Journal of Automata, Languages and Combinatorics **7** (2002) 4, –
© Otto-von-Guericke-Universität Magdeburg

STATE COMPLEXITY OF PROPORTIONAL REMOVALS [1]

Michael Domaratzki
School of Computing, Queen's University
Kingston, ON K7L 3N6 Canada
e-mail: `domaratz@cs.queensu.ca`

ABSTRACT

We examine the state complexity of proportional removals such as $\frac{1}{2}(L)$. For $\frac{1}{2}(L)$, we show a bound which is tight in the case that L is a unary language, and an nearly optimal bound for arbitrary languages. We also compute the average state complexity for $\frac{1}{2}(L)$ if L is unary. We study other proportional removals and give bounds for certain reset automata.

Keywords: State complexity, reset automata, proportional removals

1. Introduction and Motivation

The *state complexity* of a regular language L is the number of states in the minimal deterministic finite automaton (DFA) recognizing L. There has been much interest lately in the study of state complexity of operations which preserve regularity (e. g. [2, 9]). These papers are generally interested in proving upper bounds on the state complexity of operations on regular languages, including the particular case of unary regular languages. In this paper, we examine proportional removals, that is, languages of the form

$$\{x \in \Sigma^* \mid \exists y \in \Sigma^* \text{ such that } r(|x|, |y|) \wedge xy \in L\} \tag{1}$$

for some language $L \subseteq \Sigma^*$, and a binary relation r. There is a complete characterization, due to Seiferas and McNaughton [6], of the relations r which ensure that (1) is a regular language, if L is regular. We obtain bounds for the equality relation, for both unary and general languages, and show this bound is tight for unary languages.

2. Notation and Definitions

We assume the reader is familiar with basic concepts in automata theory and formal languages. For any unfamiliar terminology, see Hopcroft and Ullman [4] or S. Yu [8].

[1]Full version of a submission presented at the Third International Workshop on *Descriptional Complexity of Automata, Grammars and Related Structures* (Vienna, Austria, July 20 – 22, 2001).

A deterministic finite automaton (DFA) is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is the set of states, Σ is the alphabet, and δ is the transition function $\delta : Q \times \Sigma \to Q$. The start state is $q_0 \in Q$ and the final states are $F \subseteq Q$. We extend δ to a function from $Q \times \Sigma^*$ to Q in the obvious way. A DFA is said to be *complete* if $\delta(q, a)$ is defined for all $q \in Q$, $a \in \Sigma$. In what follows, we will assume that all DFAs are complete. A DFA is said to be *initially connected* if, for all $q \in Q$, there exists a word $w \in \Sigma^*$ such that $\delta(q_0, w) = q$, that is, every state q is reachable from the initial state by reading some word w.

A string $w \in \Sigma^*$ is accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ if $\delta(q_0, w) \in F$. Define the language accepted by a DFA M by $L(M) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. A DFA $M = (Q, \Sigma, \delta, q_0, F)$ is minimal for a regular language L if $L(M) = L$ and for all DFAs $M' = (Q', \Sigma, \delta', q_0', F')$ with $L(M') = L$, we have $|Q| \leq |Q'|$. The *state complexity* of a regular language L is denoted by $\mathrm{sc}(L)$ and is defined as the number of states in the minimal DFA for L.

If Σ is an alphabet, we use the notation $\Sigma^k = \{x \in \Sigma^* \mid |x| = k\}$.

We adopt the notation of Seiferas and McNaughton [6]: For any binary relation $r \subseteq \mathbb{N} \times \mathbb{N}$ and any language $L \subseteq \Sigma^*$, let the language $P(r, L)$ be defined as

$$P(r, L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \text{ such that } r(|x|, |y|) \wedge xy \in L\}.$$

For any set $A \subseteq \mathbb{N}$, we say that A is *ultimately periodic* (u.p.) if there exist integers $n_0 \geq 0$ and $p > 0$ such that for all $n > n_0$, $n \in A \iff n + p \in A$. If n_0 and p are chosen to be minimal among all integers satisfying those respective conditions, we say that n_0 is the *preperiod* of A, and p is the *period* of A. For any relation $r \subseteq \mathbb{N} \times \mathbb{N}$ and any set A, define

$$r^{-1}(A) = \{i \mid \exists j \in A \text{ such that } r(i, j)\}.$$

We call r *u.p.-preserving* if A u.p. implies $r^{-1}(A)$ u.p. The following is the complete characterization of relations r which preserve regularity, due to Seiferas and McNaughton [6].

Theorem 1 *For all languages L, the following are equivalent*

1. *L regular $\Rightarrow$ $P(r, L)$ is regular, and*
2. *r is u.p.-preserving.* $\square$

We say that a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is unary if $|\Sigma| = 1$. We observe that for any unary DFA, there is exactly one transition going out of any state. Thus, the transition diagram of the DFA consists of a chain of states (the 'tail') and a simple loop of states (the 'loop').

Let $H(n)$ be defined by $H(n) = e^{\sqrt{n \log n}}$. This function is useful in the following result of Chrobak [3].

Theorem 2 *For any n state unary NFA M there is a unary DFA M' with a 'tail' of $O(n^2)$ states and a 'loop' of size at most $O(H(n))$ such that $L(M) = L(M')$.* $\square$

3. State Complexity of $\frac{1}{2}(L)$

We begin with a particular removal, namely that given by the binary relation $r = \{(n, n) \mid n \geq 0\}$. We denote this special case by $\frac{1}{2}(L)$:

$$\frac{1}{2}(L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \text{ with } |x| = |y| \text{ and } xy \in L\}.$$

Our main theorem is a general bound on the state complexity of $\frac{1}{2}(L)$.

