On the Non-Deterministic Communication Complexity of Regular Languages

Anil Ada*

Abstract

In this paper we study the non-deterministic communication complexity of regular languages. We show that a regular language has either constant or at least logarithmic non-deterministic communication complexity. We prove several linear lower bounds which we know cover a wide range of regular languages with linear complexity. Furthermore we find evidence that previous techniques (Tesson and Thérien 2005) for proving linear lower bounds, for instance in deterministic and probabilistic models, do not work in the non-deterministic setting.

1 Introduction

The notion of communication complexity was introduced by Yao [16] in light of its applications to parallel computers. Following this seminal work, it has been shown to have many more applications where the need for communication is not explicit and thus has become the "Swiss Army knife" of complexity theory. These applications include time/space lower bounds for VLSI chips [9], time/space tradeoffs for Turing Machines [3], data structures [9], boolean circuit lower bounds [6, 8], pseudorandomness [3], separation of proof systems [4] and lower bounds on the size of polytopes representing NP-complete problems [15].

It is an intriguing task to better understand the landscape of communication complexity and thus other areas of complexity theory. A natural starting point is to comprehend the complexity of regular languages, which in some sense are the simplest languages with respect to the usual time/space complexity framework. Perhaps surprisingly, regular languages form a non-trivial case study with respect to communication complexity. There are hard regular languages even in very powerful models of communication complexity. Furthermore, some of the very well-known and studied functions in this area such as Disjointness and Inner Product are equivalent to regular languages from a communication complexity perspective.

*School of Computer Science, McGill University. aada@cs.mcgill.ca. Supported by the research grants of Prof. Denis Thérien.

In [13], it was established that the class of regular languages having $O(f)$ deterministic communication complexity forms a language variety and so the question of the communication complexity of regular languages has an algebraic answer. In a follow up work [14], a complete algebraic characterization of the communication complexity of regular languages was established in the deterministic, simultaneous, probabilistic, simultaneous probabilistic and Mod_p-counting models. These results unmasked an interesting complexity gap: In all of the above models, the complexity of a regular language falls into one of four classes $O(1), \Theta(\log \log n), \Theta(\log n)$ or $\Theta(n)$. In contrast, we note that for any increasing function f with $1 \leq f \leq n$, it is possible to construct a non-regular language with complexity $\Theta(f)$ for any of these models.

In this paper we are interested in the non-deterministic communication complexity of regular languages. To get a similar characterization for the non-deterministic model, one needs the notions of *positive language varieties* and *ordered monoids*. This is because the syntactic monoid of a regular language does not distinguish between a language and its complement. Differing from the models mentioned earlier, non-deterministic complexity of a function and its complement may not be equal. So regular languages having $O(f)$ non-deterministic communication complexity do not form a variety but a positive variety.

Adopting this refined approach, we take the first steps towards a complete classification for the non-deterministic communication complexity of regular languages. We identify the regular languages having constant non-deterministic complexity. We show that if a regular language does not have constant complexity then it has $\Omega(\log n)$ complexity, revealing a complexity gap. We also obtain several linear lower bound results which we know cover a wide range of regular languages having linear complexity. These bounds point out sufficient conditions for not being in the positive variety $Pol(\mathcal{C}om)$, providing us with some nice combinatorial intuition about this variety. Finally we find evidence that previous techniques used in [14] for proving linear lower bounds, for instance in deterministic and probabilistic models, do not work in the non-deterministic setting.

Organization. In Sect. 2 and Sect. 3, we give the necessary background on algebraic automata theory and communication complexity respectively. In Sect. 4, we define the communication complexity of a regular language and a monoid. Furthermore, we show that the non-deterministic communication complexity of regular languages admits an algebraic characterization. Section 5 is devoted to the bounds we have on the non-deterministic communication complexity of regular languages and ordered monoids.

2 Algebraic Automata Theory

In this section we set some notation and recall the definitions we need from algebraic automata theory. We refer the reader to [11] for further background with an emphasis on the more general theory of ordered monoids.

A *monoid* $(M,\cdot)$ is a set M together with an associative binary operation $\cdot$ and an identity $1_M \in M$ which satisfies $1_M \cdot m = m \cdot 1_M = m$ for any $m \in M$. An *order* relation on a set S is a relation that is reflexive, anti-symmetric and transitive and it is denoted by $\leq$. We say that $\leq$ is a *stable order relation* on a monoid M if for all $x, y, z \in M$, $x \leq y$ implies $zx \leq zy$ and $xz \leq yz$. An *ordered monoid* $(M, \leq_M)$ is a monoid M together with a stable order relation $\leq_M$ that is defined on M. A *morphism of ordered monoids* $\Phi : (M, \leq_M) \to (N, \leq_N)$ is a morphism between M and N that also preserves the order relation, i.e. for all $m, m' \in M$, $m \leq_M m'$ implies $\Phi(m) \leq_N \Phi(m')$.

A subset $I \subseteq M$ is called an *order ideal* if for any $y \in I$, $x \leq_M y$ implies $x \in I$. Every order ideal I in a finite monoid M has a generating set $x_1, \ldots, x_k$ such that $I = \langle x_1, \ldots, x_k \rangle := \{y \in M : \exists x_i \text{ with } y \leq_M x_i\}$. We say that a language $L \subseteq \Sigma^*$ is *recognized* by an ordered monoid $(M, \leq_M)$ if there exists a morphism of ordered monoids $\Phi : (\Sigma^*, =) \to (M, \leq_M)$ and an order ideal $I \subseteq M$ such that $L = \Phi^{-1}(I)$.

