in L_3$.

Now an NFA can be constructed using these two observations, as follows: we nondeterministically "guess" an odd prime power $p^k \leqslant 4.4 \log n$, and then, on input 0^r (with $r \leqslant n$), compute $r \pmod{p^k 2^s}$ for s satisfying $2^s < p^k < 2^{s+1}$. We accept if $r \not\equiv 0 \pmod{p^k}$ and $r \equiv 0 \pmod{2^s}$. This requires $1 + \sum_{p^k \leqslant 4.4 \log n} O((p^k)^2)$ states, which is $O((\log n)^3/(\log \log n))$. □

Our last result is the following: define

$$S(q,k) = \{r \in \mathbb{Z}^{\geqslant 0} : r \not\equiv 0 \pmod{q} \text{ and } r \equiv 0 \pmod{2^k}\}.$$

Define

$$\mathscr{B} = \{3,5,7,9,11,13,17,19,23,25,\ldots\},$$

the set of odd prime powers. Given a function f from the integers $\geqslant 3$ into the reals $\geqslant 1$, define the set A_f as follows:

$$A_f = \bigcup_{q \in \mathscr{B}} S(q, \lfloor \log_2 f(q) \rfloor).$$

Then we have

Theorem 28. *Let f be any function from the integers $\geqslant 3$ to the reals $\geqslant 1$. Assume that f is a (not necessarily strictly) increasing unbounded function such that $\lfloor \log_2 f(p^e) \rfloor$ takes on all positive integer values, as p ranges over all odd primes and $e \geqslant 1$. Define $L_f = \{0^n : n \in A_f\}$. Then L_f is not a regular language, and we have $N_{L_f} = f(5 \log n) O((\log n)^2/(\log \log n))$.*

Before giving the proof, we remark that the function $f(x) = x$ satisfies the hypotheses. In this case, we obtain the language L_3 above.

Furthermore, suppose we define $\lg^{(i)} x$ as follows: $\lg x = 1$, if $x \leqslant 2$, and $\lg x = \log_2 x$ if $x > 2$. Also, $\lg^{(1)} x = \lg x$, and $\lg^{(i)} = \lg \lg^{(i-1)} x$ for $i \geqslant 2$. Then the function $\lg^{(i)} x$ satisfies the hypotheses. Thus, using the series of functions $\lg^{(1)} x, \lg^{(2)} x, \lg^{(3)} x, \ldots$, we can obtain a language with nondeterministic automaticity arbitrarily close to the bound $O((\log n)^2/(\log \log n))$.

Proof. First we show that L_f is not regular. We do so by assuming that the complement $\overline{L_f}$ is in fact regular, and obtaining a contradiction.

For each positive integer k, let q_k be the largest odd prime power p^e for which $k = \lfloor \log_2 f(p^e) \rfloor$. (Such a q_k exists by our hypothesis on the function f.) Clearly, $\lfloor \log_2 f(q) \rfloor > k$ for all prime powers $q > q_k$.

Now define, for each integer $k \geqslant 1$,

$$m_k = 2^{\lfloor \log_2 f(q_k) \rfloor} \operatorname*{lcm}_{\substack{q \leqslant q_k \\ q \in \mathscr{B}}} q.$$

Then $2^k \| m_k$, where by $r^a \| n$ we mean $r^a \mid n$ and $r^{a+1} \nmid n$. Note that $0^{m_k} \in \overline{L_f}$, since $m_k \equiv 0 \pmod{q}$ for $q \leqslant q_k$, and $m_k \not\equiv 0 \pmod{2^{k+1}}$.

Since $\overline{L_f}$ is regular, we may write

$$\overline{L_f} = \bigcup_{j \in A} (0^{t_j})^* 0^{s_j}$$

for some finite set A and nonnegative integers s_j, t_j. If $t_j = 0$, then $\overline{L_f}$ is finite, and the result follows immediately. Otherwise, assume $t_j \geqslant 1$. Since $0^{m_k} \in \overline{L_f}$, we may write $m_k = s_j + nt_j$ for some $j \in A$ and integer $n \geqslant 0$. We may assume that k is sufficiently large such that every nonzero t_j divides m_k. Define $n' = n + m_k/t_j$. Then $2m_k = s_j + n't_j$. Hence $0^{2m_k} \in \overline{L_f}$. Let r be the least odd prime power $> q_k$. Note that, by our hypothesis on the range of $\lfloor \log_2 f(p^e) \rfloor$, we have $\lfloor \log_2 f(r) \rfloor = k + 1$. Then $2m_k \not\equiv 0 \pmod{r}$ and $2m_k \equiv 0 \pmod{2^{\lfloor \log_2 f(r) \rfloor}}$. Thus $0^{2m_k} \in L_f$. This contradiction proves that L_f is not regular.

