Electronic Colloquium on Computational Complexity, Report No. 76 (2000)

Measures of Nondeterminism in Finite Automata

*

Juraj Hromkovič[1] † Juhani Karhumäki[2] † Hartmut Klauck[3]
Georg Schnitger[3] Sebastian Seibert[1] †

[1]Lehrstuhl für Informatik I, RWTH Aachen,
Ahornstraße 55, 52074 Aachen, Germany
[2]Department of Mathematics, University of Turku, Finland
[3]FB Informatik, Johann-Wolfgang-Goethe-Universität,
60054 Frankfurt am Main, Germany

*An extended abstract of this paper has been presented at ICALP 2000.

†Supported by DFG grant Hr-1413-2 and the project "Descriptional Complexity and Efficient Transformations of Formal Languages over Words, Trees, and Graphs" (common grant 864524 of DAAD and of the Academy of Finland).

ISSN 1433-8092

Abstract

While deterministic finite automata seem to be well understood, surprisingly many important problems concerning nondeterministic finite automata (nfa's) remain open.

One such problem area is the study of different measures of nondeterminism in finite automata and the estimation of the sizes of minimal nondeterministic finite automata. In this paper the concept of communication complexity is applied in order to achieve progress in this problem area. The main results are as follows:

1. Deterministic communication complexity provides lower bounds on the size of unambiguous nfa's. Applying this fact, the proofs of several results about nfa's with limited ambiguity can be simplified.
2. For an nfa A we consider the complexity measures $advice_A(n)$ as the number of advice bits, $ambig_A(n)$ as the number of accepting computations, and $leaf_A(n)$ as the number of computations for worst case inputs of size n. These measures are correlated as follows (assuming that the nfa A is minimal):
$advice_A(n), ambig_A(n) \leq leaf_A(n) \leq O(advice_A(n) \cdot ambig_A(n))$.
3. $leaf_A(n)$ is always either a constant, between linear and polynomial in n, or exponential in n.
4. There is a family of languages KON_{k^2} with an exponential size gap between nfa's with polynomial leaf number/ambiguity and nfa's with ambiguity k. This partially provides an answer to the open problem posed by Ravikumar and Ibarra [SIAM J. Comput. 18 (1989), 1263–1282], and Hing Leung [SIAM J. Comput. 27 (1998), 1073–1082].

Keywords: finite automata, nondeterminism, limited ambiguity, descriptional complexity, communication complexity.

1 Introduction

In this paper the classical models of one-way finite automata (dfa's) and their nondeterministic counterparts (nfa's) [RS59] are investigated. While the structure and fundamental properties of dfa's are well understood, this is not the case for nfa's. For instance, we have efficient algorithms for constructing minimal dfa's, but the complexity of approximating the size of a minimal nfa is still unresolved (whereas finding a minimal nfa solves a PSPACE complete problem). Hromkovič, Seibert and Wilke [HSW97] proved that the gap between the length of regular expressions and the number of edges of corresponding nfa's is between $n \log^2 n$ and $n \log n$, but the exact relation is unknown. Another principal open question is to determine whether there is an exponential gap between two-way deterministic finite automata and two-way nondeterministic ones. The last partially successful attack on this problem was done in the late seventies by Sipser [S80], who established an exponential gap between determinism and nondeterminism for so-called sweeping automata (the property of sweeping is essential [M80]).
Our main goal is to contribute to a better understanding of the power of nondeterminism in finite automata (see [RS59], [MF71], [Mo71], [Sc78] for very early papers on this topic). We focus on the following problems:

1. The best known method for proving lower bounds on the size of minimal nfa's is based on nondeterministic communication complexity [Hr97]. All other known methods are special cases of this method. Are there methods that provide better lower bounds at least for some languages? How can one prove lower bounds on the size of unambiguous nfa's (unfa's), that is nfa's which have at most one accepting computation for every word?