Define the *syntactic congruence* as follows: $x \equiv_L y$ if for all $u, v \in \Sigma^*$ we have $uxv \in L$ iff $uyv \in L$. The *syntactic monoid* is the quotient monoid $M(L) = \Sigma^* / \equiv_L$. Let $x \preceq_L y$ if for all $u, v \in \Sigma^*$, $uyv \in L \implies uxv \in L$. So $x \equiv_L y$ if and only if $x \preceq_L y$ and $y \preceq_L x$. Now $\preceq_L$ induces a well-defined stable order $\leq_L$ on $M(L)$ given by $[x] \leq_L [y]$ if and only if $x \preceq_L y$. The ordered monoid $(M(L), \leq_L)$ is the *syntactic ordered monoid* of L.

We say that an ordered monoid $(N, \leq_N)$ *divides* an ordered monoid $(M, \leq_M)$ if there exists a surjective morphism of ordered monoids from a submonoid of $(M, \leq_M)$ onto $(N, \leq_N)$. We know that $(M(L), \leq_L)$ recognizes L and divides any other ordered monoid that also recognizes L.

We say that a family of ordered monoids $\mathbf{V}$ is a *variety of ordered monoids* if it is closed under division of ordered monoids and finite direct product. The order in a finite direct product $M_1 \times \cdots \times M_n$ is given by $(m_1, \ldots, m_n) \leq (m'_1, \ldots, m'_n)$ iff $m_i \leq m'_i \quad \forall i \in [n]$.

A *class of regular languages* is a function $\mathcal{C}$ that maps every alphabet Σ to a set of regular languages in Σ^*. A class of languages $\mathcal{C}$ is called a *positive variety of languages* if

(i) for any alphabet Σ, $\mathcal{C}(\Sigma)$ is closed under finite intersection and finite union,

(ii) $\mathcal{C}$ is closed under inverse morphisms: given any alphabets Σ and Γ, for any morphism $\Phi : \Sigma^* \to \Gamma^*$, if $L \in \mathcal{C}(\Gamma)$ then $\Phi^{-1}(L) \in \mathcal{C}(\Sigma)$,

(iii) $\mathcal{C}$ is closed under left and right quotients: for $L \in \mathcal{C}(\Sigma)$ and $s \in \Sigma$, we have $s^{-1}L := \{w \in L \mid sw \in L\}$ and $Ls^{-1} := \{w \in L \mid ws \in L\}$ are in $\mathcal{C}(\Sigma)$.

Given a variety of finite ordered monoids $\mathbf{V}$, let $\mathcal{V}(\Sigma)$ be the set of languages over Σ whose syntactic ordered monoid belongs to $\mathbf{V}$. This is equivalent to saying that $\mathcal{V}(\Sigma)$ is the set of languages over Σ that are recognized by an ordered monoid in $\mathbf{V}$. The Variety Theorem, originally due to Eilenberg [5] and adapted to the ordered case by Pin [10], is as follows.

Theorem 2.1 (The Variety Theorem). *$\mathcal{V}$ is a positive variety of languages and the mapping $\mathbf{V} \mapsto \mathcal{V}$ defines a one to one correspondence between the varieties of finite ordered monoids and the positive varieties of languages.*

The *polynomial closure* of a set of languages $\mathcal{L}$ in Σ^* is a family of languages such that each of them is a finite union of $L_0a_1L_1 \cdots a_kL_k$, where $k \geq 0$, $a_i \in \Sigma$ and $L_i \in \mathcal{L}$. If $\mathcal{V}$ is a variety of languages, then we denote by $Pol(\mathcal{V})$ the class of languages that is the polynomial closure of $\mathcal{V}$. We know that $Pol(\mathcal{V})$ is a positive variety [12].

We say that the concatenation $L_0a_1L_1 \cdots a_kL_k$ is *unambiguous* if all words $x \in L_0a_1L_1 \cdots a_kL_k$ has a unique factorization $x = w_0a_1w_1 \cdots a_kw_k$ with $w_i \in L_i$. We denote by $UPol(\mathcal{V})$ the variety of languages consisting of *disjoint* unions of unambiguous concatenations $L_0a_1L_1 \cdots a_kL_k$ with $L_i \in \mathcal{V}$ (in some sense, there is only one witness for x in $L \in UPol(\mathcal{V})$). Similarly we denote by $Mod_pPol(\mathcal{V})$ the language variety generated by the languages for which membership depends on the number of factorizations mod p.

An element $e \in M$ is called *idempotent* if $e^2 = e$. For any finite M, there is a number $k > 0$ such that for every element $m \in M$, m^k is an idempotent. We call k an *exponent* of M.

3 Communication Complexity

We present here a quick introduction to communication complexity but refer the reader to the great book of Kushilevitz and Nisan [9] for further details.

In the deterministic model, two players, Alice and Bob, wish to compute a function $f : S^{n_A} \times S^{n_B} \to T$ where S and T are finite sets. Alice is given $x \in S^{n_A}$ and Bob $y \in S^{n_B}$ and they collaborate in order to obtain $f(x, y)$ by exchanging bits using a common blackboard according to some predetermined *communication protocol* $\mathcal{P}$. This protocol determines whose turn it is to write, furthermore what a player writes is a function of that player's input and the information exchanged thus far. When the protocol ends, its output $\mathcal{P}(x, y) \in T$ is a function of the blackboard's content. We say that $\mathcal{P}$ computes f if $\mathcal{P}(x, y) = f(x, y)$ for all x, y and define the *cost* of $\mathcal{P}$ as the maximum number of bits exchanged for any input. The *deterministic communication complexity* of f, denoted $D(f)$ is the cost of the cheapest protocol computing f. We will be interested in the complexity of functions $f : S^* \times S^* \to T$ and will thus consider $D(f)$ as a function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$ and study its asymptotic behaviour.