It remains to give an upper bound on the size of the smallest NFA accepting some nth-order approximation to L_f. First we prove the following lemma:

Lemma 29. *Let n be an integer $\geqslant 3$. Then the least odd prime power nondivisor of n is $\leqslant 5 \log n$.*

Proof. From the proof of Lemma 26, we know that

$$\psi'(x) := \sum_{\substack{p^k \leqslant x \\ p \geqslant 3}} \log p \geqslant 0.84x - \log x > 0.75x$$

for $x \geqslant 101$. For $3 \leqslant x \leqslant 101$, it can be verified by a short computation that $\psi'(x) \geqslant 0.21x$. Now if n has no odd prime power nondivisor $\leqslant 5 \log n$, it must be the case that n is divisible by all the odd prime powers $\leqslant 5 \log n$. Hence $n \geqslant e^{\psi'(5 \log n)} \geqslant e^{1.05 \log n} > n$, a contradiction. □

Now if $n \in A_f$, and $n \geqslant 2$, then $n \in S(q,k)$ for some odd prime power q. We claim that in fact there exists an odd prime power $q \leqslant 5 \log n$ for which $n \in S(q,k)$. For by Lemma 29, the least odd prime power q_0 which is a nondivisor of n is $\leqslant 5 \log n$. Let $k_0 = \lfloor \log_2 f(q_0) \rfloor$. If $2^{k_0} \nmid n$, then $n \notin S(q,k)$ for all $k \geqslant k_0$, and hence for all $q \geqslant q_0$. But

$q \mid n$ for all odd prime powers $q < q_0$, so $n \notin S(q,k)$ for all odd prime powers $q < q_0$. Therefore $n \notin A_f$, a contradiction. Hence $2^{k_0} \mid n$, and so $n \in S(q_0, k_0)$.

The total number of states needed to accept an nth-order approximation to L_f is therefore

$$1 + \sum_{\substack{q \leqslant 5 \log n \\ k = \lfloor \log_2 f(q) \rfloor}} q \cdot 2^k < f(5 \log n) \sum_{\substack{q \leqslant 5 \log n \\ q \in \mathscr{B}}} q$$

$$= f(5 \log n) O((\log n)^2 / (\log \log n)).$$

This completes the proof of Theorem 28. □

Acknowledgements

We would like to acknowledge with thanks the conversations with Eric Bach and Lisa Hellerstein. Some of the results in this paper were presented at the STACS 94 conference in Caen, France [32]. We express our thanks to Drew Vandeth and Dave Hamm, who read the manuscript with great care. Thanks also go to the seminonymous referee, who made several useful suggestions.

References

[1] E.W. Allender, On the number of cycles possible in digraphs with large girth, *Discrete Appl. Math.* **10** (1985) 211–225.

[2] J.-P. Allouche and M. Bousquet-Mélou, On the conjectures of Rauzy and Shallit for infinite words, *Comment. Math. Univ. Carolinae* **36** (1995) 705–711.

[3] H. Alt and K. Mehlhorn, A language over a one symbol alphabet requiring only o(log log n) space, *SIGACT News* **7**(4) (1975) 31–33.

[4] E. Bach, Explicit bounds for primality testing and related problems, *Math. Comp.* **55** (1990) 355–380.

[5] E. Bach and L. Huelsbergen, Statistical evidence for small generating sets, *Math. Comp.* **61** (1993) 69–82.

[6] J. Berstel, Mots de Fibonacci, in: *Séminaire d'Informatique Théorique*, Laboratoire Informatique Théorique, Institut Henri Poincaré (1980/1981) 57–78.

[7] J. Berstel, Fibonacci words – a survey, in: G. Rozenberg and A. Salomaa, eds., *The Book of L* (Springer, Berlin, 1986) 13–27.

[8] D.R.L. Brown, K.R. Davidson and J. Shallit, Elementary problem proposal 10433, *Amer. Math. Monthly* **102** (1995) 170.

[9] M. Chrobak, Finite automata and unary languages, *Theoret. Comput. Sci.* **47** (1986) 149–158.

[10] W.-F. Chuan, Extraction property of the golden sequence, *Fibonacci Quart.* **33** (1995) 113–122.

[11] A. Condon, L. Hellerstein, S. Pottle and A. Wigderson, On the power of finite automata with both nondeterministic and probabilistic states, in: *Proc. 26th Ann. ACM Symp. Theor. Comput.* (ACM, New York, 1994) 676–685.