2. It is a well known fact [MF71], [Mo71] that there is an exponential gap between the sizes of minimal dfa's and nfa's for some regular languages. This is even known for dfa's and unfa's [Sc78], [SH85], [RI89], for unfa's and nfa's with constant ambiguity [Sc78], [RI89], and for ufa's with polynomial ambiguity and nfa's [HL98] (We apologize that we made the mistake in the extended abstract of this paper [HKK00], where we state also the above results as our contribution instead of referring to [Sc78], [SH85], [RI89], [HL98]). But, it is open [RI89], [HL98] whether there exists an exponential gap between the sizes of minimal nfa's with constant ambiguity and nfa's with polynomial ambiguity.

3. The degree of nondeterminism is measured in the literature in three different ways. Let A be an nfa. The first measure $advice_A(n)$ equals the number of advice bits for inputs of length n, i.e., the maximum number of nondeterministic guesses in computations for inputs of length n. The second measure $leaf_A(n)$ determines the maximum number of computations for inputs of length n. $ambig_A(n)$ as the third measure equals the maximum number of accepting computations for inputs of length at most n. Obviously the second and third measure may be exponential in the first one. The question is whether the measures are more specifically correlated.

To attack these problems we establish some new bridges between automata theory and communication complexity. The communication complexity of two-party protocols was introduced by Yao [Y79] (and implicitly considered by Abelson [Ab78], too). The initial goal was to develop a method for proving lower bounds on the complexity of distributive and parallel computations (see, for instance, [Th79, Th80, Hr97, KN97]). Due to the well developed, nontrivial mathematical machinery for determining the communication complexity of concrete problems (see, for instance [AUY83, DHS96, Hr97, Hr00, KN97, L90, NW95, PS82]), communication complexity has established itself as a sub-area of complexity theory. The main contributions of the study of communication complexity lie especially in proving lower bounds on the complexity of specific problems, and in comparing the power of different modes of computation.
Here, for the first time, communication complexity is applied for the study of nondeterministic finite automata, with the emphasis on the tradeoff between the size and the degree of nondeterminism of nfa's. Our procedure is mainly based on the following facts:

(i) The theory of communication complexity contains deep results about the nature of nondeterminism (see, e.g. [KNSW94, HS96]) that use the combinatorial structure of the communication matrix as the computing problem representation.

(ii) In [DHRS97, Hr97], the non-uniform model of communication protocols for computing finite functions was extended to a uniform model for recognizing languages in such a way that several results about communication complexity can be successfully applied for uniform computing models like automata.

Combining (i) and (ii) with building of new bridges between communication complexity and nfa's we establish the following main results.

1. Let $cc(L)$ resp. $ncc(L)$ denote the deterministic resp. nondeterministic communication complexity of L. It is well known that $2^{cc(L)}$ and $2^{ncc(L)}$ are lower bounds on the sizes of the minimal dfa for L and a minimal nfa for L respectively. First we show that there are regular languages L for which there is an exponential gap between $2^{ncc(L)}$ and the minimal size of nfa's for L. This means, that the lower bound method based on communication complexity may be very weak. Then we show as a somewhat surprising result that $2^{\sqrt{cc(L)/k}} - 2$ is a lower bound on the size of nfa's with ambiguity k for L. We furthermore show that $Rank(M)^{1/k} - 1$ is a lower bound for the number of states for nfa's with ambiguity k, where M is a communication matrix associated with L. It is possible that this lower bound is always better than the first one (see [KN97] for a discussion of the quality of the so-called rank lower bound on communication complexity).

   As a corollary we present a sequence of regular languages NID_m such that the size of a minimal nfa is linear in m, while the size of every unfa for NID_m is exponential in m. This substantially simplifies the proofs of similar in [Sc78], [SH85].

2. We establish the relation

$$advice_A(n), ambig(n)_A \leq leaf_A(n) \leq O(advice_A(n) \cdot ambig_A(n))$$

   for any minimal nfa A. Observe that the upper bound on $leaf_A(n)$ implies that minimal unambiguous nfa's may have at most $O(advice_A(n)) \subseteq O(n)$ different computations on any input of size n, and an exponential gap between $advice_A(n)$ and $leaf_A(n)$ is possible only if the degree of ambiguousity is exponential in n.

   Furthermore we show that $leaf_A(n)$ is always either bounded by a constant, at least linear but polynomially bounded, or at least exponential in the input length.