In a *non-deterministic communication protocol* $\mathcal{P}$ another player, say God, having access to *both* x and y first sends to Alice and Bob a proof π. Alice and Bob then follow an ordinary deterministic protocol $\mathcal{P}'$ with output in $\{0, 1\}$. The protocol $\mathcal{P}$ accepts the input (x, y) if and only if there is some proof π such that the output of the ensuing deterministic protocol $\mathcal{P}'$ outputs 1. The cost of a non-deterministic protocol is the maximum number of bits exchanged in the protocol (*including* the bits of π) for any input (x, y). We denote the non-deterministic communication complexity of a language

L as $N^1(L)$. The co-non-deterministic communication complexity of L, denoted $N^0(L)$ is the non-deterministic communication complexity of L's complement.

Let $PDISJ$ be the following promise problem. Alice gets a set $x \subseteq [n]$ and Bob a set $y \subseteq [n]$ with the guarantee that $|x \cap y| \leq 1$ and $PDISJ(x, y) = 1$ if and only if $x \cap y = \emptyset$. One can show $N^1(PDISJ) = \Omega(n)$ (see Section 2.2.3 of [1]). Define two more problems: $LT(x, y) = 1$ iff $x \leq y$ when x and y are viewed as n-bit integers; $IP_q(x, y) = 1$ iff $\sum_{i=1}^{n} x_i y_i \equiv 0 \mod q$. It is well known that both functions have $\Omega(n)$ non-deterministic communication complexity.

Communication complexity classes were introduced in [2] in which an "efficient" protocol was defined to have cost no more than poly-logarithmic, i.e. $O(\log^c n)$ for a constant c. Thus one obtains communication complexity classes analogous to P and NP in the following way: $P^{cc} := \{f | D(f) = polylog(n)\}$, $NP^{cc} := \{f | N^1(f) = polylog(n)\}$.

4 Algebraic Approach to Communication Complexity

In general, we want to study the communication complexity of functions which do not explicitly have two inputs. In the case of regular languages and ordered monoids we use a form of *worst-case partition* definition. Formally, the communication complexity of a pair (M, I) where M is a finite ordered monoid and I is an order ideal in M is the communication complexity of the monoid evaluation problem corresponding to M and I: Alice is given $m_1, m_3, \ldots, m_{2n-1}$ and Bob is given $m_2, m_4, \ldots, m_{2n}$ such that each $m_i \in M$. They want to decide if the product $m_1 m_2 \cdots m_{2n}$ is in I. The communication complexity of M is the maximum complexity of (M, I) where I ranges over all order ideals in M.

Similarly, the *communication complexity of a regular language* $L \subseteq A^*$ is the communication complexity of the following problem: Alice and Bob respectively receive $a_1, a_3, \ldots a_{2n-1}$ and $a_2, a_4, \ldots, a_{2n}$ where each a_i is either in A or is the neutral letter ϵ and they want to determine whether $a_1 a_2 \cdots a_{2n}$ belongs to L.

The following two lemmas establish the soundness of an algebraic approach to the communication complexity of regular languages.

Lemma 4.1. *Let $L \subseteq A^*$ be regular and $M = M(L)$. Then $N^1(M) = \Theta(N^1(L))$.*

Proof. It is straightforward to show $N^1(L) = O(N^1(M))$. To show $N^1(M) = O(N^1(L))$, we present a protocol for (M, I) where $I = \langle i_1, ..., i_k \rangle$ is some order ideal in M.

Let Φ be the accepting morphism. For each monoid element m, fix a word that is in the preimage of m under Φ, and denote it by w_m. Let $Y_a := \{(u, v) : uav \in L\}$. Recall that $a \preceq_L b$ if for all $u, v \in \Sigma^*$, $ubv \in L \implies uav \in L$. So $\Phi(a) \leq_L \Phi(b)$ iff $a \preceq_L b$ iff $Y_b \subseteq Y_a$. For each Y_a and Y_b with $Y_b \not\subseteq Y_a$, pick (u, v) such that $(u, v) \in Y_b$ but $(u, v) \notin Y_a$. Let K be the set of all these (u, v). One can think of K as containing a witness for $Y_b \not\subseteq Y_a$ for each such pair. Note that K is finite. Now pad each w_m and each word appearing in a pair in K with the neutral letter ϵ so that each of these words have the same constant length.

Now the protocol is as follows. Suppose Alice is given $m_1^a, m_2^a, \ldots, m_n^a$ and Bob is given $m_1^b, m_2^b, \ldots, m_n^b$. For each i_j they want to determine if $m_1^a m_1^b \cdots m_n^a m_n^b \preceq_L i_j$. This is equivalent to determining if $w_{m_1^a m_1^b \cdots m_n^a m_n^b} \preceq_L w_{i_j}$, which is equivalent to $w_{m_1^a} w_{m_1^b} \cdots w_{m_n^a} w_{m_n^b} \preceq_L w_{i_j}$. If this is not the case, $Y_{w_{i_j}} \not\subseteq Y_{w_{m_1^a} w_{m_1^b} \cdots w_{m_n^a} w_{m_n^b}}$ and so there will be a witness of this in K, i.e. there exists (u, v) such that $uw_{i_j}v \in L$ but $uw_{m_1^a} w_{m_1^b} \cdots w_{m_n^a} w_{m_n^b} v \notin L$. If indeed $w_{m_1^a} w_{m_1^b} \cdots w_{m_n^a} w_{m_n^b} \preceq_L w_{i_j}$ then for each $(u, v) \in K$ with $uw_{i_j}v \in L$, we will have $uw_{m_1^a} w_{m_1^b} \cdots w_{m_n^a} w_{m_n^b} v \in L$. Using the protocol for L, Alice and Bob check which of the two cases is true. □

In particular the non-deterministic complexity of an ordered monoid M is, up to a constant, the maximal communication complexity of any regular language that it can recognize.

