AUTOMATICITY III: POLYNOMIAL AUTOMATICITY AND CONTEXT-FREE LANGUAGES

Ian Glaister and Jeffrey Shallit

Abstract. If L is a formal language, we define $A_L(n)$ to be the number of states in the smallest deterministic finite automaton that accepts a language that agrees with L on all inputs of length $\leq n$. This measure is called *automaticity.* In this paper, we first study the closure properties of the class DPA of languages of deterministic polynomial automaticity, i.e., those languages L for which there exists k such that $A_L(n) = O(n^k)$. Next, we discuss similar results for a nondeterministic analogue of automaticity, introducing the classes NPA (languages of nondeterministic polynomial automaticity) and NPLA (languages of nondeterministic poly-log automaticity). We conclude by showing how to construct a context-free language of automaticity arbitrarily close to the maximum possible.

Key words. automaticity, finite automata, nondeterminism
Subject classifications. Primary 68Q68; Secondary 68Q75 68Q45.

1. Introduction.

In two previous papers (Shallit & Breitbart (1996), Pomerance *et al.* (1996)), the second author and co-authors studied the concept of *automaticity*: roughly speaking, how closely a formal language L can be approximated by regular languages L'; also see Shallit & Breitbart (1994). In addition to its evident intrinsic interest, automaticity has proved useful in obtaining nontrivial lower bounds. For example, in Dwork & Stockmeyer (1989), Dwork & Stockmeyer (1990), and Kaneps & Freivalds (1991) the measure was used to obtain lower bounds on computation by two-way probabilistic finite automata. In Kaneps & Freivalds (1990) it was used to obtain lower bounds on the space complexity of probabilistic Turing machines.

In this paper, the third of a series, we introduce three new automaticity-based complexity classes, and study their properties. Additional results on automaticity can be found in Shallit (1996).

As usual, we define a finite automaton M to be a 5-tuple, $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *states*, Σ is a finite *input alphabet*, q_0 is the *start state*, and F is a set of *final states*. The map δ is called the *transition function*. If M is *deterministic*, then δ maps $Q \times \Sigma$ to Q, and is extended in the usual way to a map $Q \times \Sigma^*$ to Q. We then define $L(M)$, the *language accepted by* M, to be the set $\{x \in \Sigma^* \ : \ \delta(q_0, x) \in F\}$. If M is *nondeterministic*, then δ maps $Q \times \Sigma$ to 2^Q, and we define $L(M) = \{\delta(q_0, x) \cap F \neq \emptyset\}$. For more on these concepts, see, for example, Hopcroft & Ullman (1979).

We denote the class of all deterministic finite automata by DFA, and the class of all nondeterministic finite automata by NFA. By $|M|$ we mean $|Q|$, the number of states in the machine M.

Let ϵ denote the empty string, and let $\Sigma^{\leq n} = \epsilon + \Sigma + \Sigma^2 + \cdots + \Sigma^n$, the set of all strings over Σ of length at most n. Let L, L' be languages with $L, L' \subseteq \Sigma^*$. If $L \cap \Sigma^{\leq n} = L' \cap \Sigma^{\leq n}$, we say that L' is an *nth order approximation* to L.

Given a language $L \subseteq \Sigma^*$, we define the function $A_L(n)$, the *deterministic automaticity* of L, as follows:

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

Informally, $A_L(n)$ counts the number of states in the smallest finite automaton that accepts some nth order approximation to L.

Similarly, we define $N_L(n)$, the *nondeterministic automaticity* of L, as follows:

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

We now introduce three new complexity classes:

1. deterministic polynomial automaticity, or DPA:

$$\text{DPA} = \{L \subseteq \Sigma^* \ : \ \exists k \text{ such that } A_L(n) = O(n^k)\}.$$

2. nondeterministic polynomial automaticity, or NPA:

$$\text{NPA} = \{L \subseteq \Sigma^* \ : \ \exists k \text{ such that } N_L(n) = O(n^k)\}.$$

3. nondeterministic poly-log automaticity, or NPLA:

$$\text{NPLA} = \{L \subseteq \Sigma^* \ : \ \exists k \text{ such that } N_L(n) = O((\log n)^k)\}.$$

(We do not define deterministic poly-log automaticity (DPLA) because if $A_L(n) = O((\log n)^k)$, then by a theorem of Karp (1967), $A_L(n) = O(1)$ and L is regular.)

There are clear analogies of these classes with more traditional ones such as P, NP, and NC. In this paper, we first discuss the closure properties of these new classes.

It is perhaps worth pointing out that, unlike the classes P and NP, and as a consequence of the non-uniformity of the model, these classes contain uncountably many languages. If this troubles the reader, one can restrict one's attention to the recursive languages in these classes without altering any of the results in this paper.

2. Results on DPA.

In this section, we discuss the properties of the class DPA. First, we show that DPA consists of a strict hierarchy of complexity classes. Second, we study the closure properties of DPA.

It is easy to see (Shallit & Breitbart (1996)) that if L is a unary language (i.e., L is defined over an alphabet with one letter), then $A_L(n) = O(n)$. In this case, DPA is trivially closed under every operation (e.g., union, concatenation, Kleene closure, etc.). Hence, for the remainder of this section and the next one, we assume that $|\Sigma| \geq 2$.