Theorem 3 *Let L be a regular language with $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(\frac{1}{2}(L)) = O(nH(n))$.*

Proof. Let L be recognized by the DFA M with $M = (Q, \Sigma, \delta, q_0, F)$ and $Q = \{q_0, q_1, \ldots, q_{n-1}\}$. Define the NFA

$$M' = (Q \cup \{q_0'\}, \{a\}, \delta', q_0', F')$$

with

$$\delta'(q_0', \epsilon) = \{q \mid q \in F\}$$

and

$$\delta'(q, a) = \{q' \mid \exists b \in \Sigma \text{ such that } \delta(q', b) = q\}$$

for each $q \in Q$. For each $i \geq 0$, we define sets $Q_i \subseteq Q$ as follows: $Q_0 = F$ and for $i > 0$, $Q_i = \delta'(q_0', a^i)$. Note that $\delta'(Q_i, a) = Q_{i+1}$ and

$$Q_j = \{q \mid \exists w \in \Sigma^* \text{ such that } |w| = j \text{ and } \delta(q, w) \in F\}. \qquad (2)$$

We may now define F' as

$$F' = Q_{2^n}.$$

We choose Q_{2^n} since there are 2^n subsets of Q. Thus, if we consider the sequence $\{Q_i\}_{i \geq 0}$, there are indices $0 \leq j < k \leq 2^n - 1$ such that $Q_j = Q_k$. Thus all possible distinct sets Q_i will be encountered by the time we consider F'. Thus, in determinizing M', all distinct subsets of Q which are states in M' will appear as a state in the deterministic equivalent.

By Theorem 2, there is a unary DFA $M'' = (Q'', \{a\}, \delta'', q_0'', F'')$ such that $L(M'') = L(M')$ and M'' has at most $O(H(n))$ states. Further, for all $i \geq 0$ we can associate Q_i with the state $q \in Q''$ such that $\delta''(q_0'', a^i) = q$. Thus, we number the states in Q'' so that $\delta''(q_0'', a^i) = q_i$ is associated with Q_i. Let M'' have r states in its tail and s states in its loop.

We now define a DFA M_h, which we claim will recognize $\frac{1}{2}(L)$. Let $M_h = (Q_h, \Sigma, \delta_h, q_{0,h}, F_h)$, with

$$Q_h = Q \times \{i \mid 0 \leq i < r + s\},$$

$$\delta_h((q, i), b) = \begin{cases} (\delta(q, b), i + 1), & \text{if } i < r + s - 1; \\ (\delta(q, b), r), & \text{if } i = r + s - 1; \end{cases}$$

and

$$q_{0,h} = (q_0, 0).$$

Thus we compute the action of M in the first component, and implement a periodic counter in the second component, with preperiod r and period s. Finally,

$$F_h = \{(q_i, j) \mid q_i \in Q_j\}.$$

We show that $L(M_h) = \frac{1}{2}(L)$. Let $x \in L(M_h)$, so that $\delta_h(q_{0,h}, x) = (q_i, j) \in F'$. Then $q_i \in Q_j = Q_{|x|}$. Thus, by (2), there exists $z \in \Sigma^*$ with $|z| = |x|$ and $\delta(q_i, z) \in F$. Thus, since $\delta(q_0, x) = q_i$, we may conclude that $x \in \frac{1}{2}(L)$, since $\delta(q_0, xz) \in F$.

Conversely, let $x \in \frac{1}{2}(L)$. Then $xy \in L$ for some y with $|x| = |y|$. Let $\delta(q_0, x) = q_j$ for some j. So $\delta(q_j, y) \in F$. Thus $q_j \in Q_{|y|} = Q_{|x|}$. Let i be the integer with $i \equiv |x| \pmod{r+s}$ with $0 \leq i < r+s$. Then $q_j \in Q_i$ by our construction, and by definition of δ_h, $\delta_h(q_{0,h}, x) = (q_j, i)$. Further, $(q_j, i) \in F_h$ since $q_j \in Q_i$, so $x \in L(M_h)$.

We conclude that $\mathrm{sc}(\frac{1}{2}(L)) \leq |Q'| = n(r+s) = O(nH(n))$. □

Note: Theorem 3 was known to S. Yu, but was not published.

Let L be a unary language, with $\mathrm{sc}(L) = p + r$, where p is the length of the 'tail' of the unary DFA recognizing L, and r is the length of the 'loop'. Then we note that M'' in the proof of Theorem 3 actually has tail of size p and loop of size r, since we can take $M'' = M$. So, simulating the proof of Theorem 3, we get the following corollary.

Corollary 4 *Let L be a unary language with $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(\frac{1}{2}(L)) \leq n$.* □

We can define a unary language L_n for each $n \geq 1$ to show that this bound is tight.

Lemma 5 *For all $n \geq 1$ there exists a unary language L_n such that $\mathrm{sc}(L_n) = \mathrm{sc}(\frac{1}{2}(L_n)) = n$.* □

It is easy to see that for $n \geq 3$, $L = a^{n-2}(a^n)^*$ will work for n odd, and $L = (a^{n-1})^+$ will work for n even. For $n = 2$, we can take $L = a^+$ and for $n = 1$, we can take $L = a^*$. Note that for both these languages, we have $\mathrm{sc}(L) = L$.