Lemma 4.2. *For any increasing $f : \mathbb{N} \to \mathbb{N}$ the class of monoids such that $N^1(M)$ is $O(f)$ forms a variety of ordered monoids.*

Proof. The closure of this class under direct product is obvious. Suppose $N \prec M$, so there is a surjective morphism ϕ from a submonoid M' of M onto N. Denote by $\phi^{-1}(n)$ a fixed element from the preimage of n. Let I be an order ideal in N. A protocol for (N, I) is as follows. Alice is given $n_1^a, n_2^a, \ldots, n_t^a$ and Bob is given $n_1^b, n_2^b, \ldots, n_t^b$. They want to decide if $n_1^a n_1^b \ldots n_t^a n_t^b \in I$. This is equivalent to deciding if $\phi^{-1}(n_1^a)\phi^{-1}(n_1^b) \cdots \phi^{-1}(n_t^a)\phi^{-1}(n_t^b) \in \phi^{-1}(I)$. It is easy to see $\phi^{-1}(I)$ is an order ideal in M' so Alice and Bob can use the protocol for M' to decide if the above is true. Therefore we have $N^1(N) \leq N^1(M')$. It is straightforward to check that $N^1(M') \leq N^1(M)$ and so $N^1(N) \leq N^1(M)$ as required. □

To compare the communication complexity of two languages K, L in different models, Babai et al. [2] defined *rectangular reductions* from K to L which are, intuitively, reductions which can be computed privately by Alice and Bob without any communication cost. We give here a form of this definition which specifically suits our needs. Let $u = u_1 u_2 \cdots u_k$ be a word over M, i.e. $u \in M^*$. We denote by $eval(u)$ the corresponding monoid element, i.e. $eval(u) = u_1 \cdot u_2 \cdots u_k$.

Definition 4.3. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, M a finite ordered monoid and I an order ideal in M. A *rectangular reduction* of length t from f to (M, I) is a sequence of $2t$ functions $a_1, b_2, a_3, \ldots, a_{2t-1}, b_{2t}$ with $a_i : \{0,1\}^n \to M$ and $b_i : \{0,1\}^n \to M$ and such that for every $x, y \in \{0,1\}^n$ we have $f(x, y) = 1$ if and only if $eval(a_1(x)b_2(y) \cdots b_{2t}(y))$ is in I.

Such a reduction transforms an input (x, y) of the function f into a sequence of $2t$ monoid elements $m_1, m_2, \ldots, m_{2t}$ where the odd-indexed m_i are obtained as a function of x only and the even-indexed m_i are a function of y.

We write $f \leq_r^t (M, I)$ to indicate that f has a rectangular reduction of length t to (M, I). When $t = O(n)$ we omit the superscript t. It should be clear that if $f \leq_r^t (M, I)$ and f has communication complexity $\Omega(g(n))$, then (M, I) has communication complexity $\Omega(g(t^{-1}(n)))$.

We will be interested in a special kind of rectangular reduction which we call a *local rectangular reduction*. In a local rectangular reduction, Alice converts each bit x_i to a sequence of s monoid elements $m_{i,1}^a, m_{i,2}^a, \ldots, m_{i,s}^a$ by applying a fixed function $a : \{0,1\} \to M^s$. Similarly Bob converts each bit y_i to a sequence of s monoid elements $m_{i,1}^b, m_{i,2}^b, \ldots, m_{i,s}^b$ by applying a fixed function $b : \{0,1\} \to M^s$. Then $f(x, y) = 1$ iff $eval(m_{1,1}^a m_{1,1}^b \cdots m_{1,s}^a m_{1,s}^b \cdot \cdots \cdot m_{n,1}^a m_{n,1}^b \cdots m_{n,s}^a m_{n,s}^b) \in I$. The reduction transforms an input (x, y) into a sequence of $2sn$ monoid elements. Let $a(z)_k$ denote the k^{th} coordinate of the tuple $a(z)$. We specify this kind of local transformation with a $2 \times 2s$ matrix:

$a(0)_1$	$b(0)_1$	$a(0)_2$	$b(0)_2$	$\cdots$	$\cdots$	$a(0)_s$	$b(0)_s$
$a(1)_1$	$b(1)_1$	$a(1)_2$	$b(1)_2$	$\cdots$	$\cdots$	$a(1)_s$	$b(1)_s$

It is convenient to see which words the transformation produces for all possible values of x_i and y_i. For simplicity let us assume s is even.

x_i	y_i	corresponding word over M
0	0	$a(0)_1 b(0)_1 \cdots a(0)_s b(0)_s$
0	1	$a(0)_1 b(1)_1 a(0)_2 b(1)_2 \cdots a(0)_s b(1)_s$
1	0	$a(1)_1 b(0)_1 a(1)_2 b(0)_2 \cdots a(1)_s b(1)_s$
1	1	$a(1)_1 b(1)_1 \cdots a(1)_s b(1)_s$