3. We present another sequence of regular languages than in [HL98] with an exponential gap between the size of nfa's with exponential ambiguity, and nfa's with polynomial ambiguity. This result is obtained by showing that small nfa's with polynomial ambiguity for the Kleene closure $(L\#)^*$ imply small unfa's that work correctly on a polynomial fraction of inputs. Our technique is more general than proof method of Hing Leung [HL98] and provides an essentially shorter proof.

   Furthermore we describe a sequence of languages KON_{k^2} such that there is an exponential gap between the size of nfa's with polynomial ambiguity and nfa's with ambiguity k. This provides a partial answer to the open question [RI89], [HL98] whether there is an exponential gap between minimal nfa's with constant ambiguity and minimal nfa's with polynomial ambiguity. Our language KON_k is a candidate for proving a gap even separating between the size of nfa's with polynomial ambiguity and nfa's with constant ambiguity.

This paper is organized as follows. In section 2 we give the basic definitions and fix the notation. In order to increase the readability of this paper for readers who are not familiar with communication complexity theory, we give more details about communication protocols and build the basic intuition of their relation to finite automata. Section 3 is devoted to the investigation of the relation between the size of nfa's and communication complexity. Section 4 studies the relation between different measures of nondeterminism in finite automata, and presents the remaining results.

2 Definitions and Preliminaries

We consider the standard one-way models of finite automata (dfa's) and nondeterministic finite automata (nfa's). For every automaton A, $L(A)$ denotes the language accepted by A. The number of states of A is called the size of A and denoted $size_A$. For every regular language L we denote the size of the minimal dfa for L by $s(L)$ and the size of minimal nfa's accepting L by $ns(L)$.

For any nfa A and any input x we use the computation tree $T_{A,x}$ to represent all computations of A on x. Obviously the number of leaves of $T_{A,x}$ is the number of different computations of A on x.

The ambiguity of an nfa A on input x is the number of accepting computations of A on x, i.e., the number of accepting leaves of $T_{A,x}$. If the nfa A has ambiguity one for all inputs, then A is called an unambiguous nfa (unfa) and $uns(L)$ denotes the size of a minimal unfa accepting L. More generally, if an nfa A has ambiguity at most k for all inputs, then A is called a k-ambiguous nfa and $ns_k(L)$ denotes the size of a minimal k-ambiguous nfa accepting L.

For every nfa A we measure the degree of nondeterminism as follows. Let Σ denote the alphabet of A. For every input $x \in \Sigma^*$ and for every computation C of A on x we define $advice(C)$ as the number of nondeterministic choices during the computation C, i.e., the number of nodes on the path of C in $T_{A,x}$ which have more than one successor. Then

$$advice_A(x) = \max\{advice(C) \mid C \text{ is a computation of } A \text{ on } x\}$$

$$\text{and } advice_A(n) = \max\{advice(x) \mid x \in \Sigma^n\}.$$

For every $x \in \Sigma^*$ we define $leaf_A(x)$ as the number of leaves of $T_{A,x}$ and set

$$leaf_A(n) = \max\{leaf(x) \mid x \in \Sigma^n\}.$$

For every $x \in \Sigma^*$ we define $ambig_A(x)$ as the number of accepting leaves of $T_{A,x}$ and set

$$ambig_A(n) = \max\{ambig(x) \mid x \in \Sigma^{\leq n}\}.$$

Since a language need not contain words of all lengths we define ambiguity over all words of length at most n which makes the measure monotone. Observe that the leaf and advice measures are monotone as well.

Note that different definitions have been used by other authors; see e.g. [GKW90], [GLW92], where the number of advice bits is maximized over all inputs and minimized over all accepting computations on those inputs. In this case there are nfa's which use more than constant but less than linear (in the input length) advice bits, but this behavior is not known to be possible for minimal nfa's.

To prove lower bounds on the size of finite automata we shall use two-party communication complexity. This widely studied measure was introduced by Yao [Y79] and is the subject of two monographs [Hr97], [KN97].