First, we state three useful lemmas. Let $L \subseteq \Sigma^*$, and let S be a finite set of strings over Σ. Suppose for all $x, y \in S$, there exists a $w \in \Sigma^*$ such that $|xw|, |yw| \leq n$, and exactly one of xw, yw is in L. Then we call S an *n-dissimilar set* of strings for L.

LEMMA 2.1 (KANEPS AND FREIVALDS). *$A_L(n)$ is equal to the cardinality of the largest n-dissimilar set of strings for L.*

PROOF. See Kaneps & Freivalds (1990), Shallit & Breitbart (1996). □

LEMMA 2.2. *For $n \geq 0$ we have*

$$A_L(n) \leq 2 + \sum_{w \in L \cap \Sigma^{\leq n}} |w| \leq 2 + n|L \cap \Sigma^{\leq n}|.$$

PROOF. See Shallit & Breitbart (1996), Theorem 2, Part 5. □

LEMMA 2.3. *The language L is regular if and only if $A_L(n) = O(1)$. The same statement holds for $N_L(n)$.*

PROOF. See Shallit & Breitbart (1996), Theorem 2, Part 2. $\square$

Lemma 2.1 is extremely useful in obtaining lower bounds on deterministic automaticity, as the following theorem shows.

THEOREM 2.4. *For all integers $k \geq 0$, there is a language L_k such that $A_{L_k}(n) = \Theta(n^k)$.*

PROOF. For L_0 we may take any regular language, by Lemma 2.3.

Now let k be an integer ≥ 1, and define

$$L_k = \{0^{a_1}\,1\,0^{a_2}\,1\,\cdots 0^{a_k}\,1\,0^{a_1}\,1\,0^{a_2}\,1\,\cdots 0^{a_k}\,1 \;:\; a_1, \ldots, a_k \geq 0\}.$$

Let $n' = \lfloor n/2 \rfloor$. We will show that

$$\binom{n'}{k} \leq A_{L_k}(n) \leq \binom{n'}{k} + 2\binom{n'+1}{k} - 1,$$

and hence we have $A_{L_k}(n) = \Theta(n^k)$, where the constant implied in the Θ depends on k.

First, the lower bound. Let

$$S = S(n,k) = \{0^{a_1}\,1\,\cdots 0^{a_k}\,1 \;:\; a_1, \ldots, a_k \geq 0 \text{ and } a_1 + \cdots + a_k + k \leq n'\}.$$

Then S is an n-dissimilar string set for L_k, since for each $v, w \in S$, we have $vv \in L_k$ and $vw \notin L_k$, and $|vv| = |vw| = 2n' \leq n$. To determine $|S|$, consider the number of ways to insert k 1's into a list of n' 0's, and let the number of 0's between two consecutive occurrences of the 1's correspond to the a_i. It follows that $|S| = \binom{n'}{k}$.

The lower bound on $A_{L_k}(n)$ now follows from Lemma 2.1.

For the upper bound, we construct an automaton $M = M_{n,k}$ that accepts an nth order approximation to L_k. The basic idea is that M accumulates the first half of the string in its state, symbol by symbol, and then, when the second half is encountered, the symbols are matched with the stored word.

More formally, define $\mathrm{ppref}(S)$, the set of proper prefixes of elements of S, as follows:

$$\mathrm{ppref}(S) = \{x \in (0+1)^* \;:\; \exists w \in (0+1)^+ \text{ with } xw \in S\}.$$

Now define $M_{k,n} = (Q, \Sigma, \delta, q_0, F)$, where

$$Q = \{[w]^+, [w]^- \ : \ w \in \text{ppref}(S)\} \ \cup \ \{[w]^+ \ : \ w \in S\} \ \cup \ \{d\},$$

$q_0 = [\epsilon]^+$, $F = \{[\epsilon]^-\}$, and δ is defined as follows:

$$\begin{aligned}
\delta([w]^+, a) &= [wa]^+, \quad \forall a \in \Sigma, \ w \in \text{ppref}(S) \\
\delta([aw]^+, a) &= [w]^-, \quad \forall a \in \Sigma, \ aw \in S \\
\delta([aw]^-, a) &= [w]^-, \quad \forall a \in \Sigma, \ aw \in \text{ppref}(S) \\
\delta(d, a) &= d, \quad \forall a \in \Sigma \\
\delta(x, a) &= d, \quad \text{for all other cases.}
\end{aligned}$$

The correctness proof is left to the reader.

It follows that $|Q| = 2|\text{ppref}(S)| + |S| + 1$, so it remains to compute $|\text{ppref}(S)|$. We claim that $|\text{ppref}(S)| = \binom{n'+1}{k} - 1$.

Define $|w|_a$ to be the number of occurrences of the symbol a in the string w. Let $s \in \text{ppref}(S)$. If $|s|_1 = r$, then $r \leq |s| \leq n' - k + r$, for otherwise s could not be a prefix of a string of length $\leq n$ with exactly k 1's. It follows that there are

$$\binom{r}{r} + \binom{r+1}{r} + \cdots + \binom{n'-k+r}{r} = \binom{n'-k+r+1}{r+1}$$

such strings. Summing this from $r = 0$ to $r = k-1$ gives

$$|\text{ppref}(S)| = \sum_{0 \leq r \leq k-1} \binom{n'-k+r+1}{r+1} = \binom{n'+1}{k} - 1,$$

as claimed. This completes the proof. □

Now we move on to the closure properties of DPA.