Theorem 6 *For infinitely many k, there exists a language L_k such that $\mathrm{sc}(L_k) = O(n)$ but $\mathrm{sc}(\frac{1}{2}(L_k)) > cH(n)$ for some constant c.*

Proof. Let p_i denote the ith prime with $p_1 = 2$. We define $n_i = \sum_{j=2}^{i} p_j$ and $m_i = \prod_{j=2}^{i} p_j$. Then we will need the following known number-theoretic estimates.

Lemma 7 [1, p. 29] *For all $i \geq 1$, $n_i \in \Theta(\frac{(i \log i)^2}{2 \log i})$.* □

Lemma 8 [1, Cor. 8.2.7] *For all $i \geq 1$, $\log(m_i) = i \log i(1 + o(1))$.* □

Thus, we may conclude that $m_i = e^{\sqrt{n_i \log n_i}(1+o(1))}$ for all i. Let $k \geq 3$. We write p for p_k in what follows. Define the language L_k as follows.

$$L_k = \bigcup_{i=2}^{k-1} (0^p)^* 0^{p-p_i} 1 0^{p_i-1} (0^{p_i})^*.$$

We show that for all $k \geq 3$, $\mathrm{sc}(L_k) = n_k + 1$ while $\mathrm{sc}(\frac{1}{2}(L_k)) \geq m_k$. This will show the result.

To show that $\mathrm{sc}(L_k) = n_k + 1$ we will construct a DFA M_k with $n_k + 1$ states such that $L(M_k) = L_k$. This will establish $\mathrm{sc}(L_k) \leq n_k + 1$, which will actually suffice for our argument (though it not hard to establish that M_k is minimal and thus $\mathrm{sc}(L_k) = n_k + 1$). For each prime p_i with $i \geq 2$, let C_i be a cycle of p_i states, $\{q_{i,0}, q_{i,1}, \ldots, q_{i,p_i-1}\}$, with transitions on 0 as follows: $\delta(q_{i,j}, 0) = q_{i,j+1}$ where $j + 1$ is taken modulo p_i. For each k define $Q_k = \bigcup_{i=2}^{k} C_i$. Finally, let $M_k = (Q_k \cup \{q_d\}, \{0, 1\}, \delta, q_{k,0}, F)$ where δ includes all the cycle transitions on 0 defined above, as well as the transitions $\delta(q_{k,p-p_i}, 1) = q_{i,0}$ for all $2 \leq i < k$. The remaining undefined transitions go to q_d. Finally,

$$F = \{q_{i,p_i-1} \mid 2 \leq i < k\}.$$

We can verify that $L(M_k) = L_k$. Figure 1 shows a DFA M_4 recognizing L_4.

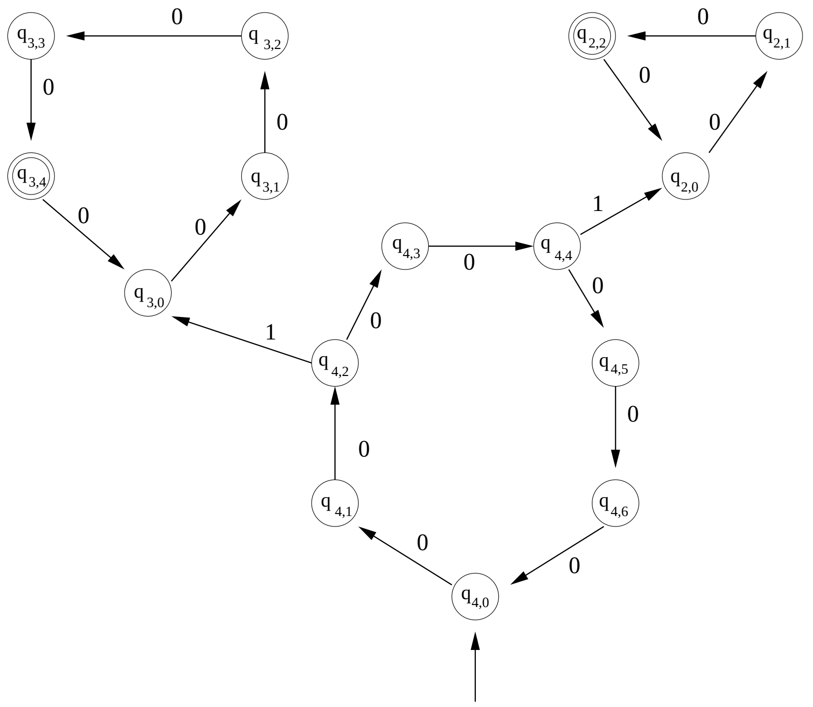

Figure 1: A DFA M_4 recognizing L_4. Unspecified transitions go to a dead state, not shown.

To show that $\mathrm{sc}(\frac{1}{2}(L_k)) \geq m_k$, we use the Myhill-Nerode Theorem. Consider the strings $w_i = 0^i$ for $0 \leq i < m_k$.

The following lemma will be useful.

Lemma 9 *For each prime p_i with $2 \leq i < k$, the following hold:*

(a) *For each integer $r \geq 0$ there is exactly one s with $0 \leq s < p_i$ such that*

$$0^{pr+p-p_i}10^s \in \frac{1}{2}(L_k).$$

(b) *For any integer s with $0 \leq s < p_i$, the condition*

$$0^{pr_1+p-p_i}10^s \in \frac{1}{2}(L_k) \iff 0^{pr_2+p-p_i}10^s \in \frac{1}{2}(L_k)$$

holds if and only if $r_1 \equiv r_2 \pmod{p_i}$.