5 Bounds for Regular Languages and Monoids

5.1 Classification Results

Lemma 5.1 ([14])**.** *If M is commutative then $D(M) = O(1)$ and thus $N^1(M) = O(1)$.*

Lemma 5.2 (Adapted from [14])**.** *If M is not commutative then for any order on M we have $N^1(M) = \Omega(\log n)$.*

Proof. Since M is not commutative, there must be $a, b \in M$ such that $ab \neq ba$. Therefore either $ab \not\leq_M ba$ or $ba \not\leq_M ab$. W.l.o.g. assume $ba \not\leq_M ab$. Let $I = \langle ab \rangle$. We show that $LT \leq_r^{2^n} (M, I)$. Alice gets x and constructs a sequence of 2^n monoid elements in which a is in position x and 1_M is in everywhere else. Bob gets y and constructs a sequence of 2^n monoid elements in which b is in position y and 1_M is everywhere else. If $x \leq y$ then the product of the monoid elements is ab which is in I. If $x > y$ then the product is ba which is not in I. ∎

Denote by $\mathcal{C}om$ the positive language variety corresponding to the variety of commutative monoids **Com**. The above two results show that regular languages that have constant non-deterministic communication complexity are exactly those languages in $\mathcal{C}om$.

Lemma 5.3. *If $L \subseteq A^*$ is a language of $Pol(\mathcal{C}om)$ then $N^1(L) = O(\log n)$.*

Proof. Suppose L is a union of t languages of the form $L_0 a_1 L_1 \cdots a_k L_k$. Alice and Bob know beforehand the value of t and the structure of each of these t languages. So a protocol for L is as follows. Assume Alice is given $x_1^a, \ldots, x_n^a$ and Bob is given $x_1^b, \ldots, x_n^b$. God communicates to Alice and Bob which of the t languages the word $x_1^a x_1^b \cdots x_n^a x_n^b$ resides in. This requires a constant number of bits to be communicated since t is a constant. Then God communicates the positions of each a_i. This requires $k \log n$ bits of communication where k is a constant. The validity of the information communicated by God can be checked by Alice and Bob by checking if the words in between the a_i's belong to the right languages. Since these languages are in $\mathcal{C}om$, this can be done in constant communication. Therefore in total we require only $O(\log n)$ communication. □

From the above proof, we see that we can actually afford to communicate $O(\log n)$ bits to check that the words between the a_i's belong to the corresponding language. In other words, we could have $L_i \in Pol(\mathcal{C}om)$. Note that this does not mean that this protocol works for a strictly bigger class since $Pol(Pol(\mathcal{C}om)) = Pol(\mathcal{C}om)$.

Denote by UP the subclass of NP in which the languages are accepted by a non-deterministic Turing Machine having *exactly* one accepting path (or one witness) for each string in the language. It is known that $UP^{cc} = P^{cc}$ [15]. From [14] we know that regular languages having $O(\log n)$ deterministic communication complexity are exactly those languages in $UPol(\mathcal{C}om)$ and regular languages having $O(\log n)$ Mod_p counting communication complexity are exactly those languages in $Mod_p Pol(\mathcal{C}om)$. Furthermore, it was shown that any regular language outside of $UPol(\mathcal{C}om)$ has linear deterministic complexity and any regular language outside of $Mod_p Pol(\mathcal{C}om)$ has linear Mod_p counting complexity. So with respect to regular languages, $UP^{cc} = P^{cc} = UPol(\mathcal{C}om)$ and $Mod_p P^{cc} = Mod_p Pol(\mathcal{C}om)$. Similarly we conjecture that with respect to regular languages $NP^{cc} = Pol(\mathcal{C}om)$ and that other regular languages have linear non-deterministic complexity.

Conjecture 5.4. *If $L \subseteq \Sigma^*$ is a regular language that is not in $Pol(\mathcal{C}om)$, then $N^1(L) = \Omega(n)$. Thus we have*

$$N^1(L) = \begin{cases} O(1) & \text{if and only if } L \in \mathcal{C}om; \\ \Theta(\log n) & \text{if and only if } L \in Pol(\mathcal{C}om) \text{ but not in } \mathcal{C}om; \\ \Theta(n) & \text{otherwise.} \end{cases}$$

In general, the gap between deterministic and non-deterministic communication complexity of a function can be exponentially large. However, it has been shown that the deterministic communication complexity of a function f is bounded above by the product $cN^0(f)N^1(f)$ for a constant c and that this bound is optimal [7]. The above conjecture, together with the result of [14] imply the following much tighter relation for regular languages.

Conjecture 5.5 (Corollary to Conjecture 5.4). *If L is a regular language then $D(L) = \max\{N^1(L), N^0(L)\}$.*

For any variety $\mathcal{V}$, we have that $Pol(\mathcal{V}) \cap co\text{-}Pol(\mathcal{V}) = UPol(\mathcal{V})$ [11]. This implies that $N^1(L) = O(\log n)$ and $N^0(L) = O(\log n)$ iff $D(L) = O(\log n)$, proving a special case of the above corollary.

Conjecture 5.4 suggests that when faced with a non-deterministic communication problem for regular languages, the players have three options. They can either follow a trivial protocol that does not exploit the power of non-determinism or apply non-determinism in the most natural way as for the complement of the functions Disjointness and Equality. Otherwise the best protocol up to a constant factor is for one of the players to send all of his/her bits to the other player, a protocol that works for any function in any model. So with respect to regular languages, there is no "tricky" way to apply non-determinism to obtain cleverly efficient protocols.