THEOREM 2.5. *The class* DPA *is closed under union, intersection, complement, and inverse homomorphism.*

PROOF. The usual product constructions (see, e.g., Hopcroft & Ullman (1979), pp. 59–60) show immediately that $A_{L_1 \cup L_2}(n) \leq A_{L_1}(n) A_{L_2}(n)$ and $A_{L_1 \cap L_2}(n) \leq A_{L_1}(n) A_{L_2}(n)$.

By interchanging the accepting and non-accepting states in an automaton accepting an nth order approximation to L, we get an automaton accepting an nth order approximation to $\overline{L}$; hence $A_L(n) = A_{\overline{L}}(n)$.

To show DPA is closed under inverse homomorphism, it suffices to adapt the standard construction (e.g., Hopcroft & Ullman (1979), p. 61). Let $L \in \Sigma^*$ be a member of DPA, and let $h : \Delta^* \to \Sigma^*$ be a homomorphism. Define $m = \max_{a \in \Delta} |h(a)|$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be an automaton accepting an mnth order approximation to L. Then consider $M' = (Q, \Delta, \delta', q_0, F)$, where $\delta'(q, a) = \delta(q, h(a))$. Clearly M' accepts an nth order approximation to $h^{-1}(L) = \{x \in \Delta^* \ : \ h(x) \in L\}$. It follows that $A_{h^{-1}(L)}(n) \leq A_L(mn)$. $\square$

Next, consider the operation $\frac{1}{2}L$ defined as follows:

$$\frac{1}{2}L = \{x \in \Sigma^* \ : \ \exists y \text{ such that } |x| = |y| \text{ and } xy \in L\}.$$

THEOREM 2.6. DPA *is closed under the operation* $\frac{1}{2}$.

PROOF. We prove the contrapositive. Assume $\frac{1}{2}L \notin$ DPA. Then, for any given $c > 1$, the inequality

$$A_{\frac{1}{2}L} > (n+1)^{c+1} \tag{2.1}$$

must hold for infinitely many n. Let n be such that (2.1) holds; then there is an n-dissimilar string set R for $\frac{1}{2}L$ of cardinality $> (n+1)^{c+1}$. By the pigeonhole principle, there exists m, $0 \leq m \leq n$, such that $|R \cap \Sigma^m| > (n+1)^c$. Set $R' = R \cap \Sigma^m$; note that R' is also an n-dissimilar string set for $\frac{1}{2}L$.

By the definition of n-dissimilar string set, for all $u, v \in R'$ with $u \neq v$, there exists w such that exactly one of uw, vw are in $\frac{1}{2}L$, and also $|uw|, |vw| \leq n$. Without loss of generality, assume $uw \in \frac{1}{2}L$ and $vw \notin \frac{1}{2}L$. Since $uw \in \frac{1}{2}L$, there exists z with $|z| = |uw|$ such that $uwz \in L$. If $vwz \in L$, then we would have $vw \in \frac{1}{2}L$, a contradiction. Hence $vwz \notin L$. It follows that u, v are $2n$-dissimilar for L, and the suffix wz distinguishes them. Hence $A_L(2n) \geq |R'| > (n+1)^c$. But c was arbitrary, and so $L \notin$ DPA. $\square$

Now we turn to properties under which DPA is not closed. There is a vague relation between these questions and the notion of "state complexity" of regular languages, discussed, for example, in Yu *et al.* (1994). In that paper, the authors showed, among other things, that for each $m, n \geq 1$, there exist an m-state DFA and an n-state DFA such that the minimal automaton for the concatenation of the corresponding languages has $m2^n - 2^{n-1}$ states. At first glance, it would seem that this result immediately implies that DPA is not closed under concatenation. This is not so, however, because the Yu *et al.* result constructs a *different pair* of languages for each pair m, n. As $m, n \to \infty$, the languages they construct tend to a "limiting language" that is actually regular.

To prove that DPA is not closed under concatenation, we need something different: namely, two DPA languages L_1, L_2 such that any nth order approximation to $L_1 L_2$ requires many states.

First, we prove the following lemma:

LEMMA 2.7. *Let $(r_k)_{k\geq 0}$ be any sequence of integers satisfying $r_0 = 1$ and $r_{k+1} \geq 2r_k$. Define $L = \{(0+1)^* 1 (0+1)^{r_k} \ : \ k \geq 0\}$. Define $n_k = r_k + r_{k-1}$ for $k \geq 1$. Then $A_L(n_k) \geq 2^{r_{k-1}}$.*

PROOF. Define $R = \{r_0, r_1, r_2, \ldots\}$. Let w, x be two distinct strings in $(0+1)^{r_{k-1}}$. Without loss of generality, we may write

$$\begin{aligned} w &= w_1 w_2 \cdots w_{t-1} 1 w_{t+1} \cdots w_{r_{k-1}} \\ x &= x_1 x_2 \cdots x_{t-1} 0 w_{t+1} \cdots x_{r_{k-1}} \end{aligned}$$

for some t with $1 \leq t \leq r_{k-1}$. Let $y = 0^{r_k + t - r_{k-1}}$. We have

$$wy \in (0+1)^{t-1} 1 (0+1)^{r_k},$$

so $wy \in L \cap \Sigma^{\leq n_k}$. On the other hand, we claim that $xy \notin L$. If it were, then there would be a suffix of xy of the form $1(0+1)^\ell$, for some $\ell \in R$. The first 1 in this suffix must come from x. Without loss of generality, let it be x_s, so $x = x_1 x_2 \cdots x_{s-1} 1 x_{s+1} \cdots x_{r_{k-1}}$, for some $s \neq t$. In this case we have $\ell = r_k + t - s \in R$. But either (i) $1 \leq s \leq t-1$, or (ii) $t+1 \leq s \leq r_{k-1}$. In case (i), we have $r_k + 1 \leq \ell \leq r_k + t - 1 < r_{k+1}$. In case (ii), we have $t \neq r_{k-1}$ and so $r_{k-1} < r_k - r_{k-1} + t \leq \ell \leq r_k - 1$. Both cases contradict $\ell \in R$.