Proof. Fix p_i. Let r be any integer. Then $0^{pr+p-p_i}10^s \in \frac{1}{2}(L_k)$ iff there is some integer $t \geq 0$ such that $0^{pr+p-p_i}10^s0^{p_i-1-s}0^{p_it} \in L_k$ and $pr+p-p_i+1+s = p_i-1-s+p_it$, or, equivalently iff there is some $t \geq 0$ such that

$$p_i(t+2) = pr+p+2+2s.$$

This implies

$$p+2+2s+rp \equiv 0 \pmod{p_i}.$$

Now, we prove the converse also holds. Assume that

$$p+2+2s+rp \equiv 0 \pmod{p_i}. \tag{3}$$

Note that $p > 0$ and $s \geq 0$. Thus $p+2+2s+rp \geq 0$. Then (3) implies that there exists some $t' \geq 0$ such that

$$p+2+2s+rp = t'p_i.$$

But now

$$\begin{aligned} t'p_i &= p(r+1)+2+2s \\ &\geq p(r+1) \\ &> p_i(r+1). \end{aligned}$$

Since $r \geq 0$, $p_i(r+1) \geq p_i$. Thus $t' > 1$. Let $t = t'-2 \geq 0$. Then we have

$$p+2+2s+rp = (t+2)p_i.$$

Thus we have established the following:

$$p+2+2s+rp \equiv 0 \pmod{p_i} \iff 0^{pr+p-p_i}10^s \in \frac{1}{2}(L_k) \tag{4}$$

This establishes part (a) of the lemma: Choose s such that

$$p+2+2s+rp \equiv 0 \pmod{p_i}$$

and $0 \leq s < p_i$. This s exists since the integers modulo p_i form a field, and $p_i \neq 2$.

Further, we can establish the statement (b) of the lemma using (4). Fix an integer s with $0 \leq s < p_i$. Then

$$\begin{aligned} & r_1 \equiv r_2 \pmod{p_i} \\ & \quad \iff (p + 2 + 2s \equiv -r_1 p \pmod{p_i} \iff p + 2 + 2s \equiv -r_2 p \pmod{p_i}) \\ & \quad \iff \Big(0^{pr_1+p-p_i}10^s \in \frac{1}{2}(L_k) \iff 0^{pr_2+p-p_i}10^s \in \frac{1}{2}(L_k)\Big). \qquad \square \end{aligned}$$

We may now prove that each of the strings 0^i for $0 \leq i < m_k$ are in different equivalence classes of the Myhill-Nerode relation. Consider two strings $w_i = 0^i$, $w_j = 0^j$ for some $0 \leq i < j < m_k$. Write

$$\begin{aligned} i &= pi' + i'' \\ j &= pj' + j'' \end{aligned}$$

where $0 \leq i', j' < m_{k-1} = m_k/p$ and $0 \leq i'', j'' < p$. We have two cases:

Case (a): $i'' \neq j''$. We will construct a string $z_{i,j} \in 0^*10^*$ such that exactly one of $w_i z_{i,j}$ and $w_j z_{i,j}$ is a member of $\frac{1}{2}(L_k)$. We have three subcases.

Case (a-i): *There exist primes p_α and p_β $(0 \leq \alpha, \beta < k)$ such that $i'' = p - p_\alpha$ and $j'' = p - p_\beta$.* Since $i'' \neq j''$, $\alpha \neq \beta$. Assume without loss of generality that $p_\alpha < p_\beta$.

According to Lemma 9, let s_1 be the unique integer (depending on i') with $0 \leq s_1 < p_\alpha$ and

$$w_i 10^{s_1} = 0^i 10^{s_1} = 0^{pi'+p-p_\alpha}10^{s_1} \in \frac{1}{2}(L_k).$$

Similarly, let s_2 be the unique integer depending on j' with $0 \leq s_2 < p_\beta$ and

$$w_j 10^{s_2} = 0^j 10^{s_2} = 0^{pj'+p-p_\beta}10^{s_2} \in \frac{1}{2}(L_k).$$

If $s_1 \neq s_2$ we are done, since then we can choose $z_{i,j} = 10^{s_1}$. Since $p_\alpha < p_\beta$ and thus $s_1 < p_\alpha < p_\beta$, s_1 is not equivalent to s_2 modulo p_β. Thus $w_j z_{i,j} = w_j 10^{s_1} \notin \frac{1}{2}(L_k)$.

If $s_1 = s_2$, then we choose $z_{i,j} = 10^{s_1+p_\alpha}$. It is simple to show that

$$w_i 10^{s_1+p_\alpha} = 0^{pi'+p-p_\alpha}10^{s_1+p_\alpha} \in \frac{1}{2}(L_k).$$

while since $p_\alpha < p_\beta$,

$$w_j 10^{s_1+p_\alpha} = 0^{pj'+p-p_\beta}10^{s_1+p_\alpha} \notin \frac{1}{2}(L_k).$$

Case (a-ii): *There exists a prime p_α such that $i'' = p - p_\alpha$ but for all primes p_ℓ with $2 \leq \ell < k$, $j'' \neq p - p_\ell$.* The same case with the roles of i'' and j'' reversed holds similarly to what follows.

