Note on Minimal Automata and Uniform Communication Protocols[1]

Galina Jirásková

Mathematical Institute
Slovak Academy of Sciences
Košice, Slovakia
jiraskova@duro.upjs.sk

Abstract. In this paper, we describe regular languages with an essential difference between their nondeterministic message complexity and the size of their minimal nondeterministic finite automata. This solves an open problem posed by Hromkovič [2]. We also define a two-way message complexity and we show that the two-way message complexity of a regular language L provides a lower bound on the size of the minimal two-way deterministic finite automaton for L. We find specific regular languages with an exponential gap between these two complexity measures and we also do the same for the nondeterministic case.

1 Introduction

The communication complexity of two-party protocols is well-established as a successful method for proving lower bounds on several fundamental complexity measures of sequential and parallel computations. In this paper,

[1]Supported by grant 2/7007/20.

we study how to use communication protocols to prove lower bounds on the size of minimal finite automata.

It is well-known that the one-way communication complexity of a regular language L provides a direct lower bound on the logarithm of the number of states of the minimal finite automaton recognizing L [1, 2]. In order to establish a closer relation between communication complexity and finite automata, Hromkovič and Schnitger [4]a introduced a uniform model of two-party communication protocols. They defined the message complexity of a regular language L as the number of distinct messages used by the optimal one-way uniform communication protocol recognizing L and they showed that the message complexity of L provides a lower bound on the size of the minimal finite automaton for L. This relation was extended to the nondeterministic case by Hromkovič [2].

This, then, is the method for proving lower bounds on the size of minimal deterministic and nondeterministic finite automata and until now it has been the best method known for this purpose. It has been shown to be very successful in the deterministic case, in which the message complexity of a regular language L is exactly equal to the size of the minimal deterministic finite automaton for L [4, 2].

The aim of this paper is to show that this method has weaknesses in the cases of nondeterministic finite automata and of two-way (deterministic and nondeterministic) finite automata. We give specific regular languages for which the difference between their nondeterministic message complexity and the size of their minimal nondeterministic automata is exponential. Further, using two-way uniform protocols we define the two-way message complexity of a regular language L. We show that it provides a lower bound on the size of the minimal two-way finite automaton for L but that the difference between these two complexity measures may be exponential. The same holds for the nondeterministic case, too.

The paper is organized as follows. Section 2 contains the definitions of a nondeterministic and two-way message complexity. In Section 3, we give specific regular languages with an exponential difference between their nondeterministic message complexity and the size of their minimal nondeterministic finite automata. In Section 4, we study the relation between the two-way message complexity and two-way finite automata.

2 Definitions

To define one-way uniform nondeterministic protocols we follow [2]. Informally, a *one-way uniform nondeterministic protocol P over an alphabet Σ* accepting a regular language $L \subseteq \Sigma^*$ can be described as follows. The first computer (C_I) receives the first part x of an input $xy \in \Sigma^*$ and the second one (C_{II}) receives the rest y. The first computer looks at its input x and

nondeterministically sends binary messages to the second one. The second computer must then decide whether the input xy is in L or not (see [2] for details). The *message complexity of the protocol* P is the number of distinct messages that can be sent by the first computer. The *protocol* P *accepts a regular language* L *over the alphabet* Σ if for all $x, y \in \Sigma^*$ there is an accepting computation of P on xy if and only if $xy \in L$. The *nondeterministic message complexity of* L is the message complexity of the best one-way uniform nondeterministic protocol accepting L.

A *two-way uniform deterministic protocol* P accepting a regular language L over an alphabet Σ can be informally described as follows. The first computer (C_I) receives the first part x of an input $xy \in \Sigma^*$ and the second one (C_{II}) receives the rest y. Then they can communicate, i.e. exchange binary messages, until one of them knows whether the input xy is in L or not. The messages exchanged are not stored by the computers, i.e. the next message sent by a computer is a function of its input and the preceding message received from the other computer. The *two-way message complexity of the protocol* P is the number of distinct messages exchanged between the two computers. The *two-way message complexity of the language* L is the two-way message complexity of the best two-way uniform deterministic protocol P accepting L. Now, let us formalize these informal definitions of two-way uniform protocols.