It follows that $(0+1)^{r_{k-1}}$ forms a set of n_k-dissimilar strings. Hence $A_L(n_k) \geq 2^{r_{k-1}}$. $\square$

THEOREM 2.8. *The class* DPA *is not closed under concatenation.*

PROOF. Let $L_1 = (0+1)^*$, and $L_2 = \{1(0+1)^{2^k} \ : \ k \geq 0\}$. Then it is easy to see that $A_{L_1}(n) = O(1)$, and $A_{L_2}(n) = O(n)$.

Now define $L = L_1 L_2$, and put $r_k = 2^k$ in Lemma 2.7. Then $n = 2^k + 2^{k-1} = 3 \cdot 2^{k-1}$. It follows that $A_L(n) \geq 2^{k-1}$. Thus, for infinitely many n, we have $A_L(n) \geq 2^{n/3}$, and so $L \notin$ DPA. $\square$

THEOREM 2.9. *The class* DPA *is not closed under reversal.*

PROOF. Let $L' = \{(0+1)^{2^k} 1 (0+1)^* \ : \ k \geq 0\}$. It is easy to see that $A_{L'}(n) = O(n)$. On the other hand, $L'^R = L$, where $L = \{(0+1)^* 1 (0+1)^{2^k} \ : \ k \geq 0\}$. From the previous theorem, $L \notin$ DPA. □

We now recall the definition of the quotient of two languages (Hopcroft & Ullman (1979), p. 62). Let $L_1, L_2 \subseteq \Sigma^*$. Then

$$L_1/L_2 = \{x \in \Sigma^* \ : \ \text{there exists } y \in L_2 \text{ such that } xy \in L_1\}.$$

THEOREM 2.10. *The class* DPA *is not closed under quotient by regular sets.*

PROOF. Let $\Sigma = \{0,1\}$, $L = \{ww\,1\,0^{2^k} \ : \ w \in (0+1)^k, \ k \geq 0\}$, and $R = 1\,0^*$. By Lemma 2.2, we know that

$$\begin{aligned}
A_L(n) &\leq 2 + \sum_{w \in L \cap \Sigma^{\leq n}} |w| \\
&\leq 2 + \sum_{0 \leq j \leq n} j |L \cap \Sigma^j| \\
&\leq 2 + \sum_{2^k + 2k + 1 \leq n} (2^k + 2k + 1) 2^k \\
&\leq 2 + \sum_{0 \leq k \leq \log_2 n} (2^k + 2k + 1) 2^k \\
&= O(n^2).
\end{aligned}$$

Now $L/R = \{ww \ : \ w \in (0+1)^*\}$. Let $S = S_n = \{w \in (0+1)^* \ : \ |w| = \lfloor n/2 \rfloor\}$. Then it is easy to see that S_n is an n-dissimilar string set for L/R, and so $A_{L/R}(n) = \Omega(2^{n/2})$. □

THEOREM 2.11. *The class* DPA *is not closed under Kleene closure.*

PROOF. Let $\Sigma = \{0,1\}$, and define $L = \{1(0+1)^{k^2-1} \ : \ k \geq 2\}$ We will show that $L \in$ DPA and $L^* \notin$ DPA.

It is easy to see that $A_L = O(n)$, for we can accept an nth order approximation to L with a linear chain of nodes.

We now show that $A_{L^*}(n) \geq n^{n^{1/8}/8}$ for all n sufficiently large.

First, we introduce some definitions. We say that a string $w \in (0+1)^*$ is *valid for position* j if there exists a way to write $w = w_1 w_2 \cdots w_r$, where $r \geq 1$, $w_i \in (0+1)^+$ for $1 \leq i \leq r$, $w_i \in L$ for $1 \leq i < r$, the first symbol of w_r is 1, and $|w_r| = j$. Note that a word may be valid for no positions, or for several.

As an example, the string 100010000000010001 is valid for positions $1, 5, 14, 18$. Then it is easy to see that $w \in L^+$ if and only if there exists a $k > 1$ such that w is valid for position k^2.

Next, let $S = \{s_1, s_2, \ldots, s_k\}$ be a nonempty set of positive integers. If S satisfies the following two conditions, then we call it *good*:

(a) $s_1 = 1$;

(b) for all i with $2 \le i \le k$, there exists an integer $t_i > 1$ such that $s_i - s_{i-1} = t_i^2$.

We call s_k the *weight* of the set S, k the *size* of the set S, and

$$\max_{2 \le i \le k} (s_i - s_{i-1})$$

the *span* of the set S.

If S is good, then there is a word $w = w(S)$ of length equal to the weight of S, such that w is valid exactly for the positions specified by S. Namely, we can take

$$w = 10^{s_k - s_{k-1} - 1} 10^{s_{k-1} - s_{k-2} - 1} 1 \cdots 10^{s_2 - s_1 - 1} 10^{s_1 - 1}.$$

Note that the map that sends S to $w(S)$ is injective.