In this case, we choose the unique s_i with $0 \leq s_i < p_\alpha$ such that $w_i 10^{s_i} = 0^{pi'+p-p_\alpha}10^{s_i} \in \frac{1}{2}(L_k)$. Since $j'' \neq p - p_\ell$ for all primes p_ℓ in the range $2 \leq \ell < k$, $w_j 10^s \notin \frac{1}{2}(L_k)$ for any s, since we cannot complete $w_j 1$ with any string of zeroes to ensure it is in L_k. This completes our case, as we can separate w_i and w_j with $z_{i,j} = 10^{s_i}$.

Case (a-iii): *For both i'' and j'' there is no prime p_ℓ with $2 \le \ell < k$ such that $i'' = p - p_\ell$ or $j'' = p - p_\ell$.* We can reduce this to case (a-i) or (a-ii): Let r be the smallest integer such that $i'' + r \equiv -p_\alpha \pmod{p}$ for some prime $p_\alpha < p$; certainly r must exist. Then $w_i 0^r$ and $w_j 0^r$ must reduce to one of the above cases, since if we let u be the integer with $0 \le u < p$ such that $u \equiv i'' + r \pmod{p}$, then p_α satisfies $u = p - p_\alpha$. Thus, let $z'_{i,j}$ be the string such that $w_i 0^r z'_{i,j} \in \frac{1}{2}(L_k)$ and $w_j 0^r z'_{i,j} \notin \frac{1}{2}(L_k)$. Then clearly we may choose $z_{i,j} = 0^r z'_{i,j}$.

Case (b): $i'' = j''$. Then we must have that $i' \ne j'$. Since $0 \le i', j' < m_{k-1}$, we must have that there exists an ℓ with $1 \le \ell < k$ such that $i' \not\equiv j' \pmod{p_\ell}$, since if this were not the case then $i' \equiv j' \pmod{p_\ell}$ for all $1 \le \ell < k$ and so $i' \equiv j' \pmod{m_{k-1}}$, which would imply $i' = j'$, as we have noted that $0 \le i', j' < m_{k-1}$. Thus, let ℓ be chosen so that $i' \not\equiv j' \pmod{p_\ell}$.

Let j_i be chosen such that $0 < j_i < p_\ell$ and

$$0^{pi'+p-p_\ell} 1 0^{j_i} \in \frac{1}{2}(L_k).$$

This is possible by Lemma 9. Also by Lemma 9, we know that since $i' \not\equiv j' \pmod{p_\ell}$,

$$0^{pj'+p-p_\ell} 1 0^{j_i} \notin \frac{1}{2}(L_k).$$

Thus, if $i''(= j'') \le p - p_\ell$, we may choose $z_{i,j} = 0^{(p-p_\ell)-i''} 1 0^{j_i}$, as $w_i z_{i,j} \in \frac{1}{2}(L_k)$, while $w_j z_{i,j} \notin \frac{1}{2}(L_k)$. If $p > i''(= j'') > p - p_\ell$, then note that $i'+1 \not\equiv j'+1 \pmod{p_\ell}$. Thus, in a similar argument to that presented above, we may choose j'_i such that

$$0^{p(i'+1)+p-p_\ell} 1 0^{j'_i} \in \frac{1}{2}(L_k).$$

and

$$0^{p(j'+1)+p-p_\ell} 1 0^{j'_i} \notin \frac{1}{2}(L_k).$$

Then we choose $z_{i,j} = 0^{p-i''} 0^{p-p_\ell} 1 0^{j'_i}$. Thus $w_i z_{i,j} = 0^{p(i'+1)+p-p_\ell} 1 0^{j'_i} \in \frac{1}{2}(L_k)$ and $w_j z_{i,j} = 0^{p(j'+1)+p-p_\ell} 1 0^{j'_i} \notin \frac{1}{2}(L_k)$. This completes the proof. □

4. State Complexity of Polynomial Removals

Let $f \in \mathbb{Z}[x]$ be a polynomial with integral coefficients. If $f(n) \in \mathbb{N}$ for all $n \in \mathbb{N}$, we denote this by $f(\mathbb{N}) \subset \mathbb{N}$. Recall that a function f is strictly monotonic if $i > j$ implies $f(i) > f(j)$. Our main theorem is:

Theorem 10 *Let $f \in \mathbb{Z}[x]$ be a strictly monotonic polynomial such that $f(\mathbb{N}) \subset \mathbb{N}$. Then the relation $r_f = \{(n, f(n)) \mid n \ge 0\}$ preserves regularity, and*

$$\mathrm{sc}(P(r_f, L)) \le O(\mathrm{sc}(L) H(\mathrm{sc}(L))).$$

The following is an easy fact about polynomials with integer coefficients:

Fact 11 *Let $f \in \mathbb{Z}[x]$ with $f(\mathbb{N}) \subseteq f(\mathbb{N})$. Then for all $n_1, n_2 \in \mathbb{N}$ with $n_2 > 0$, $f(n_1) \equiv f(n_1 + n_2) \pmod{n_2}$.* □

Note that the previous fact is not true if we replace f in the statement by $f \in \mathbb{Q}[x]$ and f integer-valued (i. e., $f(k) \in \mathbb{Z}$ for all $k \in \mathbb{Z}$). For example, let $f(n) = \frac{n(n-1)}{2}$. Then f is integer valued, and $f(4+2) = 15 \equiv 3 \pmod 4$ but $f(2) = 1 \equiv 1 \pmod 4$.