First, we introduce the standard, non-uniform model of (communication) protocols for computing finite functions. A **(two-party communication) protocol** P consists of two computers C_I and C_{II} of unbounded computational power (sometimes called Alice and Bob in the literature) and a communication link between them. P computes a finite function $f : U \times V \to Z$ in the following way. At the beginning C_I gets an input $\alpha \in U$ and C_{II} obtains an input $\beta \in V$. Then C_I and C_{II} communicate according to the rules of the protocol by exchanging binary messages until one of them knows $f(\alpha, \beta)$. C_I and C_{II} may be viewed as functions in this communication, where the arguments of C_I (C_{II}) are its input α (β) and the whole previous communication history (the sequence $c_1, c_2, \ldots, c_k$ of all messages exchanged between C_I and C_{II} up until now), and the output is the new message submitted. We also assume that C_I (C_{II}) completely knows the behavior of C_{II} (C_I) in all situations (for all arguments). Another important assumption is that every protocol has the **prefix-freeness** property. This means, that for any $\delta, \gamma \in U$ $[V]$, and any communication history $c_1, c_2, \ldots, c_k$, the message $C_I(\delta, (c_1, c_2, \ldots, c_k))$ is no proper prefix of $C_I(\gamma, (c_1, c_2, \ldots, c_k))$ [the message $C_{II}(\delta, (c_1, c_2, \ldots, c_k))$ is no proper prefix of $C_{II}(\gamma, (c_1, c_2, \ldots, c_k))$]. Informally, this means that the messages are self-delimiting and we do not need any special symbol marking the end of the message.

Formally, the computation of a protocol (C_I, C_{II}) on an input is a sequence $c_1, c_2, \ldots, c_m, \gamma$, where $c_i \in \{0,1\}^+$ for $i = 1, \ldots, m$ are the messages and $\gamma \in Z$ is the result of the computation. The **communication complexity of the computation of P on an input (α, β)** is the sum of the lengths of all messages exchanged in the communication. The **communication complexity of the protocol P**, $cc(P)$, is the maximum of the communication complexities over all inputs from $U \times V$.

Due to the prefix-freeness property of messages we have that if, for two computations $c_1, c_2, \ldots, c_l, \gamma$ and $d_1, d_2, \ldots, d_r, \delta$, $c_1 c_2 \ldots c_l = d_1 d_2, \ldots d_r$, then $l = r$, and $c_i = d_i$ for $i = 1 \ldots, l$. So, if $Z = \{0,1\}$ and a protocol allows m different computations, then its communication complexity must be at least $\lceil \log_2 m \rceil - 1$.

The **communication complexity of f, $cc(f)$**, is the communication complexity of the best protocol for f, i.e.,

$$\boldsymbol{cc(f)} = min\{cc(P) \mid P \text{ computes } f\}.$$

The protocols whose computations consist of one message only (i.e. C_I sends a message to C_{II} and then C_{II} must compute the result) are called **one-way protocols**. For every finite function f,

$$\boldsymbol{cc_1(f)} = min\{cc(P) \mid P \text{ is a one-way protocol computing } f\}$$

is the **one-way communication complexity of f**.

The representation of a finite function $f : U \times V \to \{0,1\}$ by the so called communication matrix is very helpful for investigating the communication complexity of f. The **communication matrix of f** is the $|U| \times |V|$ Boolean matrix $M_f[u,v]$ defined by

$$M_f[u,v] = f(u,v)$$

for all $u \in U$ and $v \in V$. So, $M_f[u,v]$ consists of $|U|$ rows and $|V|$ columns. If one wants to fix this representation (which is not necessary for the relation to the communication complexity of f), one can consider some kind of lexicographical order for elements in U and V. But, the special order of rows and columns does not matter for our applications. Figure 1 presents the communication matrix M_f for the Boolean function $f : \{0,1\}^3 \times \{0,1\}^3 \to \{0,1\}$ defined by

$$f((x_1, x_2, x_3)(y_1, y_2, y_3)) = x_1 \oplus x_2 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3,$$

where $\oplus$ is addition modulo 2.