Definition 1 *Let* Σ *be an alphabet and let* $L \subseteq \Sigma^*$. *A* two-way uniform protocol over Σ *is a pair* $P = \langle \Phi, \varphi \rangle$, *where:*

$$\Phi, \varphi : \Sigma^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\overline{0}, \overline{1}\}$$

are functions which have the prefix freeness property (i.e. $\Phi(x, c)$ *is not a proper prefix of* $\Phi(x', c)$*; the same for* φ*).*

A computation *of* P *on an input* $xy \in \Sigma^*$ *is a string* $c = c_1\$c_2\$\ldots \$c_k\$c_{k+1}$, *where* $k \geq 0$, $c_1, \ldots, c_k \in \{0,1\}^*, c_{k+1} \in \{\overline{0}, \overline{1}\}$ *and such that:*

(i) $c_1 = \Phi(x, \lambda)$,

(ii) if l *is odd, then* $c_{l+1} = \varphi(y, c_l)$,

(iii) if l *is even, then* $c_{l+1} = \Phi(x, c_l)$.

A computation $c = c_1\$c_2\$\ldots\$c_k\$c_{k+1}$ *is called* accepting (rejecting) *if* $c_{k+1} = \overline{1}$ ($\overline{0}$). *We say that a* protocol P accepts a language L *if for all* $x, y \in \Sigma^*$ *the computation of* P *on the word* xy *is accepting iff* $xy \in L$.

The two-way message complexity of the protocol P *is:*

$$2mc(P) = |\{\Phi(x,c) \mid x \in \Sigma^*, c \in \{0,1\}^*\} \cup \{\varphi(y,c) \mid y \in \Sigma^*, c \in \{0,1\}^*\}|,$$

i.e. the number of distinct messages used by the protocol P.

The two-way message complexity of a language L *is:*

$$2mc(L) = \min\{2mc(P) \mid P \text{ is a two-way uniform protocol accepting } L\}.$$

Definition 2 *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a (nondeterministic, two-way) finite automaton recognizing a regular language L over the alphabet Σ. The* size of the automaton M *is the number of its states (i.e. $|Q|$). A (nondeterministic, two-way) finite automaton M is called* minimal for L *if it recognizes L with the minimal number of states.*

3 Nondeterministic Message Complexity Versus Nondeterministic Finite Automata

Hromkovič [2] showed that the nondeterministic message complexity of a regular language L provides a lower bound on the size of the minimal nondeterministic finite automaton for L.

Klauck and Schnitger [5] showed that there are regular languages with an essential difference between the nondeterministic message complexity and the size of the minimal nondeterministic finite automaton.

In this section, we give specific regular languages whose nondeterministic message complexity is much smaller then the size of their minimal nondeterministic finite automata. For an even integer k, let:

$A_k = \{xy \in \{0,1\}^* \mid |x| = |y| = k, x \neq y \text{ or } x = ww\}$

be the regular language over the alphabet $\{0,1\}$ that contains words of length $2k$ with different halves or having equal halfs of the first half of the word.

Theorem 1 *The nondeterministic message complexity of A_k is $O(k^2)$.*

Proof. It is sufficient to show that there exists a one-way uniform nondeterministic protocol P_k over the alphabet $\{0,1\}$ accepting A_k with the message complexity $O(k^2)$. Let us informally describe it. The computation of the protocol P_k on a word $xy \in \{0,1\}^*$ is as follows.

(i) If $x = \lambda$ then C_I submits the message "I have no bit". In this case, C_{II} accepts (rejects) if $y \in A_k$ ($y \notin A_k$).

(ii) If $|x| > 2k$ then C_I submits the message "I have more than 2k bits" and C_{II} rejects the input xy.

(iii) If $1 \leq |x| < k$ then C_I nondeterministically sends the messages $(|x|, i, x_i), i = 1, \ldots, |x|$. Because computer C_{II} has the information about the length of x and the $i-th$ bit of x, it accepts if $|y| + |x| = 2k$ and the corresponding bit of y is different from x_i, or $|y| + |x| = 2k$ and the last k bits of y can be written in the form ww. Otherwise, it rejects.