We have the following definitions, which will allow us to characterize polynomials which preserve regularity. They are from Seiferas and McNaughton [6] and Siefkes [7]: A function $f : \mathbb{N} \to \mathbb{N}$ is u.p.-reducible if, for every modulus m, there is a period r_m such that the following congruence holds for all but finitely many $n \in \mathbb{N}$: $f(n) \equiv f(n + r_m) \pmod m$. A function $f : \mathbb{N} \to \mathbb{N}$ is essentially increasing if, for every k, $f(n) \geq k$ for all but finitely many $n \in \mathbb{N}$.

Theorem 12 [6, Thm. 3, p. 151] *If f is essentially increasing and u.p.-reducible, then f is u.p.-preserving.* □

The following corollary is immediate, considering Fact 11.

Corollary 13 *If $f \in \mathbb{Z}[x]$ with $f(\mathbb{N}) \subset \mathbb{N}$ and f strictly monotonic, then f is u.p.-preserving.* □

The following fact is simply a consequence of defining the period of a set as the smallest of all possible periods.

Fact 14 *Let A be a u.p. set with period p, and preperiod n_0. Let p' be any integer satisfying*

$$a + p' \in A \iff a \in A$$

for all $a \geq n_0$. Then $p | p'$. □

Fact 15 *Let $f \in \mathbb{Z}[x]$ be a strictly monotonic polynomial with $f(\mathbb{N}) \subset \mathbb{N}$. Then $f(n) \geq n$ for all $n \in \mathbb{N}$.* □

We are ready for the proof of Theorem 10:

Proof. Let A be the ultimately periodic set constructed in the proof of Theorem 3 by the unary portion of the DFA for $\frac{1}{2}(L)$.

By Corollary 13, f is u.p.-preserving. Thus $r_f^{-1}(A)$ is u.p. Let p_f be the period of $r_f^{-1}(A)$. We will show that if p is the period of A, then $p_f | p$, and thus $p_f \leq p$. This will give the result.

Let $i \in r_f^{-1}(A)$. Assume that $i \geq n_0$, where n_0 is the preperiod of A. By definition of $r_f^{-1}(A)$, there is a $j_i \in A$ such that $j_i = f(i) \geq i$.

By Fact 11, $f(i) \equiv f(i+p) \pmod p$. Stated another way, this tells us that

$$f(i+p) = f(i) + \ell p.$$

for some integer ℓ. Since $f(i+p) > f(i)$, we can further deduce that $\ell > 0$. Thus, $f(i+p) \in A$, because p is the period of A. We may conclude that $i + p \in r_f^{-1}(A)$ by definition, and so $i \in r_f^{-1}(A) \Rightarrow i + p \in r_f^{-1}(A)$. We can similarly prove the reverse implication. Thus, by Fact 14, $p_f | p$. Since we assume that both p and p_f are positive integers, this gives us that $p_f \leq p$.

Thus, we can construct a DFA for $P(r_f, L)$ in the same manner of Theorem 3, replacing a loop of period p by a loop of period p_f. This gives the result. $\square$

5. Reset Automata

A reset automaton is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that there exists a word $w \in \Sigma^*$ and a state $q_s \in Q$ such that for all $q \in Q$, $\delta(q, w) = q_s$. We call any such w a reset word, and q_s the reset state.

We consider the state complexity of proportional removals on reset automata with accepting reset state as an demonstration of a class of automata for which the state complexity of proportional removals is exponentially more efficient than the general case.

Recall our definition of the sets Q_i:

$$Q_i = \{q \in Q \mid \exists w \in \Sigma^i \text{ with } \delta(q, w) \in F\}. \tag{5}$$

We will relate these Q_i to reset automata. First we note that if the Q_i eventually reach the entire set Q of states, then the state complexity of $\frac{1}{2}(L)$ is low.

Lemma 16 *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. If $Q_r = Q$ for some r then $Q_{r+1} = Q$.* $\square$

Lemma 17 *If $Q_\ell = Q$ for some $\ell \geq 0$, then $\mathrm{sc}(\frac{1}{2}(L)) \leq O(\mathrm{sc}(L)^3)$.*

Proof. Let $\mathrm{sc}(L) = n$. Let M'' be the deterministic unary component of the construction given in Theorem 3. Then M'' has a tail of $r = O(n^2)$ states, and a loop of length $s = O(H(n))$. Consider $t = r + s$. By Lemma 16, since $Q_\ell = Q$ for some $\ell \geq 0$, we have $Q_t = Q$. By another application of Lemma 16, $Q_s = Q$, since $Q_{t+1} = Q_s$ by definition of the unary component given in Theorem 3, and the loop structure of the unary DFA. Then we can conclude that $Q_i = Q$ for all $s \leq i \leq t = r + s$. Thus, we may replace all s such states with a single state s_t which loops to itself on all transitions. This reduces the number of states in M'' to t and thus an upper bound on the number of states in a DFA recognizing $\frac{1}{2}(L)$ is $n \cdot O(n^2) = O(n^3)$. $\square$

The following lemma follows directly from the definition of reset automata.

Lemma 18 *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a reset automaton. If $q_s \in F$ then $Q_r = Q$ for some r.* $\square$

Thus, combining Lemmas 18 and 17, we have the following theorem.