Now suppose S and T are different good sets of weight at most m. Then we claim there exists a word y with $|y| \le \lceil (m+2)/2 \rceil^2$ such that $w(S)y \in L^+$ and $w(T)y \notin L^+$. Since S and T are different, without loss of generality there exists c such that $c \in S$ and $c \notin T$. Choose $y = 0^{\lceil (m+2)/2 \rceil^2 - c}$. Then $w(S)$ is valid for position c, so $w(S)y$ is valid for position $\lceil (m+2)/2 \rceil^2$. Hence $w(S)y \in L^+$.

On the other hand, suppose $w(T)$ is valid for a position p. Then $p \ne c$ and $1 \le p \le m$. Thus if $w(T)y$ is valid for a position q, we must have

$$\lceil (m+2)/2 \rceil^2 - c + 1 \le q \le \lceil (m+2)/2 \rceil^2 - c + m.$$

This implies that

$$\lceil (m+2)/2 \rceil^2 - m + 1 \le q \le \lceil (m+2)/2 \rceil^2 + m. \tag{2.2}$$

If m is even, say $m = 2t$, then (2.2) implies that $t^2 < q < (t+2)^2$. If $w(T)y \in L^+$, q must be a square, and so $q = (t+1)^2$. But then $w(T)$ is valid for position c, a contradiction. Similarly, if m is odd, say $m = 2t+1$, then (2.2) implies that $(t+1)^2 < q < (t+3)^2$. Then $q = (t+2)^2$, and $w(T)$ is valid for position c, a contradiction. It follows that $w(T)y \notin L^+$.

Now fix an n, and define the collection $\mathcal{C}_n$ to be those good sets S of span at most $(n^{1/8} + 2)^2$ and size $k = \lceil n^{1/8} \rceil + 1$. Each set has weight at most $1 + (n^{1/8} + 2)^2 (k-1) = O(n^{3/8})$.

Now define

$$U = \{w(S) \ : \ S \in \mathcal{C}_n\}.$$

The cardinality of U is the cardinality of $\mathcal{C}_n$, which corresponds to the number of possible choices of S with the given span and size. Each of the $k-1$ possible differences $s_i - s_{i-1}$ can be any one of at least $n^{1/8}$ possible squares, and so there are $(n^{1/8})^{n^{1/8}}$ possibilities for S. We claim that, for all n sufficiently large, U is an n-dissimilar string set for L^+ (and hence, for L^*). This is clear, since by the reasoning above, if S and T are different elements of $\mathcal{C}_n$, then $w(S)$ and $w(T)$ are distinguishable by a string y whose length is $O((n^{3/8})^2) = O(n^{3/4})$. Hence $|w(S)y| = O(n^{3/4} + n^{3/8})$, and the same length bound holds for $w(T)y$. It therefore follows that $A_{L^*}(n) \geq |U| \geq (n^{1/8})^{n^{1/8}}$, and so $L^* \notin$ DPA. □

THEOREM 2.12. *If* $|\Sigma| \geq 3$, *then* DPA *is not closed under homomorphism.*

PROOF. Let $L = \{(0+1)^* \, 2 \, (0+1)^{2^k} \ : \ k \geq 0\}$, and define $h(0) = 0$, $h(1) = h(2) = 1$. Then it is easy to see that $A_L(n) = O(n)$, but $h(L)$ is not in DPA by Theorem 2.8. □

3. Results on NPA.

In this section, we obtain results on languages of nondeterministic polynomial automaticity: the class NPA.

We start by applying notions of communication complexity (e.g., Yao (1979), Aho *et al.* (1983), Condon *et al.* (1994)) to the computation of nondeterministic automaticity. Let U be a finite set of strings. Then we say that U is a set of *uniformly n-dissimilar* strings if for each string $u \in U$ there exists a string w such that

(i) $|uw| \leq n$ and $uw \in L$; and

(ii) for every string $v \in U$ such that $u \neq v$, we have $|vw| \leq n$ and $vw \notin L$.

We sometimes call the string w a *witness* for u.

Then we have the following

LEMMA 3.1. *If U is a set of uniformly n-dissimilar strings for L, then $N_L(n) \geq |U|$.*

Proof. Consider a string $u \in U$. By the definition, there exists a witness string w satisfying conditions (i) and (ii). Let $M = (Q, \Sigma, \delta, q_0, F)$ be any nondeterministic finite automaton that accepts an nth order approximation to L. Now $uw \in L$, and since M accepts all strings in L of length $\leq n$, we have $\delta(q_0, uw) \cap F \neq \emptyset$. Hence there exists at least one state $q \in \delta(q_0, u)$ such that $p \in \delta(q, w)$, where $p \in F$.

However, for every other string $v \in U$, with $v \neq u$, we must have $q \notin \delta(q_0, v)$. For if $q \in \delta(q_0, v)$, we would have $p \in \delta(q_0, vw)$ and so $vw \in L$, a contradiction (since $|vw| \leq n$).

Hence every set $\delta(q_0, u)$ contains a state q which does not appear in any other set $\delta(q_0, v)$ for $u \neq v$. It follows that there must be at least $|U|$ different states in Q. □

This simple, but powerful, lemma will allow us to estimate the nondeterministic automaticity for a wide variety of languages; see below. However, unlike the case of deterministic automaticity, the lower bound provided by Lemma 3.1 is not tight. An example of this is the following: consider the set $L = \{0^i 1^j \ : \ i \neq j\}$. Then a simple argument shows that a set of uniformly n-dissimilar strings for L can contain no more than 2 strings. Yet, we know from Lemma 2.3 that $N_L(n) \neq O(1)$.

