Automaticity II: Descriptional Complexity in the Unary Case

Carl Pomerance*
Department of Mathematics
University of Georgia
Athens, GA 30601-3024 USA
carl@math.uga.edu

John Michael Robson
Department of Computer Science
Australian National University
Canberra, ACT 0200 Australia
jmr@cs.anu.edu.au

Jeffrey Shallit†
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
shallit@graceland.uwaterloo.ca

November 30, 1995

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 $\leq n$. 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, J. Shallit and Y. Breitbart examined the properties of this measure of descriptional complexity in the case $|\Sigma| \geq 2$. In this paper, we continue the study of automaticity, focusing on the case where $|\Sigma| = 1$.

*Research supported in part by DMS-9206784.

†Research supported in part by a grant from NSERC. Partial support under NSF Grant DCR 920-8639 and the Wisconsin Alumni Research Foundation.

We prove that $A_f(n) \leq n + 1 - \lfloor \log_\ell n \rfloor$, 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 R. Karp that $\limsup_{n\to\infty} A_L(n)/n \geq 1/2$. 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 [24], the third author and Y. Breitbart examined the notion of *automaticity*, a measure of descriptional complexity for functions and languages defined over finite alphabets Σ. Their work covered the case $k = |\Sigma| \geq 2$. In this paper we will examine the same notion, but concentrate on the unary case, when $k = |\Sigma| = 1$.

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

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:

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

Roughly speaking, $A_f(n)$ counts the minimum number of states in any DFAO M that simulates f correctly on all strings of length $\leq n$; 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,

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

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

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

We note that our model of nondeterministic finite automaton is that defined in [12], 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 $|w| \leq n$. Similarly, we say that a language $L \subseteq \Sigma^*$ is an *nth–order approximation* to a language $L' \subseteq \Sigma^*$ if we have $L \cap \Sigma^{\leq n} = L' \cap \Sigma^{\leq n}$.

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

2 Properties of Automaticity

In this section we recall from [24] 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 $A_L(n) \geq (n+3)/2$ for infinitely many n.*

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

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

Proof.

1. See [24, Thm. 3] or [14].

2. See [24, Thm. 7].

3. The inequality $A_L(n) \leq ce^{\sqrt{N_L(n)\log N_L(n)}}$ for some constant c follows from results in [8]. (See also [18, 19, 20, 21, 10].)

■

3 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 $\Sigma = \{0\}$. Oddly enough, this case does not seem to have been investigated previously.

In the unary case, the sequences $S_0(n), S_{00}(n)$, etc. are nothing more than the suffixes of the sequence $S_\epsilon(n)$. 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 $w = w(f) = f(\epsilon)f(0)f(0^2)f(0^3)\cdots$. We call $w(f)$ the *characteristic word* of f.

Lemma 2 *Let $\Sigma = \{0\}$, $\ell = |\Delta| \geq 2$, 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

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

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

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 $A_f(n) \leq n+1$; in fact, this bound can be attained for any *particular* value of n by setting $f(0^i) = 0$ for $0 \leq i < n$, and $f(0^n) = 1$. We have $S_\epsilon = 0^n 1$, 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 $\ell = |\Delta| \geq 2$. Then for any function $f : \Sigma^* \to \Delta$ the inequality $A_f(n) \leq n + 1 - \lfloor \log_\ell n \rfloor$ 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

$$S_\epsilon = f(\epsilon)f(0)f(0^2)\cdots f(0^n) = w_0 w_1 \cdots w_n.$$

Contained in the string S_ϵ are $\ell^m + 1$ (overlapping) factors of length m, where by *factor* we mean a string of consecutive symbols. 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 $w_k w_{k+1} \cdots w_{k+m-1}$, and let the second occurrence be $w_{n'+1-m} \cdots w_{n'}$. Then, by Lemma 2,

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

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

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

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 $O(\log n)$. This can be done as follows. First, we provide a solution when $\ell = 3$: we write down all possible binary strings of length 0, 1, 2, etc., separated by 2's:

$$220212002012102112000200120102011210021012110211120000 2 \cdots$$

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

To make this work with a binary alphabet, we simply recode: 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 $n - A_f(n) = O(\log n)$.

A construction improving the 4 to 2 was given independently by Condon, Hellerstein, Pottle, and Wigderson [9].