Theorem 19 *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a reset automaton with reset state $q_s \in F$, and let $L = L(M)$. Let $n = |Q|$. Then $\mathrm{sc}(\frac{1}{2}(L)) \leq O(n^3)$.* $\square$

6. Average state complexity of $\frac{1}{2}(L)$

Recently, Nicaud [5] has examined the average state complexity of operations on regular languages. This turns out to be much harder than worst-case complexity, as it is currently unknown how many non-isomorphic automata there are over a two-letter alphabet. However, for unary automata, this is known. In this section, we follow the work of Nicaud and give an average state complexity for the $\frac{1}{2}(\cdot)$ operation on unary languages.

For any unary DFA M, let $loop(M)$ denote the automaton formed by removing the tail of M. An n-loop is a unary automaton which is a loop of n states. Let $\mathcal{U}_n$ denote the set of complete, deterministic and initially connected unary DFAs with n states. We may enumerate such unary automata using the following notation. Let $M(n,k,F)$ denote the automaton $(\{0,\dots,n-1\},\{a\},0,\delta,F)$ with $\delta(i,a)=i+1$ if $0 \le i < n-1$ and $\delta(n-1,a)=k$.

Theorem 20 *The average complexity of the $\frac{1}{2}(\cdot)$ operation on an n-state unary automaton is equal to $(\frac{5}{8}n+c)(1+\lambda(n))$ for some function λ satisfying $\lambda(n) \to 0$ as $n \to \infty$ and some constant c.*

This tells us that in the average case we differ from the worst case by only a constant factor, but we are closer to the intuition that $\frac{1}{2}(L)$ should only take half as many states as L. We first state a few preliminary lemmas which are direct interpretations of the action of $\frac{1}{2}(\cdot)$ on unary languages.

Lemma 21 *Let $M=(Q,\{a\},\delta,q_0,F)$ be the minimal unary DFA for L, and suppose that M has $k \ge 0$ states in its tail. Then any automaton recognizing $\frac{1}{2}(L)$ needs at most $\lceil \frac{k}{2} \rceil$ states in its tail.* □

Lemma 22 *Let $M=(Q,\{a\},\delta,q_0,F)$ be the minimal unary DFA for L, and suppose that M has $r \ge 0$ states in its loop. Then any automaton recognizing $\frac{1}{2}(L)$ needs at most r states in its loop if r is odd, and $r/2$ states in its loop if r is even.* □

These lemmas are easily proved by direct observation. For each n-loop $\mathcal{L}$, define $k(\mathcal{L})$ as

$$k(\mathcal{L}) = \min\{k \mid \delta(F,a^k)=F\}$$

$k(\mathcal{L})$ exists since n is an integer satisfying the above condition. Consider the following lemma of Nicaud [5, Lemma 2].

Lemma 23 *For each n-loop $\mathcal{L}$, the minimal automaton of $\mathcal{L}$ has $k(\mathcal{L})$ states and $k(\mathcal{L})|n$. In particular, an n-loop is minimal iff $k(\mathcal{L})=n$.* □

From this we may observe the following lemma.

Lemma 24 *Let M be an automaton with $n=|loop(M)|$. If n is even (resp. odd) and $loop(M)$ is minimal, then for all M' with $L(M')=\frac{1}{2}(L(M))$, $|loop(M')| \ge \frac{n}{2}$ (resp. $\ge n$).* □

We also have the following lemma due to Nicaud [5, Thm. 1].

Lemma 25 *There are exactly* $L(n) = \sum_{d|n} \mu(n/d)2^d$ *minimal* n*-loops. Further, there exists a function* $\lambda(n)$ *such that* $L(n) = 2^n(1+\lambda(n))$ *and* $\lambda(n) \to 0$ *as* $n \to \infty$.□

We may now prove the statement of Theorem 20:

Proof. We are interested in examining

$$\frac{1}{|\mathcal{U}_n|} \sum_{M \in \mathcal{U}_n} \mathrm{sc}\Big(\frac{1}{2}(L(M))\Big). \tag{6}$$

By [5] or by direct observation, we know $|\mathcal{U}_n| = n2^n$. We first show an upper bound for (6). Let $[n]$ denote the set $\{0, 1, \ldots, n-1\}$. Consider that

$$\begin{aligned}
\sum_{M \in \mathcal{U}_n} \mathrm{sc}(\frac{1}{2}(L(M))) &= \sum_{k=0}^{n-1} \sum_{F \subseteq [n]} \mathrm{sc}\Big(\frac{1}{2}(L(M(n,k,F)))\Big) \\
&= \sum_{\substack{k=0 \\ k-n \text{ odd}}}^{n-1} \sum_{F \subseteq [n]} \mathrm{sc}\Big(\frac{1}{2}(L(M(n,k,F)))\Big) \\
&\quad + \sum_{\substack{k=0 \\ k-n \text{ even}}}^{n-1} \sum_{F \subseteq [n]} \mathrm{sc}\Big(\frac{1}{2}(L(M(n,k,F)))\Big) \\
&\le \sum_{\substack{k=0 \\ k-n \text{ odd}}}^{n-1} \sum_{F \subseteq [n]} \Big(n - \frac{k}{2} + 1\Big) + \sum_{\substack{k=0 \\ k-n \text{ even}}}^{n-1} \sum_{F \subseteq [n]} \Big(\frac{n}{2} + 1\Big).
\end{aligned}$$