Proposition 3.2. *Let $L = \{0^i 1^i \ : \ i \geq 0\}$. Then $N_L(n) = \Omega(n)$.*

Proof. The set $\{\epsilon, 0, 00, \ldots, 0^{\lfloor n/2 \rfloor}\}$ forms a set of uniformly n-dissimilar strings for L; the witness for 0^i is 1^i. It follows that $N_L(n) \geq \lfloor n/2 \rfloor + 1$. □

Proposition 3.3. *Let $L = \{0^i 1^i 2^i \ : \ i \geq 0\}$. Then $N_L(n) = \Omega(n^2)$.*

Proof. The set $S = \{0^i 1^j \ : \ 0 \leq j \leq i \text{ and } 0 \leq i + j \leq n/3\}$ forms a set of uniformly n-dissimilar strings for L. The witness for $0^i 1^j$ is $1^{i-j} 2^i$. The cardinality of S is easily computed to be $(\lfloor n/6 \rfloor + 1)(\lceil n/6 \rceil + 1) = \Omega(n^2)$. □

Using Lemma 3.1, we can also prove a theorem analogous to Theorem 2.4:

Proposition 3.4. *For all integers $k \geq 0$, there is a language L_k such that $N_{L_k}(n) = \Theta(n^k)$.*

Proof. Consider the languages L_k introduced in the proof of Theorem 2.4. The set S there is actually a uniformly n-dissimilar string set for L_k, and so exactly the same upper and lower bounds follow. □

Here is another application of Lemma 3.1:

Proposition 3.5. *Let* $L = \{ww \ : \ w \in (0+1)^*\}$. *Then* $N_L(n) = \Omega(2^{n/2})$.

Proof. The set $S = S_n = (0+1)^{\lfloor n/2 \rfloor}$ forms a uniformly n-dissimilar string set for L; the witness for w is w itself. It follows that $N_L(n) \geq 2^{\lfloor n/2 \rfloor}$. □

Next, we prove a simple result on some operation under which the class NPA is closed:

Proposition 3.6. *The class* NPA *is closed under the operations of union, intersection, concatenation, Kleene closure, and inverse homomorphism.*

Proof. Let M_1 be an NFA accepting an nth order approximation to L_1, and let M_2 be an NFA accepting an nth order approximation to L_2. Then we can make an NFA accepting an nth order approximation to $L_1 \cup L_2$ by using the usual construction, as given, for example, in Hopcroft & Ullman (1979), p. 31. The construction gives an automaton with $|M_1| + |M_2| + 2$ states. The other properties can be proved similarly. □

Proposition 3.7. *The class* NPA *is not closed under complement.*

Proof. See Shallit & Breitbart (1996), §5, Example 4. □

Proposition 3.8. *The class* NPA *is not closed under quotient by regular sets.*

Proof. Consider the language $L = \{ww\,1\,0^{2^k} \ : \ w \in (0+1)^k, \ k \geq 0\}$ introduced in the proof of Theorem 2.10. By the same argument given there, $L \in$ NPA. Let $R = 1\ 0^*$. Then $L/R = \{ww \ : \ w \in (0+1)^*\}$. But by Proposition 3.5, we have $N_{L/R} = \Omega(2^{n/2})$. □

4. Results on NPLA.

Finally, we examine the languages of nondeterministic poly-log automaticity: the class NPLA.

Theorem 4.1. *The class* NPLA *is closed under the operations of union, intersection, concatenation, Kleene closure, and inverse homomorphism.*

Proof. Left to the reader. □

Theorem 4.2. *The class* NPLA *is not closed under complement.*

PROOF. Let $L = \{w \in (0+1)^* \; : \; |w|_0 \neq |w|_1\}$. By Theorem 17 of Shallit & Breitbart (1996), we know that $L \in$ NPLA. (If $|w|_0 \neq |w|_1$, then there is a "small" prime p for which $|w|_0 \not\equiv |w|_1 \pmod{p}$; an NFA can "guess" the correct prime and then verify the inequality (mod p).)

Now $\overline{L} = \{w \in (0+1)^* \; : \; |w|_0 = |w|_1\}$. If this language were in NPLA, then so would $\overline{L} \cap 0^*1^*$, by Theorem 4.1. But $\overline{L} \cap 0^*1^* = \{0^i 1^i \; : \; i \geq 0\}$, which by Proposition 3.2 is not in NPLA. Hence $\overline{L} \notin$ NPLA. □

5. Relationships between complexity classes.

In this section, we examine the relationship between the classes DPA, NPA, and NPLA introduced in this paper, and the more familiar language classes REG (the regular sets) and CFL (the context-free languages).

Clearly we have the trivial inclusions REG $\subseteq$ CFL, REG $\subseteq$ DPA $\subseteq$ NPA, and REG $\subseteq$ NPLA $\subseteq$ NPA. The examples we give show that there are no other inclusion relationships between these classes, with the possible exception of the following interesting

Open Question. Is NPLA $\subseteq$ DPA?

We do not know any example of a language in NPLA $-$ DPA; nor do we have a proof that no such language exists.