We now prove the following "almost all" result:

Theorem 4 *Suppose $\Sigma = \{0\}$ and $|\Delta| = \ell \geq 2$. 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

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

it follows that the number of functions from $\Sigma^{\leq n}$ to Δ that are given by DFAO's with $\leq q$ 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 $\Sigma^{\leq n}$ to Δ is ℓ^{n+1}, the probability that a randomly chosen function f satisfies $A_f(n) \leq c(n)$ is bounded above by

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

Since $\sum_{n \geq 2} \frac{1}{n(\log n)^2}$ converges, by the Borel-Cantelli lemma [11, p. 188], we must have $A_f(n) > c(n)$ for all sufficiently large n. ■

For languages, we immediately get the following corollary:

Corollary 5 *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.

4 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 6 *There are $O(q/\log q)^q$ 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. First, we introduce some notation borrowed from the computer language APL. We assume that $L \subseteq a^*$ is a unary language. We say that L is *c-monotonic* if, for all $n \geq 0$, 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 $n \geq N$, we have $a^n \in L$ iff $a^{n+c} \in L$. Note that if L is c-monotonic, then there exists a constant N such that L is c-periodic after N, but if L is c-periodic after N, it may not necessarily be c-monotonic.

Lemma 7 *(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 $1 \leq i \leq r$.

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 s.

Lemma 8 *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 $p_i = s$, 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 9 *Let M, s, and $L(M,s)$ be as in Lemma 8. Then $L(M,s)$ is c-periodic after $(c+1)(q-1)$.*

Proof. Suppose $w = a^\ell \in L = L(M)$, with $\ell \geq (c+1)q - 1$, 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 $k \geq 0$. The result will then follow from Lemma 8.

Let the accepting path for w be $\mathcal{P} = (p_0, p_1, \ldots, p_\ell)$, 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 $\geq c+2$. If any two of the occurrences of p' or two of the occurrences of s' in $\mathcal{P}$ are separated by a subpath $\mathcal{P}'$ of length $k \equiv 0 \pmod{c}$, then we obtain an accepting path for $a^{\ell-kc}$ by simply cutting out $\mathcal{P}'$.

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 $\mathcal{P}$ by d, and the section between the first occurrence of s' in S, and some later occurrence of s' in S, shortening $\mathcal{P}$ by e. Now consider the $\geq c$ possible values $d \pmod{c}$ and $-e \pmod{c}$. Since by Assumption A, no two of the possible choices for d are equal $(\mathrm{mod}\ c)$, the choices for d must be distributed in the *non-zero* residue classes $(\mathrm{mod}\ c)$. 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 $\geq c$. If G is acyclic, its girth is defined to be infinite.

Lemma 10 *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 [16]; also see [1, 26]. ■

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

Lemma 11 *Let M be a unary NFA with q states. Then there exists an integer $r \geq 0$, 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*

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

and, for $1 \leq i \leq r$, *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* $r \geq 1$, *then* $q' \geq c_r/2$. *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, $c_r \leq 2n_r/3$. Since $q' = n_r - c_r$, we have $q' \geq 3c_r/2 - c_r \geq c_r/2$. The fact that L_i is c_i-monotonic and c_i-periodic after $(c_r + 1)(q - 1)$ follows from Lemmas 7–9. ■

We are now ready to prove Theorem 6. The idea is to count the number of languages accepted by an NFA with q states by parameterizing the decomposition given in Lemma 11.

Proof. 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 $1 \leq i \leq r$ and $0 \leq j < c_i$, whether or not there exists an $n \geq 0$ with $n \equiv j \pmod{c_i}$ and $a^n \in L_i$;
3. For each pair (i, j), with $1 \leq i \leq r$ and $0 \leq j < c_i$, the cardinality of
$$L_{i,j} = \{a^n \in L_i - (\cup_{1 \leq t < i} L_t) \ : \ n < (c_r + 1)(q - 1) \text{ and } n \equiv j \pmod{c_i}\};$$
4. The residual language $L(M_{r+1})$.