(iv) If $k \leq |x| \leq 2k$, then C_I nondeterministically sends the messages $(|x|, i, x_i), i = 1, \ldots, k$ and the message whether the first k bits of x can be written in the form ww or not. With this information C_{II} can again accept or reject the input xy.

It is not difficult to see that P_k is a one-way uniform nondeterministic protocol over the alphabet $\{0,1\}$ accepting A_k with the message complexity $O(k^2)$. □

Theorem 2 *Any nondeterministic finite automaton recognizing the language A_k has at least $2^{k/6}$ states.*

Proof. Let $M = (Q, \{0,1\}, \delta, q_0, F)$ be a nondeterministic finite automaton recognizing the language A_k. For any two different words $u, v \in \{0,1\}^{k/2}$ the words $uuuu$ and $vvvv$ belong to A_k and so there are accepting computations of the automaton M on these words. Let $q_0, q_1, q_2, \ldots, q_{2k}$ be an accepting computation of M on $uuuu$ and $q_0, p_1, p_2, \ldots, p_{2k}$ be an accepting computation of M on $vvvv$ $(q_i, p_i \in Q)$. We claim that the 3-tuple $(q_{k/2}, q_k, q_{3k/2})$ is different from the 3-tuple $(p_{k/2}, p_k, p_{3k/2})$. If we assume the contrary, that $q_{k/2} = p_{k/2}, q_k = p_k$, and $q_{3k/2} = p_{3k/2}$, then $q_0, q_1, \ldots, q_{k/2} = p_{k/2}, p_{k/2+1}, \cdots, p_k = q_k, q_{k+1}, \cdots, q_{3k/2} = p_{3k/2}, p_{3k/2+1}, \cdots, p_{2k}$ is an accepting computation of M on the word $uvuv$ which does not belong to the language A_k, a contradiction. So, we have proved that $(q_{k/2}, q_k, q_{3k/2}) \neq (p_{k/2}, p_k, p_{3k/2})$. This implies that $|Q|^3$ is at least $2^{k/2}$ (the number of all words of length $k/2$) and so the number of states of M is at least $2^{k/6}$. □

Corollary 1 *For any language A_k considered above, there is an exponential difference between the nondeterministic message complexity of A_k and the size of the minimal nondeterministic finite automaton for A_k.*

4 Two-Way Message Complexity Versus Two-Way Finite Automata

In this section, we show that the two-way message complexity of a regular language L provides a lower bound on the size of the minimal two-way deterministic automaton for L. We also find specific regular languages with the exponential difference between these two complexity measures. These results hold in the nondeterministic case, too.

Theorem 3 *For any regular language L over an alphabet Σ, the two-way message complexity of L is not greater than the size of the minimal two-way deterministic automaton for L.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a minimal $2dfa$ for L. It is sufficient to prove that there is a two-way uniform deterministic protocol P accepting L with at most $|Q|$ messages. The computation of the protocol P on an input $xy \in \Sigma^*$ can be informally described as follows. The first computer looks at its input x and simulates the work of M on x until the last symbol of x is read by M and the step to the right has to be made by M. In this case, the first computer sends a message coding the state of M to the second computer. Then the second computer can simulate M on its input y until the first symbol of y is read by M and the step to the left has to be made by M. In this case, the second computer sends the message that codes the state of M to the first computer. Further, the communication between these two computers proceeds in the same way.It is not difficult to see that P is a two-way uniform deterministic protocol accepting L with the message complexity $|Q|$. □

In the following part of this section we give examples of regular languages with the essential difference between their two-way message complexity and the size of their minimal two-way deterministic automata.