5.2 Regular Languages with Linear Complexity

To prove a linear lower bound for the regular languages outside of $Pol(\mathcal{C}om)$, we need a convenient algebraic description for the syntactic monoids of these languages since in most cases lower bound arguments rely on these algebraic properties. So an important question that arises in this context is: What does it mean to be outside of $Pol(\mathcal{C}om)$? An algebraic description exists based on a result of [12] that describes the ordered monoid variety corresponding to $Pol(\mathcal{C}om)$.

Lemma 5.6. *Suppose L is not in $Pol(\mathcal{C}om)$ and M is the syntactic ordered monoid of L with exponent ω. Then there exists $u, v \in M^*$ such that*

(i) *for any monoid $M' \in$ **Com** and any morphism $\phi : M \to M'$, we have $\phi(eval(u)) = \phi(eval(v))$ and $\phi(eval(u)) = \phi(eval(u^2))$,*

(ii) *$eval(u^\omega v u^\omega) \not\leq eval(u^\omega)$.*

We now present the linear lower bound results. The proofs of the next two lemmas can be adapted from [14] to the non-deterministic case using the following simple fact.

Proposition 5.7. *Any stable order defined on a group G must be the trivial order (equality).*

Proof. Let $a, b \in G$ such that $a \leq b$. This means $1 \leq a^{-1}b =: g$. Since $1 \leq g$, we have $1 \leq g \leq g^2 \leq \ldots \leq g^k = 1$ for some k. This implies $1 = g$, i.e. $a = b$. □

Lemma 5.8. *If M is a non-commutative group then $N^1(M) = \Omega(n)$.*

We say that M is a T_q monoid if there exists idempotents $e, f \in M$ such that $(ef)^q e = e$ but $(ef)^r e \neq e$ when q does not divide r.

Lemma 5.9. *If M is a T_q monoid for $q > 1$ then $N^1(M) = \Omega(n)$.*

The next lemma captures regular languages that come close to the description of Lemma 5.6. A word w is a *shuffle* of n words $w_1, \dots, w_n$ if

$$w = w_{1,1}w_{2,1}\cdots w_{n,1}w_{1,2}w_{2,2}\cdots w_{n,2}\cdots\cdots w_{1,k}w_{2,k}\cdots w_{n,k}$$

with $k \geq 0$ and $w_{i,1}w_{i,2}\cdots w_{i,k} = w_i$ is a partition of w_i into subwords for $1 \leq i \leq n$.

Lemma 5.10. *If M and $u, v \in M^*$ are such that* (i) *$u = w_1w_2$ for $w_1, w_2 \in M^*$,* (ii) *v is a shuffle of w_1 and w_2,* (iii) *$eval(u)$ is an idempotent, and* (iv) *$eval(uvu) \not\leq eval(u)$, then $N^1(M) = \Omega(n)$.*

Proof. We show that $PDISJ \leq_r (M, I)$ where $I = \langle eval(u)\rangle$. Since v is a shuffle of w_1 and w_2, there exists $k \geq 0$ such that $v = w_{1,1}w_{2,1}w_{1,2}w_{2,2}\cdots w_{1,k}w_{2,k}$. The reduction is essentially local and is given by the following matrix when $k = 3$. The transformation easily generalizes to any k.

w_1	ϵ	ϵ	ϵ	ϵ	$w_{2,1}$	ϵ	$w_{2,2}$	ϵ	$w_{2,3}$
$w_{1,1}$	$w_{2,1}$	$w_{1,2}$	$w_{2,2}$	$w_{1,3}$	$w_{2,3}$	ϵ	ϵ	ϵ	ϵ

x_i	y_i	corresponding word
0	0	$w_1w_{2,1}w_{2,2}w_{2,3} = u$
0	1	$w_1w_{2,1}w_{2,2}w_{2,3} = u$
1	0	$w_{1,1}w_{1,2}w_{1,3}w_{2,1}w_{2,2}w_{2,3} = u$
1	1	$w_{1,1}w_{2,1}w_{1,2}w_{2,2}w_{1,3}w_{2,3} = v$

After x and y have been transformed into words, Alice prepends her word with u and appends it with $|u|$ many ϵ's, where $|u|$ denotes the length of the word u. Bob prepends his word with $|u|$ many ϵ's and appends it with u. Let $a(x)$ be the word Alice has and let $b(y)$ be the word Bob has after these transformations. If $PDISJ(x, y) = 0$, there exists i such that $x_i = y_i = 1$. By the transformation, this means $a(x)_1b(x)_1a(x)_2b(x)_2\cdots a(x)_sb(x)_s$ is of the form $u\cdots uvu\cdots u$ and since $eval(u)$ is idempotent,

$$eval(a(x)_1b(x)_1a(x)_2b(x)_2\cdots a(x)_sb(x)_s) = eval(uvu) \not\leq eval(u).$$

If $PDISJ(x, y) = 1$, then by the transformation, $a(x)_1b(x)_1\cdots a(x)_sb(x)_s$ is of the form $u...u$ and so $eval(a(x)_1b(x)_1a(x)_2b(x)_2\cdots a(x)_sb(x)_s) = eval(u)$. Thus $PDISJ \leq_r (M, \langle eval(u)\rangle)$. ☐

The conditions of this lemma imply the conditions of Lemma 5.6: since $eval(u)$ is idempotent, for any monoid $M' \in$ **Com** and any morphism $\phi : M \to M'$, we have $\phi(eval(u)) = \phi(eval(u^2))$ and since v is a shuffle of w_1 and w_2 we have $\phi(eval(u)) = \phi(eval(v))$. Also, since $eval(u)$ is idempotent, $eval(u^\omega) = eval(u)$, and in this case $eval(uvu) \not\leq eval(u)$ is equivalent to $eval(u^\omega vu^\omega) \not\leq eval(u^\omega)$.