First, let us argue that these specifications suffice. From Lemma 11, 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 (mod 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 (mod 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 $\cup_{1 \leq t < i} L_t$. 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 $c_1 < c_2 < \cdots < c_r \leq q$, 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 non-negative integers whose sum is $\leq m$ 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 $c_1 + c_2 + \cdots + c_r = q - q'$ non-negative 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 11 that $c_r \leq 2q'$, so, by Stirling's approximation,

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

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

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 10, 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 $B = L \cap \Sigma^{\geq q'}$. 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 $\mathcal{P}$ of the accepting path from v to p_f. Either $\mathcal{P}$ 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 $\geq q'$ can be split into three parts: an (i) 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 $\geq 2q'/3$.

   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 $O(q/\log q)^q$. ■

5 Bounds on Nondeterministic Automaticity: The Unary Case

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

Corollary 12 *Suppose $L \subseteq 0^*$. Then there exists a constant c such that for almost all L we have $N_L(n) > \frac{cn}{\log n}$ for all sufficiently large n.*

Proof. The result follows immediately from the Borel-Cantelli lemma and Theorem 6 of the previous section. ■

It is also easy to prove the following upper bound:

Theorem 13 *Let $L \subseteq 0^*$. Then $N_L(n) \leq n + 1 - \lfloor \log_2 n \rfloor$ for infinitely many n.*

Proof. This follows immediately from Theorem 3 and Theorem 1, Part 3. ■

6 Lower Bounds for Nonregular Languages when $k = 1$

Recall that Karp's theorem (Theorem 1, Part 1) says that if L is not regular, then $A_L(n) \geq (n+3)/2$ 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

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

However, the bound of $1/2$ does not seem attainable in the unary case. We make the following

Conjecture 14 *If $L \subseteq 0^*$ is not regular, then*

$$\limsup_{n \to \infty} \frac{A_L(n)}{n} \geq \frac{\sqrt{5}-1}{2} \doteq .61803.$$

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

Conjecture 14′ *Let $w = w_0 w_1 w_2 \ldots$ 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 $w_0 w_1 \ldots w_n$ that is also a factor of $w_0 w_1 \ldots w_{n-1}$. Then*

$$\liminf_{n \to \infty} \frac{s_w(n)}{n} \leq \frac{3-\sqrt{5}}{2} \doteq .38197.$$

J.-P. Allouche has kindly informed us that Conjecture 14′ is related to a similar one of G. Rauzy [22, §5.2]. This paper also mentions that Rauzy's conjecture has been proved by C. Rauzy in the case of the so-called Sturmian words.

We do not know how to prove Conjecture 14. 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* [5, 6].

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 $h_n \leq h_{n+1}$ for all $n \geq 2$, 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

$$f = f_0 f_1 f_2 \cdots = 0100101001001 \cdots .$$

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 $i \geq 2$.

There is a remarkable description of f in terms of *Fibonacci representations*. It is well known (see, for example, [17, 28]) that every integer $n \geq 0$ can be expressed uniquely as

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

where $a_i = a_i(n) \in \{0, 1\}$, and $a_i a_{i+1} = 0$ for all $i \geq 1$. We can think of the Fibonacci representation of n as an infinite string $a_1 a_2 a_3 \ldots$ where only finitely many of the a_i's are equal to 1; we write $n_{(F)} = a_1 a_2 a_3 \ldots$. Also, we define $\mathrm{fval}(a_1 a_2 \ldots a_k) = \sum_{1\leq i\leq k} a_k F_{k+1}$.

We have the following well-known theorem [15, Ex. 1.2.8.36].

Theorem 15 *Let n be a non-negative integer. Then $f_n = a_1(n)$.*

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

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

It is known that f is not ultimately periodic (this follows, for example, from Karhumäki's result [13] 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 $w_{a..b}$ we mean the word $w_a w_{a+1} \cdots w_b$. If $v = v_1 v_2 v_3 \cdots$ and $w = w_1 w_2 w_3 \cdots$ 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 16 *Let r, s be non-negative 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 $t = \mathrm{fval}(a_1(r)a_2(r)\cdots a_{k-1}(r))$.*

Proof.

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

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 $k \geq 2$. 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 $a_{1..k-1}(r) = 0^{k-1}$, $a_{1..k-1}(s) = 0^{k-2}1$, 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 $a_{1..k-2}(r'') = 0^{k-3}1$, $a_{1..k-2}(s'') = 0^{k-2}$, 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 $a_{1..k-3}(r''') = 0^{k-3}$ and $a_{1..k-3}(s''') = 0^{k-4}(1)$.