Since there are 2^n possible choices for $F \subseteq [n]$, we may replace this accordingly:

$$\sum_{M \in \mathcal{U}_n} \mathrm{sc}\Big(\frac{1}{2}(L(M))\Big) \le \sum_{\substack{k=0 \\ k-n \text{ odd}}}^{n-1} \Big(n - \frac{k}{2} + 1\Big)2^n + \sum_{\substack{k=0 \\ k-n \text{ even}}}^{n-1} \Big(\frac{n}{2} + 1\Big)2^n.$$

Thus we have that

$$\frac{1}{|\mathcal{U}_n|} \sum_{M \in \mathcal{U}_n} \mathrm{sc}\Big(\frac{1}{2}(L(M))\Big) \le \frac{3}{4} + \frac{5}{8}n$$

which establishes the upper bound. For the lower bound, we follow the idea of Nicaud [5]. First, we will construct a set $G(\ell)$ of ℓ-loops which are minimal. Then we note that by Lemma 24

$$\sum_{M \in \mathcal{U}_n} \mathrm{sc}\Big(\frac{1}{2}(L(M))\Big) \ge \sum_{\ell=1}^{n} \sum_{\mathcal{L} \in G(\ell)} \sum_{\substack{M \in \mathcal{U}_n \\ loop(M) = \mathcal{L}}} h(\ell),$$

where $h : \mathbb{N} \to \mathbb{N}$ is the function defined by $h(r) = r$ if r is odd and $h(r) = r/2$ if r is even. Further, for every ℓ-loop $\mathcal{L}$ with $1 \leq \ell \leq n$, there are exactly $2^{n-\ell}$ n-state automata whose loop is $\mathcal{L}$. Thus

$$\sum_{M \in \mathcal{U}_n} \mathrm{sc}\Big(\frac{1}{2}(L(M))\Big) \geq 2^n \sum_{\ell=1}^{n} |G(\ell)| 2^{-\ell} h(\ell).$$

Thus, we must simply construct a set $G(\ell)$ which is large enough so that

$$2^n \sum_{\ell=1}^{n} |G(\ell)| 2^{-\ell} h(\ell) = \Big(\frac{5}{8}n^2 + cn\Big)(1 + \lambda(n))$$

for some λ with $\lambda(n) \to 0$ as $n \to \infty$ and some constant c. But from Lemma 25, we know that we can choose $G(\ell)$ to be the set of all minimal ℓ-loops, whereby we will get $|G(\ell)| = 2^\ell(1 + \lambda(\ell))$ for a function λ satisfying those constraints. Thus,

$$2^n \sum_{\ell=1}^{n} |G(\ell)| 2^{-\ell} h(\ell) = 2^n \sum_{\ell=1}^{n} h(\ell)(1 + \lambda(\ell)).$$

But we can easily observe that for all n, we get

$$\sum_{\ell=1}^{n} h(\ell) = \frac{5}{8}n^2 + O(n)$$

This gives the result. $\square$

7. Conclusions and Further Work

In this paper we have considered the state complexity of the operation $\frac{1}{2}(\cdot)$. We have shown an upper bound of $ne^{\sqrt{n \log n}}$ for arbitrary languages, and a lower bound which is matching up to a factor of n. In the unary case, the matching upper and lower bounds are n. In section 5, we have given a condition on automata such that the upper bound for $\frac{1}{2}(\cdot)$ is $O(n^3)$.

In section 4, we considered the state complexity of polynomial removals. However, as we noted above, the result of Seiferas and McNaughton [6] gives a much broader classification of relations which preserve regularity. It remains open whether or not one can give an expression for the worst case state complexity of a relation $\{(n, f(n)) \mid n \geq 0\}$ in terms of an arbitrary function f.

Câmpeanu et al. have considered the state complexity of operations on finite languages [2]. The state complexity of proportional removals on finite languages remains open.

Acknowledgements

Thanks to Jeff Shallit for proposing this problem and for carefully reading this paper. The author would also like to thank both the referees of DCAGRS 2001 and the referees of J.ALC for their many helpful suggestions which improved the presentation of the results in this paper.

References

[1] E. Bach, J. Shallit, *Algorithmic Number Theory, Volume I: Efficient Algorithms.* MIT Press, Cambridge, Mass., 1996.

[2] C. Câmpeanu, K. Culik II, K. Salomaa, S. Yu, State complexity of basic operations on finite languages. In: *Proc. 4rd Workshop on Implementing Automata, 1999.* LNCS **2214**, Springer-Verlag, 2001, 60–70.

[3] M. Chrobak, Finite automata and unary languages. *Theoretical Computer Science* **47** (1986), 149–158.

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

[5] C. Nicaud, Average state complexity of operations on unary automata. In: M. Kutylowski, L. Pacholski, T. Wierzbicki (eds.), *Proc. 24th Int. Symp. on Mathematical Foundations of Computer Science, 1999.* LNCS **1672**, Springer-Verlag, 1999, 230–240.

[6] J. I. Seiferas, R. McNaughton, Regularity-preserving relations. *Theoretical Computer Science* **2** (1976) 147–154.

[7] D. Siefkes, Decidable extensions of monadic second order successor arithmetic. In: *Automatentheorie und formale Sprachen.* Bibliographisches Institut, Mannheim, 1970, 441–472.

[8] S. Yu, Regular languages. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Language, Vol. 1.* Springer-Verlag, Berlin, 1997, 46–110.

[9] S. Yu, Q. Zhuang, K. Salomaa, The state complexities of some basic operations on regular languages. *Theoretical Computer Science* **125** (1994), 315–328.

(Received: November 6, 2001; revised: September 12, 2002)