Lemma 5.10 gives us a corollary about the monoid BA_2^+ which is defined to be the syntactic ordered monoid of $(ab)^* \cup a(ba)^*$. The syntactic ordered monoid of the complement of this language is BA_2^-. The unordered syntactic monoid is denoted by BA_2 and is known as the Brandt monoid.

Corollary 5.11. $N^1(BA_2^+) = \Omega(n)$.

Proof. It is easy to verify that BA_2^+ is the monoid $\{a, b\}^*$ with the relations $aa = bb, aab = aa, baa = aa, aaa = a, aba = a, bab = b$. All we need to know about the order relation is that $eval(aa)$ is greater than any other element. This can be derived from the definition of the syntactic ordered monoid since for any w_1 and w_2, w_1aaw_2 is not in L. So $w_1aaw_2 \in L \implies w_1xw_2 \in L$ trivially holds for any word x. Let $u = ab$ and $v = ba$. These u and v satisfy the four conditions of Lemma 5.10. The last condition is satisfied because $eval(uvu) = eval(abbaab) = eval(aa)$ and $eval(ab) \neq eval(aa)$. Thus $N^1(BA_2^+) = \Omega(n)$. □

Denote by U^- the syntactic ordered monoid of the regular language $(a \cup b)^*aa(a \cup b)^*$. The syntactic ordered monoid of the complement of this language is U^+. The unordered syntactic monoid is denoted by U. Observe that $N^1(U^-) = O(\log n)$ since all we need to do is check if there are two consecutive a's. One also easily sees that $N^1(BA_2^-) = O(\log n)$. By an argument similar to the one for Corollary 5.11, one can show that $N^1(U^+) = \Omega(n)$.

Combining the linear lower bound results we can conclude the following.

Theorem 5.12. *If M is a T_q monoid for $q > 1$ or is divided by one of BA_2^+, U^+ or a non-commutative group, then $N^1(M) = \Omega(n)$.*

We underline the relevance of the above result by stating a theorem which we borrow from [14].

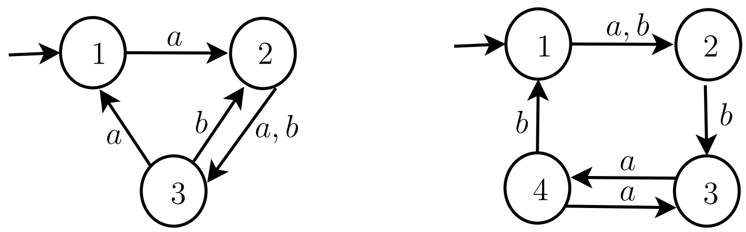
Theorem 5.13 (implicit in [14])**.** *If M is such that $D(M) \neq O(\log n)$ then M is either a T_q monoid for some $q > 1$ or is divided by one of BA_2, U or a non-commutative group.*

As a consequence, we know that if an ordered monoid M is such that $N^1(M) \neq O(\log n)$ then M is either a T_q monoid or is divided by one of BA_2^+, BA_2^-, U^+, U^- or a non-commutative group.

As a corollary to Theorem 5.12 and Lemma 5.3 we have:

Corollary 5.14. *If $M(L)$ is a T_q monoid or is divided by one of BA_2^+, U^+ or a non-commutative group, then L is not in $Pol(\mathcal{C}om)$.*

Consider the syntactic ordered monoid of the regular language recognized by the automaton in Fig. 1(a). One can show that it does not contain a non-commutative group, is not a T_q monoid and is not divided by BA_2^+ nor U^+. On the other hand, using Lemma 5.10 with $u = abbaa$ and $v = aabab$ we can show that it requires linear non-deterministic communication. Thus this lower bound is not achievable by previously known methods and highlights the importance of Lemma 5.10.



(a) $L_4 \in Pol(\mathcal{G}_{Sol})$. (b) $L_5 \in Pol(\mathcal{G}_{Nil,2})$.

Figure 1: Two examples. The missing arrows go to an accepting sink state.

5.3 Limitation of Current Techniques

The regular language L_5 accepted by the automaton in Fig. 1(b) is a concrete example of a language not in $Pol(\mathcal{C}om)$ and where all our techniques fail. In particular, it shows that the conditions in Lemma 5.10 do not cover every regular language outside of $Pol(\mathcal{C}om)$. We know that L_5 lies in $Pol(\mathcal{G}_{Nil,2})$ where $\mathcal{G}_{Nil,2}$ denotes the variety of languages whose syntactic monoid is a nilpotent group of class 2. These groups are "almost" commutative so L_5 in some sense comes close to being in $Pol(\mathcal{C}om)$.

For the deterministic and probabilistic models where $PDISJ$ is a hard function, one can observe that all the linear lower bounds obtained in [14] go through a local rectangular reduction from $PDISJ$ since $PDISJ$ reduces both to Disjointness and Inner Product. One might hope to obtain all the non-deterministic lower bounds in this manner as well. Given Lemma 5.6, one would want a local reduction of the form

x_i	y_i	corresponding word
0	0	u^ω
0	1	u^ω
1	0	u^ω
1	1	v

where u and v satisfy the conditions of Lemma 5.6.