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

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

$$\begin{aligned} F_{k+1} - 1 &\leq m(r{\downarrow}f,\ s{\downarrow}f) \leq F_{k+2} - 2; \\ F_{k+1} &\leq d(r{\downarrow}f,\ s{\downarrow}f) \leq F_{k+2} - 1. \end{aligned}$$

Theorem 18 *Let $L = L_F$, the unary Fibonacci language. Suppose $F_n - 2 \leq k \leq F_{n+1} - 3$. Then $A_L(k) = F_{n-1}$.*

Proof. First we show that $A_L(k) \geq F_{n-1}$. 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 $S_\epsilon = f_{0..F_n-2}$, and $S_{0^i} = f_{i..F_n-2}$ for $0 \le i \le F_n - 2$. Define $x_i = f_{i..F_n-2}$. We then partition the collection $\{S_{0^i} \ : \ 0 \le i < F_{n-1}\}$ as follows:

$$\begin{aligned} D_1 &= \{x_i \ : \ 0 \le i \le F_{n-2}\} \\ D_2 &= \{x_i \ : \ F_{n-2} < i < F_{n-1}\} \end{aligned}$$

We will show that $D_1 \cup D_2$ consists of F_{n-1} pairwise incomparable strings 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 $\le$ 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 17, $d(i{\downarrow}f,\ j{\downarrow}f) \le F_{\ell+2}-1$, where $\ell = d(i_{(F)},\ j_{(F)})$. Since $i, j \le F_{n-2}$, it follows that $\ell \le n-3$. Hence we have $d(i{\downarrow}f,\ j{\downarrow}f) \le F_{n-1} - 1$, 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 $\le$. 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 $d(i_{(F)},\ j_{(F)}) \le n-5$. It follows from Corollary 17 that $d(i{\downarrow}f,\ j{\downarrow}f) \le F_{n-3} - 1$. But $|x_i|, |x_j| \ge (F_n-2)-(F_{n-1}-1)+1 = F_{n-2}$. 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 $0 \le i \le F_{n-2}$, and $F_{n-2} < j < F_{n-1}$. If $d(i{\downarrow}f,\ j{\downarrow}f) \le n-4$, 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 16 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 $A_L(k) \ge F_{n-1}$.

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

$$\begin{aligned} D &= \{y_i \ : \ 0 \le i < F_{n-1}\} \\ N &= \{y_i \ : \ F_{n-1} \le i \le F_{n+1} - 3\} \end{aligned}$$

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 $1 \le i \le F_n - 3$. But from Lemma 16, 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 19 *We have*

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

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

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

Hence

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

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

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

and so

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

■

In the last theorem of this section, we prove a somewhat stronger result than Conjecture 14 under a somewhat stronger hypothesis.

Theorem 20 *Let $a_1 < a_2 < a_3 < \cdots$ be a strictly increasing sequence of non-negative integers, and define*

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

Then

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

The constant $3/4$ cannot be replaced by any larger number.

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

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

Now it can be shown (see [7]) that for all sequences of positive real numbers $a_1, a_2, \ldots$ we have

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

From this, the first result follows.

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

7 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 21 *There exists a constant c' (which does not depend on L) such that if $L \subseteq 0^*$ is not regular, then $N_L(n) \geq c'(\log n)^2/(\log\log n)$ 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, Part 3, we have $A_L(n) < cn^{\sqrt{c'/2}}$ for all n sufficiently large. (Here c is the constant in Theorem 1, Part 3.) By choosing c' sufficiently small, we get a contradiction with Theorem 1, Part 1. ■

We now give a "natural" unary language with nondeterministic automaticity close to the lower bound in Theorem 21.

Theorem 22 *Let $L_1 = \{0^{n^2} : n \text{ odd}, \geq 1\}$. Then $L_2 = \overline{L_1} = 0^* - L_1$ is not a regular language. Assuming Conjecture 23 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 $\left(\frac{r}{p}\right) = -1$. Also define

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

Then the ERH implies that $J(m) < 3(\log m)^2$; see [3, 27].

The "reasonable conjecture" is the following:

Conjecture 23 *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, [4].

Proof of Theorem 22.