In what follows, we give examples corresponding to the eight non-trivial possibilities that remain.

Example 1. A non-CFL outside NPA: $L_1 = \{ww \; : \; w \in (0+1)^*\}$. Then we know from Proposition 3.5 that $L_1 \notin$ NPA. On the other hand, a standard argument using the pumping lemma shows that L_1 is not a CFL.

Example 2. A non-CFL in NPA, but not in DPA or NPLA:

$$L_2 = \{(0+1)^* 1 (0+1)^{2^k} \; : \; k \geq 0\}.$$

From Theorem 2.8, we know that $A_{L_2}(n) \geq 2^{n/3}$ for infinitely many n. It follows that L_2 cannot be in NPLA, for if it were, we would have $A_{L_2}(n) \leq 2^{(\log n)^c}$ for some constant c, a contradiction. On the other hand, it is easy to construct a nondeterministic automaton with $O(n)$ states that accepts an nth order approximation to L_2. It is also easy to see that L_2 is not a CFL.

Example 3. A non-CFL in DPA, but not in NPLA: $L_3 = \{0^{2^i} 1^{2^i} \; : \; i \geq 0\}$. It is easy to see that L_3 is not a CFL. On the other hand, L_3 is in DPA by

Lemma 2.2. To see that L_3 is not in NPLA, we claim that $\{0, 00, \ldots, 0^{\lfloor n/4 \rfloor}\}$ is a uniformly n-dissimilar string set for L_3. The "witness" for 0^i is $0^{2^k-i}1^{2^k}$, where $2^{k-1} < i \leq 2^k$. The total length of the strings in question is therefore bounded by $\frac{n}{4} + \frac{n}{2} + \frac{n}{4} \leq n$. Hence $N_{L_3} \geq \lfloor n/4 \rfloor$ by Theorem 3.1.

Example 4. A non-CFL in DPA and NPLA: $L_4 = \{0^n \ : \ n \geq 1$ and the least positive integer not dividing n is not a power of $2\}$. Since L_4 is a unary language, L_4 is trivially in DPA. By Theorem 25 of Pomerance *et al.* (1996) we have $N_{L_4}(n) = O((\log n)^3/(\log\log n))$. Also, L_4 is not a CFL; if it were, since it is unary, it would be regular, and it is proved in Pomerance *et al.* (1996) that L_4 is not regular.

Example 5. A CFL outside NPA: $L_5 = \{w \in (0+1)^* \ : \ w = w^R\}$. Clearly L_5 is a CFL. To see that L_5 is not in NPA, it suffices to observe that $(0+1)^{\lfloor n/2 \rfloor}$ is a uniformly n-dissimilar string set for L_5.

Example 6. A CFL in NPA, but not in DPA or NPLA: $L_6 = \{w \in (0+1)^* \ : \ w \neq w^R\}$. In §5, Example 4 of Shallit & Breitbart (1996), it is proved that $N_{L_6}(n) = \Theta(n)$ and $A_{L_6}(n) = \Omega(2^{n/2})$.

Example 7. A CFL in DPA, but not in NPLA: $L_7 = \{0^n 1^n \ : \ n \geq 0\}$. In §5, Example 1 of Shallit & Breitbart (1996), it is proved that $A_{L_7}(n) = \Theta(n)$. In Proposition 3.2 above, we showed that $N_{L_7} = \Omega(n)$.

Example 8. A non-regular CFL in DPA and NPLA: $L_8 = \{w \in (0+1)^* \ : \ |w|_0 \neq |w|_1\}$. In Theorem 17 of Shallit & Breitbart (1996) it is shown that $N_{L_8}(n) = O((\log n)^2/(\log\log n))$. On the other hand, it is easy to show that $A_{L_8}(n) = n + 1$.

6. Automaticity and context-free languages

In this section we briefly discuss the automaticity of context-free languages. As shown in Theorem 5 of Shallit & Breitbart (1996), there exists a CFL L_s such that $A_{L_s}(n) = \lfloor (n+3)/2 \rfloor$. By a theorem of Karp (1967), we know that if L is nonregular, then $A_L(n) \geq (n+3)/2$ for infinitely many n, so L_s is the language of essentially the lowest-possible deterministic automaticity. On the other hand, in §5, Example 4 of Shallit & Breitbart (1996), it is shown that the CFL $L_4 = \{w \in (0+1)^* \ : \ w \neq w^R\}$ has automaticity $A_{L_4}(n) = \Omega(2^{n/2})$. This raises the question, what is the maximum possible deterministic automaticity for a context-free language over $\{0, 1\}$?

We know from Theorem 9 of Shallit & Breitbart (1996) that if $L \subseteq (0+1)^*$, then $A_L(n) = O(2^n/n)$. We have not been able to find a CFL with deterministic automaticity $\Omega(2^n/n)$, but in the following theorem we construct a sequence of languages with deterministic automaticity arbitrarily close to 2^n:

THEOREM 6.1. *For all real $\epsilon > 0$, there exists a CFL of deterministic automaticity $\Omega(2^{n(1-\epsilon)})$.*

PROOF. First, we introduce the following notation. If w is a string with $|w| = n$, then by w_{-i}, $(1 \leq i \leq n)$ we mean the symbol w_{n-i+1}.