Definition 1 *Let $U = \{\alpha_1, \ldots, \alpha_k\}$, $V = \{\beta_1, \ldots, \beta_m\}$ be two sets and let $f : U \times V \to \{0,1\}$. Let $M_f = [a_{\alpha\beta}]_{\alpha \in U, \beta \in V}$. For every $\alpha \in U$, the row of α in M_f is*

$$row_\alpha = (a_{\alpha\beta_1}, a_{\alpha\beta_2}, \ldots, a_{\alpha\beta_m}).$$

$$\begin{array}{c|cccccccc|}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\hline
000 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
001 & \boxed{1} & 0 & 0 & \boxed{1} & 0 & \boxed{1} & \boxed{1} & 0 \\
010 & \boxed{1} & 0 & 0 & \boxed{1} & 0 & \boxed{1} & \boxed{1} & 0 \\
011 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
100 & \boxed{1} & 0 & 0 & \boxed{1} & 0 & \boxed{1} & \boxed{1} & 0 \\
101 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
110 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
111 & \boxed{1} & 0 & 0 & \boxed{1} & 0 & \boxed{1} & \boxed{1} & 0 \\
 & & & & & & & & \\
\hline
\end{array}$$

Figure 1:

For every $\beta \in V$, the column of β in M_f is

$$column_\beta = (a_{\alpha_1\beta}, a_{\alpha_2\beta}, \ \dots \ , a_{\alpha_k\beta})^T.$$

***Row*$(\boldsymbol{M_f})$** *is the number of different rows of M_f.*
A **submatrix** *of M_f is any intersection of a non-empty set of rows with a non-empty set of columns. A* **$\boldsymbol{\delta}$-monochromatic submatrix***, $\delta \in \{0,1\}$ of M_f is any submatrix of M_f whose elements are all equal to δ (Figure 1 depicts the 1-monochromatic submatrix that is the intersections of rows 001, 010, 100 and 111 with the columns 000, 011, 101 and 110).*
Let $S = \{M_1, M_2, \ \dots \ , M_k\}$ be a set of monochromatic submatrices of a Boolean matrix M_f. We say that S is a **cover of $\boldsymbol{M_f}$** *if, for every element $a_{\alpha\beta}$ of M_f, there exists an $m \in \{1, \ \dots \ , k\}$ such that $a_{\alpha\beta}$ is an element of M_m. We say that S is an* **exact cover of $\boldsymbol{M_f}$** *if S is a cover of M_f and $M_r \cap M_s = \emptyset$ for every $r \neq s$, $r, s \in \{1, \ \dots \ , k\}$. The* **tiling complexity of $\boldsymbol{M_f}$** *is*

$$\mathbf{Tiling}(\mathbf{M_p}) = min\{|S| \mid S \textit{ is an exact cover of } M_f\}$$

□

The work of a protocol $(C_I,\ C_{II})$ for f can be viewed as a game on the communication matrix M_f. C_I with input α knows the row row_α, C_{II} with input β knows the column $column_\beta$, and they have to determine $f(\alpha, \beta)$. [1] A communication message c_1 submitted from C_I to C_{II} can be viewed as the reduction of M_f to a submatrix $M_f(c_1)$ consisting of rows for which C_I sends c_1 because C_{II} knows the behavior of C_I. Similarly the second message c_2 sent from C_{II} to C_I restricts $M_f(c_1)$ to $M_f(c_1, c_2)$ which consists of the columns of $M_f(c_1)$ for which C_{II} with the second argument c_1 sends c_2. Whenever row_α ($column_\beta$) of $M_f(c_1, c_2, \ \dots \ , c_k)$ is monochromatic, C_I (C_{II}) knows the result. So, every computation of $(C_I,\ C_{II})$ that finishes with 1 (0) defines a 1-monochromatic (0-monochromatic) submatrix of M_f. This means that all inputs

[1] Note that they do not need to estimate the coordinates of the intersection of row_α and $column_\beta$.

(δ, μ) contained in this monochromatic submatrix have the same computation of the protocol C_I and C_{II}. So, $(C_I,\ C_{II})$ unambiguously determine an exact cover of M_f by monochromatic submatrices. More precisely, a protocol with k different computations determines an exact cover of cardinality k. The immediate consequence is:

Fact 1 *For every finite function $f : U \times V \to \{0, 1\}$,*

$$cc(f) \geq \lceil \log_2(Tiling(M_f)) \rceil.$$

Another important consequence is the following fact.