Lemma 1 *For an integer k, let $B_k = \{1^k\}$ be the unary language that contains the only word (of length k). Any two-way deterministic finite automaton for B_k has at least k states.*

Proof. Assume to the contrary that $M = (Q, \{1\}, \delta, q_0, F)$ is a $2dfa$ for B_k that has fewer than k states. Since $1^k \in B_k$, the automaton M accepts the word 1^k. Let us consider the sequence of states which M enters when computing on 1^k. Denote by q_i $(i = 1, 2, \ldots, k)$ the state that M enters on the move which takes M to the $i+1$st cell for the first time (before reaching this cell M may move its head back and forth on cells 1 through i many times). Since M has less than k states, there exist $i < j$ such that $q_i = q_j$. But then the computation of M on the word $1^i 1^{j-i} 1^{j-i} 1^{k-j}$ of the length more than k is also accepting, which is a contradiction. □

Lemma 2 *Let $C_k = \{1^{k^k-1}\}$ be the unary language that contains the only word of length $k^k - 1$. There is a two-way uniform deterministic protocol P_k accepting C_k and using $O(k^2)$ messages.*

Proof. The protocol P_k over the alphabet $\{1\}$ can be informally described as follows. Let $xy \in \Sigma^*$ be an input word. If $|x| \geq k^k$ or $|y| \geq k^k$ a constant number of messages is sufficient for the computers C_I and C_{II} to reject the input. In the other case (i.e. $|x| < k^k$ and $|y| < k^k$) both computers can unambiguously write the length of their inputs in the form:

$C_I : |x| = a_0 + a_1 k + a_2 k^2 + \ldots + a_{k-1} k^{k-1}$,

$C_{II} : |y| = b_0 + b_1 k + b_2 k^2 + \ldots + b_{k-1} k^{k-1}$,

where $a_i, b_i \in \{0, 1, \ldots, k-1\}$. Note that $|x| + |y| = k^k - 1$ iff $a_i + b_i = k - 1$ for all $i = 0, 1, \ldots, k-1$. So, C_I will successively send a subscript i and the value of the coefficient a_i (starting with a_0) and C_{II} will check whether $a_i + b_i = k - 1$. $O(k^2)$ messages are sufficient for them to accept or reject the input xy (the number of the coefficients a_i is k and their values are from the set $\{0, 1, 2, \ldots, k-1\}$). □

By the two lemmata above, for the language $C_k = \{1^{k^k-1}\}$ it holds, that there is a two-way uniform deterministic protocol accepting C_k with the message complexity $O(k^2)$, while any two-way deterministic finite automaton recognizing $C_k = B_{k^k-1}$ has at least $k^k - 1$ states. So, there is an exponential difference between the two-way message complexity of C_k and the size of the minimal two-way deterministic automaton for C_k.

The two-way nondeterministic message complexity of regular languages can be defined simply by allowing Φ and φ in the definition 2.1 to be relations on $\Sigma^* \times \{0,1\}^* \times (\{0,1\}^* \cup \{\overline{0}, \overline{1}\})$, as opposed to functions. Theorem 3 and Lemma 1 can be proved for the nondeterministic case, too. So, we get that the two-way nondeterministic message complexity of a regular language L provides a lower bound on the size of the minimal two-way nondeterministic automaton for L. But for the language C_k the difference between these two complexity measures is exponential.

The languages considered in this section were over the alphabet $\{1\}$. Similar considerations can be made for the regular languages $\{(01)^{k^k-1}\}$ over the alphabet $\{0,1\}$ or the regular languages $\{(abc)^{k^k-1}\}$ over the alphabet $\{a, b, c\}$.

Acknowledgement

I am grateful to Prof. Juraj Hromkovič for his comments concerning this work.

References

[1] J. Hromkovič, Relation between Chomsky hierarchy and communication complexity hierarchy, *Acta Mathematica Univ. Com.*, 48-49 (1986), 311–317.

[2] J. Hromkovič, *Communication Complexity and Parallel Computing.* Springer, Berlin, 1997.

[3] J. Hromkovič and G. Schnitger, Determinismus versus Las Vegas, 1995, unpublished ms.

[4] J. Hromkovič and G. Schnitger, On the power of the number of advice bits in nondeterministic computations. In *Proceedings of the 28th ACM STOC*, 1996, 551–560.

[5] H. Klauck and G. Schnitger, Nondeterministic finite automata versus nondeterministic uniform communication complexity, University of Frankfurt, 1996, unpublished ms.