To prove the result, we will show that for all integers $r \geq 1$, there exists a language L_r of automaticity $\Omega(2^{\lfloor rn/(r+1) \rfloor - r})$. Let

$$L_r = \{w\ 0\ 1^a\ 0^b\ :\ w \in (0+1)^*,\ 1 \leq a \leq r,\ b \geq 0,\ w_{-(rb+a)} = 1\}.$$

First, we claim that L_r is a CFL for all $r \geq 1$. To see this, consider a PDA M that reads the symbols of w and puts them on its stack. When it sees a 0 in the input, it (nondeterministically) can choose to enter a state in which it pops symbols off the stack (1 symbol, if it sees a 1 in the input; r symbols if it sees a 0 in the input). If the last symbol popped off the stack is a 1, M may (nondeterministically) choose to accept. We leave it to the reader to verify that M accepts L_r.

We now prove that $A_{L_r}(n) \geq 2^{\lfloor rn/(r+1) \rfloor - r}$. To do this, we exhibit an n-dissimilar string set $S = S_{n,r}$ of cardinality $2^{\lfloor rn/(r+1) \rfloor - r}$.

Assume $n \geq r+5$, so that $\lfloor rn/(r+1) \rfloor - r \geq 1$, and define

$$S = S_{n,r} = \{w\ :\ |w| = \lfloor rn/(r+1) \rfloor - r\}.$$

Pick two distinct strings from S, say x and y. There must be some position k at which x and y differ, say $x_{-k} \neq y_{-k}$. Clearly $1 \leq k \leq \lfloor rn/(r+1) \rfloor - r$ by construction. Without loss of generality, assume $x_{-k} = 1$ and $y_{-k} = 0$. Write $k = rb + a$ with $1 \leq a \leq r$ and $0 \leq b \leq n/(r+1) - 1$.

Consider $z = 0\ 1^a\ 0^b$. Then clearly $xz \in L$, but $yz \notin L$. Also,

$$\begin{aligned} |xz| = |yz| &= \left\lfloor \frac{rn}{r+1} \right\rfloor - r + a + b + 1 \\ &\leq \left\lfloor \frac{rn}{r+1} \right\rfloor - r + r + \frac{n}{r+1} - 1 + 1 \\ &= \left\lfloor \frac{rn}{r+1} \right\rfloor + \frac{n}{r+1} \leq n, \end{aligned}$$

so x and y are n-dissimilar.

To complete the proof, take $r = \lceil 2/\epsilon \rceil$. Then for $n \geq (r+1)^2$, we have $A_{L_r}(n) \geq 2^{n(1-\epsilon)}$. □

7. Acknowledgments.

This paper is based in part on the M. Math. thesis of the first author (Glaister (1995)). A preliminary version was presented at the 21st International Symposium on Mathematical Foundations of Computer Science (1996) in Kraków, Poland as Glaister & Shallit (1996). Jean-Paul Allouche and Jonathan Buss read a draft of this paper and offered several corrections. We would like to thank the referees for an extremely careful reading of this paper.

References

A. V. Aho, J. D. Ullman, and M. Yannakakis, On notions of information transfer in VLSI circuits. In *Proc. Fifteenth Ann. ACM Symp. Theor. Comput.* ACM, 1983, 133–139.

A. Condon, L. Hellerstein, S. Pottle, and A. Wigderson, On the power of finite automata with both nondeterministic and probabilistic states. In *Proc. Twenty-sixth Ann. ACM Symp. Theor. Comput.* ACM, 1994, 676–685.

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, 1989, 480–485.

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

I. Glaister, Automaticity and closure properties. Master's thesis, University of Waterloo, 1995.

I. Glaister and J. Shallit, Polynomial automaticity, context-free languages, and fixed points of morphisms. In *Proc. 21st Annual Symposium, Mathematical Foundations of Computer Science, MFCS '96*, ed. W. Penczek and A. Szałas, vol. 1113 of *Lecture Notes in Computer Science*, 382–393. Springer-Verlag, 1996.

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

J. Kaneps and R. Freivalds, Minimal nontrivial space complexity of probabilistic one-way Turing machines. In *MFCS '90 (Mathematical Foundations of Computer Science)*, ed. B. Rovan, vol. 452 of *Lecture Notes in Computer Science*, 355–361. Springer-Verlag, 1990.

J. Kaneps and R. Freivalds, Running time to recognize nonregular languages by 2-way probabilistic automata. In *ICALP '91 (18th International Colloquium on Automata, Languages, and Programming)*, ed. J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, vol. 510 of *Lecture Notes in Computer Science*, 174–185. Springer-Verlag, 1991.

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

C. Pomerance, J. M. Robson, and J. O. Shallit, Automaticity II: Descriptional complexity in the unary case. To appear, *Theoret. Comput. Sci.*, 1996.

J. Shallit, Automaticity IV: Sequences, sets, and diversity. To appear, *J. Théorie des Nombres de Bordeaux*, 1996.

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

J. Shallit and Y. Breitbart, Automaticity I: Properties of a measure of descriptional complexity. To appear, *J. Comput. System Sci.*, 1996.

A. C. Yao, Some complexity questions reltated to distributive computing. In *Proc. Eleventh Ann. ACM Symp. Theor. Comput.* ACM, 1979, 209–213.

S. Yu, Q. Zhuang, and K. Salomaa, The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.* **125** (1994), 315–328.

Manuscript received December 14, 1995

Ian Glaister
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

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

Current address of Ian Glaister:
Array Systems Computing, Inc.
1120 Finch Avenue West
8th Floor
North York, ON M3J 3H7
Canada
ian@array.ca