[12] J. Dénes, K.H. Kim and F.W. Roush, Automata on one symbol, in: *Studies in Pure Mathematics: To the Memory of Paul Turán* (Birkhäuser, Basel, 1983) 127–134.

[13] C. Dwork and L. Stockmeyer, On the power of 2-way probabilistic finite state automata, in: *Proc. 30th Ann. Symp. Found. Comput. Sci.* (IEEE Press, New York, 1989) 480–485.

[14] C. Dwork and L. Stockmeyer, A time complexity gap for two-way probabilistic finite-state automata, *SIAM J. Comput.* **19** (1990) 1011–1023.

[15] W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. I (Wiley, New York, 1957).
[16] E. Garel, Conjecture 14 de Shallit et séparateurs, unpublished manuscript, August, 1995.
[17] I. Glaister and J. Shallit, Polynomial automaticity, context-free languages, and fixed points of morphisms, in: W. Penczek and A. Szałas, eds., *Proc. 21st Internat. Symp. Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol. 1113 (Springer, Berlin, 1996) 382–393.
[18] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, MA, 1979).
[19] J. Kaneps and R. Freivalds, Minimal nontrivial space complexity of probabilistic one-way Turing machines, In: B. Rovan, ed., *MFCS '90 (Mathematical Foundations of Computer Science)*, Lecture Notes in Computer Science, Vol. 452 (Springer, Berlin, 1990) 355–361.
[20] J. Kaneps and R. Freivalds, Running time to recognize nonregular languages by 2-way probabilistic automata, in: J. Leach Albert, B. Monien and M. Rodríguez Artalejo, eds., *ICALP '91 (18th International Colloquium on Automata, Languages, and Programming)*, Lecture Notes in Computer Science, Vol. 510 (Springer, Berlin, 1991) 174–185.
[21] J. Karhumäki, On cube-free ω-words generated by binary morphisms, *Discrete Appl. Math.* **5** (1983) 279–297.
[22] R.M. Karp, Some bounds on the storage requirements of sequential machines and Turing machines, *J. ACM* **14** (1967) 478–489.
[23] D.E. Knuth, *The Art of Computer Programming*, Vol. I: Fundamental Algorithms (Addison-Wesley, Reading, MA, 1973).
[24] S.R. Kosaraju, On independent circuits of a digraph, *J. Graph Theory* **1** (1977) 379–382.
[25] C.G. Lekkerkerker, Voorstelling van natuurlijke getallen door een som van getallen van Fibonacci, *Simon Stevin* **29** (1952) 190–195.
[26] Ju.I. Lyubich, Estimates of the number of states that arise in the determinization of a nondeterministic autonomous automaton, *Dokl. Akad. Nauk SSSR* **155** (1964) 41–43 (in Russian); an English translation appears in *Soviet Math.* **5** (1964) 345–348.
[27] Ju.I. Lyubich, Estimates for optimal determinization of nondeterministic autonomous automata, *Sibirskiĭ Matematicheskiĭ Zhurnal* **5** (1964) 337–355 (in Russian).
[28] Ju.I. Lyubich and E.M. Livshits, Estimates for the weight of a regular event over a 1-letter alphabet, *Sibirskiĭ Matematicheskiĭ Zhurnal* **6** (1965) 122–126 (in Russian).
[29] R. Mandl, Precise bounds associated with the subset construction on various classes of nondeterministic finite automata, in: *Proc. 7th Princeton Conf. on Information and System Sciences* (1973) 263–267.
[30] G. Rauzy, Suites à termes dans un alphabet fini, *Sém. de Théorie des Nombres de Bordeaux* (1982–1983) 25-01–25-16.
[31] J.B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers, *Illinois J. Math.* **6** (1962) 64–94.
[32] J. Shallit and Y. Breitbart, Automaticity: Properties of a measure of descriptional complexity, in: P. Enjalbert, E.W. Mayr and K.W. Wagner, eds., *STACS 94: 11th Annual Symp. Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Vol. 775 (Springer, Berlin, 1994) 619–630.
[33] J. Shallit and Y. Breitbart, Automaticity I: Properties of a measure of descriptional complexity, *J. Comput. System Sci.* **53** (1996) 10–25.
[34] C. Thomassen, On digraphs with no two disjoint cycles, *Combinatorica* **7** (1987) 145–150.
[35] H.C. Williams and J.O. Shallit, Factoring integers before computers, in: W. Gautschi, ed., *Mathematics of Computation, 1943–1993: A Half-Century of Computational Mathematics, Proc. Symposia Appl. Math.*, Vol. 48 (American Mathematical Society, Providence, RI, 1994) 481–531.
[36] E. Zeckendorf, Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas, *Bull. Soc. Royale des Sciences de Liège* **41**(3–4) (1972) 179–182.