Theorem 5.15. *There is no local reduction from $PDISJ$ to L_5 as described above.*

Proof. Since u^ω is an idempotent, it must induce a state transition function in which one of the following holds.

1. Every state is sent to the sink state.

2. One state, say state s, is sent to itself and every other state is sent to the sink state.

3. Two states are sent to themselves and every other state is sent to the sink state.

4. More than two states are sent to themselves.

We now show that none of the above can occur. Observe that it cannot be the case that u^ω is a partial identity on more than two states so case 4 is eliminated.

If u^ω satisfies condition 2, then we claim that we cannot have $eval(u^\omega v u^\omega) \nleq eval(u^\omega)$. To get a contradiction suppose this is true. Then there exists w_1, w_2 such that $w_1 u^\omega w_2 \in L$ and $w_1 u^\omega v u^\omega w_2 \notin L$. Since the latter is true, it must be the case that w_1 takes state 1 to s and w_2 must take s to a state other than the sink state. These w_1 and w_2 do not satisfy $w_1 u^\omega w_2 \in L$, so we get a contradiction. This shows we cannot have condition 2.

Similar to the above, one can show that u^ω cannot satisfy condition 1, which leaves us with condition 3. This means u^ω is either $(abab)^k$ or $(baba)^k$ for some $k > 0$. We assume it is $(abab)^k$. The argument for $(baba)^k$ is very similar.

Given $u^\omega = (abab)^k$, and the fact that we want to satisfy $eval(u^\omega v u^\omega) \nleq eval(u^\omega)$, one can show that the state transition function induced by v must be one of the following.

1. State 1 is sent to 3 and any other state is sent to the sink state.

2. State 3 is sent to 1 and any other state is sent to the sink state.

3. State 1 is sent to 3, 3 is sent to 1 and any other state is sent to the sink state.

Suppose v satisfies condition 1.

Case 1: $v = (ab)^{2t-1}$ for $t > 0$. Consider the matrix representation of the local reduction. In this matrix A, we count the parity of the a's in two ways and get a contradiction. First we count it by looking at the rows. The first row must produce the word $u^\omega = (abab)^k$ and the second row must produce the word $v = (ab)^{2t-1}$ so in total we have odd number of a's. Now we count the parity of a's by looking at $A_{1,1}A_{2,2}A_{1,3}A_{2,4}\ldots$ and $A_{2,1}A_{1,2}A_{2,3}A_{1,4}\ldots$. Both of these must produce the word $(abab)^k$ so in total we must have an even number of a's.

Case 2: $v = (ab)^{2t}bb\ldots$ for $t \geq 0$ (i.e. the prefix of v is of the form $(ab)^{2t}bb$). Let c be the column where we find the second b in the second row. Give value 1 to entries of A which are a and give value -1 to entries of b. Other entries (the ϵ's) get value 0. In terms of these values we have

$$\sum_{i=1}^{c} A_{2,i} = -2$$

and

$$\sum_{i=1}^{c} A_{1,i} \in \{0, 1\}.$$

Adding the two sums, we get a negative value. Now we count the same total in a different order. Assuming c is even we have

$$\sum_{i=1}^{c/2} A_{1,2i-1} + \sum_{i=1}^{c/2} A_{2,2i} \in \{0, 1\}$$

and

$$\sum_{i=1}^{c/2} A_{1,2i} + \sum_{i=1}^{c/2} A_{2,2i-1} \in \{0,1\}.$$

The total is positive. This is a contradiction.
Case 3: $v = (ab)^{2t-1}a(aa)^{2r}b\ldots$ for $t, r > 0$. Similar argument as above.

The same ideas show that v cannot satisfy neither conditions 2 nor 3. □

Acknowledgements

The author gratefully acknowledges Pascal Tesson and Denis Thérien for introducing him to the problem and for very insightful discussions. We also thank Jean-Eric Pin whose valuable input has been acquired through Pascal Tesson.

References

[1] A. Ada. Non-Deterministic Communication Complexity of Regular Languages. *ArXiv e-prints*, Jan. 2008.

[2] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory (preliminary version). In *FOCS '86: Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*, pages 337–347, 1986.

[3] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[4] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovasz–Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007.

[5] S. Eilenberg. *Automata, Languages, and Machines*. Academic Press, Inc., Orlando, FL, USA, 1974.

[6] V. Grolmusz. Separating the communication complexities of MOD m and MOD p circuits. In *IEEE Symposium on Foundations of Computer Science*, pages 278–287, 1992.

[7] B. Halstenberg and R. Reischuk. On different modes of communication. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 162–172, New York, NY, USA, 1988. ACM.

[8] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

[9] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[10] J.-E. Pin. A variety theorem without complementation. *Russian Mathematics (Izvestija vuzov.Matematika)*, 39:80–90, 1995.

[11] J.-E. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 1, chapter 10, pages 679–746. Springer, 1997.

[12] J.-E. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory Comput. Systems*, 30:383–422, 1997.

[13] J.-F. Raymond, P. Tesson, and D. Thérien. An algebraic approach to communication complexity. In *ICALP '98: Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, pages 29–40, London, UK, 1998. Springer-Verlag.

[14] P. Tesson and D. Thérien. Complete classifications for the communication complexity of regular languages. *Theory Comput. Syst.*, 38(2):135–159, 2005.

[15] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, Dec. 1991.

[16] A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC '79: Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 209–213, New York, NY, USA, 1979. ACM Press.