We construct an NFA M such that $L(M) \cap \Sigma^{\leq n} = L_2 \cap \Sigma^{\leq n}$ as follows: we "guess" an odd prime p and on input 0^j, compute $j \pmod p$ with a cyclic counter. If $\left(\frac{j}{p}\right) = -1$ (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 $9 + \sum_{2<p\leq J(n)} p$, which is $O((\log n)^2(\log\log n))$ assuming Conjecture 23, 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 24 *Define* $\theta(x) = \sum_{p \le x} \log p$. *Then* $\theta(x) > .23x$ *for* $x \ge 2$.

Proof. Rosser and Schoenfeld [23, Thm. 10] proved that $\theta(x) > .84x$ for $x \ge 101$. The stated inequality can now easily be verified for $2 \le x < 101$. ■

Theorem 25 *Define* $L_3 = \{0^n \ : \ n \ge 1$ 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 [2] 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 $p \ge 3$, $k \ge 1$ and $p^k \le 4.4 \log n$ such that $n \not\equiv 0 \pmod{p^k}$ and $n \equiv 0 \pmod{2^s}$ where $s \ge 0$ is an integer with $2^s < p^k < 2^{s+1}$;

(ii) if there exists a prime power p^k ($p \ge 3$, $k \ge 1$) such that $n \not\equiv 0 \pmod{p^k}$ and $n \equiv 0 \pmod{2^s}$ with $s \ge 1$ 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 $t \le 4.4 \log n$. Suppose not; then n is divisible by all the integers $\le 4.4 \log n$. Hence

$$n \ \ge \ \operatorname*{lcm}_{1 \le k \le 4.4 \log n} k = e^{\psi(4.4 \log n)} \ge e^{\theta(4.4 \log n)} > n,$$

a contradiction. (Here $\psi(x) = \sum_{p^k \le x} \log p$, and we have used Lemma 24.)

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 $n \not\equiv 0 \pmod{p^k}$ ($p \ge 3$, $k \ge 1$) and $n \equiv 0 \pmod{2^s}$ for $s \ge 1$ with $2^s < p^k < 2^{s+1}$. Let t be the least integer not dividing n. Then $t \le p^k$. However, since $n \equiv 0 \pmod{2^s}$ for all s with $2^s \le t$, 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 $p^k \le 4.4 \log n$, and then, on input 0^r (with $r \le n$), 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 $1 + \sum_{p^k \le 4.4 \log n} O((p^k)^2)$ states, which is $O((\log n)^3/(\log\log n))$. ■

Our last result is the following: define

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

Define

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

the set of odd prime powers. Given a function $f : \mathbb{Z}^{\ge 3} \to \mathbb{R}^{\ge 1}$, define the set A_f as follows:

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

Then we have

Theorem 26 *Let $f : \mathbb{Z}^{\geq 3} \to \mathbb{R}^{\geq 1}$ be any (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 $e \geq 1$. 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 $x \leq 2$, and $\lg x = \log_2 x$ if $x > 2$. Also, $\lg^{(1)} x = \lg x$, and $\lg^{(i)} = \lg \lg^{(i-1)} x$ for $i \geq 2$. 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 $k \geq 1$,

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

Then $2^k \,||\, m_k$. Note that $0^{m_k} \in \overline{L_f}$, since $m_k \equiv 0 \pmod{q}$ for $q \leq q_k$, 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 non-negative integers s_j, t_j. If $t_j = 0$, then $\overline{L_f}$ is finite, and the result follows immediately. Otherwise, assume $t_j \geq 1$. Since $0^{m_k} \in \overline{L_f}$, we may write $m_k = s_j + n t_j$ for some $j \in A$ and integer $n \geq 0$. We may assume that k is sufficiently large such that every non-zero 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 27 *Let n be an integer ≥ 3. Then the least odd prime power nondivisor of n is $\leq 5 \log n$.*

Proof. From the proof of Lemma 24, we know that

$$\psi'(x) = \sum_{\substack{p^k \leq x \\ p \geq 3}} \log p \geq .84x - \log x > .75x$$

for $x \geq 101$. For $3 \leq x \leq 101$, it can be verified by a short computation that $\psi'(x) \geq .21x$. Now if n has no odd prime power nondivisor $\leq 5 \log n$, it must be the case that n is divisible by all the odd prime powers $\leq 5 \log n$. Hence $n \geq e^{\psi'(5 \log n)} \geq e^{1.05 \log n} > n$, a contradiction. ■

Now if $n \in A_f$, and $n \geq 2$, then $n \in S(q,k)$ for some odd prime power q. We claim that in fact there exists an odd prime power $q \leq 5 \log n$ for which $n \in S(q,k)$. For by Lemma 27, the least odd prime power q_0 which is a nondivisor of n is $\leq 5 \log n$. Let $k_0 = \lfloor \log_2 f(q_0) \rfloor$. If $2^{k_0} \nmid n$, then $n \notin S(q,k)$ for all $k \geq k_0$, and hence for all $q \geq q_0$. 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

$$\begin{aligned} 1 + \sum_{\substack{q \leq 5 \log n \\ k = \lfloor \log_2 f(q) \rfloor}} q \cdot 2^k &< f(5 \log n) \sum_{\substack{q \leq 5 \log n \\ q \in \mathcal{B}}} q \\ &= f(5 \log n) O((\log n)^2 / (\log \log n)). \end{aligned}$$

This completes the proof of Theorem 26. ■

8 Acknowledgments

We would like to acknowledge with thanks conversations with Eric Bach and Lisa Hellerstein. Some of the results in this paper were presented at the STACS 94 conference in Caen, France [25]. Drew Vandeth read the manuscript with great care.

References

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

[2] 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.

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

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

[5] J. Berstel. Mots de Fibonacci. In *Séminaire d'Informatique Théorique*, pages 57–78. Laboratoire Informatique Théorique, Institut Henri Poincaré, 1980/81.

[6] J. Berstel. Fibonacci words—a survey. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 13–27. Springer-Verlag, 1986.

[7] D. Brown, K. Davidson, and J. Shallit. Elementary problem proposal 10433. *Amer. Math. Monthly*, problem section, to appear, February 1995.

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

[9] A. Condon, L. Hellerstein, S. Pottle, and A. Wigderson. On the power of finite automata with both nondeterministic and probabilistic states. Manuscript in preparation, 1993.

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

[11] W. Feller. *An Introduction to Probability Theory and its Applications*, Vol. I. John Wiley & Sons, New York, 1957.

[12] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[13] J. Karhumäki. On cube-free ω-words generated by binary morphisms. *Disc. Appl. Math.* **5** (1983), 279–297.

[14] R. M. Karp. Some bounds on the storage requirements of sequential machines and Turing machines. *J. Assoc. Comput. Mach.* **14** (1967), 478–489.

[15] D. E. Knuth. *Fundamental Algorithms*, Vol. I of *The Art of Computer Programming.* Addison-Wesley, Reading, Mass., 1973.

[16] S. R. Kosaraju. On independent circuits of a digraph. *J. Graph Theory* **1** (1977), 379–382.

[17] C. G. Lekkerkerker. Voorstelling van natuurlijke getallen door een som van getallen van Fibonacci. *Simon Stevin* **29** (1952), 190–195.

[18] 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. English translation in *Soviet Mathematics* **5** (1964), 345–348.

[19] Ju. I. Lyubich. Estimates for optimal determinization of nondeterministic autonomous automata. *Sibirskiĭ Matematicheskiĭ Zhurnal* **5** (1964), 337–355. In Russian.

[20] 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.

[21] R. Mandl. Precise bounds associated with the subset construction on various classes of nondeterministic finite automata. In *Proc. 7th Princeton Conference on Information and System Sciences*, pages 263–267. 1973.

[22] G. Rauzy. Suites à termes dans un alphabet fini. *Sém. de Théorie des Nombres de Bordeaux* (1982-1983), 25–01–25–16.

[23] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Ill. J. Math.* **6** (1962), 64–94.

[24] J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. Submitted, 1994.

[25] J. Shallit and Y. Breitbart. Automaticity: Properties of a measure of descriptional complexity. In P. Enjalbert, E. W. Mayr, and K. W. Wagner, editors, *STACS 94: 11th Annual Symposium on Theoretical Aspects of Computer Science*, Vol. 775 of *Lecture Notes in Computer Science*, pages 619–630. Springer-Verlag, 1994.

[26] C. Thomassen. On digraphs with no two disjoint cycles. *Combinatorica* **7** (1987), 145–150.

[27] H. C. Williams and J. O. Shallit. Factoring integers before computers. Mathematics of Computation, 1943–1993: A half-century of computation on mathematics, *Proc. Symp. Appl. Math.*, to appear.

[28] 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.