Fact 2 *For every finite function $f : U \times V \to \{0, 1\}$,*

$$cc_1(f) = \lceil \log_2(Row(M_f)) \rceil.$$

PROOF: For no two different rows row_α and row_δ, a one-way protocol computing f can send the same message c because C_{II} cannot determine the result for any μ such that $column_\mu$ has different values on the intersections with row_α and row_δ. On the other hand, $Row(M_f)$ different messages are enough (one message for a group of identical rows) to construct a one-way protocol for f. □

Since the number of 1-monochromatic matrices in any exact cover of all ones in M_f is a trivial upper bound on the rank of M_f, Fact 1 implies:

Fact 3 *For every finite function $f : U \times V \to \{0, 1\}$, and every field F with neutral elements 0 and 1,*

$$cc(f) \geq \lceil \log_2(Rank_F(M_f)) \rceil.$$

Let $\mathbb{Q}$ be the set of rational numbers. Since it is well-known that

$$Rank_{\mathbb{Q}}(M) = max\{Rank_F(M_f) \mid M \text{ is a field with neutral elements 0 and 1}\}$$

we formulate Fact 3 as

$$cc(f) \geq \lceil \log_2(Rank_Q(M_f)) \rceil$$

for every finite function f.

Now, we consider nondeterministic communication complexity and its relation to some combinatorial properties of M_f. A **nondeterministic protocol** P computing a finite function $f : U \times V \to \{0, 1\}$ consists of two nondeterministic computers C_I and C_{II} that have a nondeterministic choice from a finite number of messages for every input argument. For any input $(\alpha, \beta) \in U \times V$, we say that P **computes** 1 (or that P **accepts** (α, β)) if there exists a computation of P on (α, β) that ends with the result 1. So, P **computes** 0 for an input (α, β) (rejects (α, β)) if all computations of P on (α, β) end with the result 0. The **nondeterministic communication complexity of *P***, denoted ***ncc(P)***, is the maximum of the communication complexities of all accepting computations of P. The **nondeterministic communication complexity of *f*** is

$$\mathbf{ncc(f)} = min\{ncc(P) \mid P \text{ is a nondeterministic protocol computing } f\}$$

Let $\boldsymbol{ncc_1(f)}$ denote the **one-way nondeterministic communication complexity of $\boldsymbol{f}$**.

Similarly as in the deterministic case, every accepting computation of P for f unambiguously determines a 1-monochromatic submatrix of M_f and the union of all such 1-monochromatic submatrices must cover all the 1's of M_f but no 0 of M_f. The difference to the deterministic case is that these 1-monochromatic submatrices may overlap, which corresponds to the fact that P may have several different accepting computations on a given input.

Definition 2 *Let M_f be a Boolean matrix, and let $S = \{M_1, M_2, \ldots, M_k\}$ be a set of 1-monochromatic submatrices of M_f. We say that S is a 1-cover of M_f if every 1 of M_f is contained in at least one of the 1-submatrices of S. We define*

$$\mathbf{cover}(\mathbf{M_f}) = min\{|S| \mid S \textit{ is a 1-cover of } M\}.$$

□

Fact 4 *For every finite function $f : U \times V \to \{0,1\}$,*

$$ncc_1(f) = ncc(f) = \lceil \log_2(Cover(M_f)) \rceil.$$

PROOF: The above consideration showing that a nondeterministic protocol with m accepting computations determines a 1-cover of M_f of cardinality m implies

$$\lceil \log_2(Cover(M_f)) \rceil \leq ncc(f).$$

Since $ncc(f) \leq ncc_1(f)$ for every f, it is sufficient to prove $ncc_1(f) \leq \lceil \log_2(Cover(M_f)) \rceil$. Let $S = \{M_1, \ldots, M_m\}$ be a 1-cover of M_f. A one-way nondeterministic protocol (C_I, C_{II}) can work on an input (α, β) as follows. C_I with input α nondeterministically chooses one of the matrices of S with a non-empty intersection with row_α and sends the binary code of its index i to C_{II}. If $column_\beta$ has a non-empty intersection with M_i, then C_{II} accepts. Since $\lceil \log_2 m \rceil$ message length suffices to code m different messages, $ncc(C_I, C_{II}) = \lceil \log_2 m \rceil$. □

The first trivial bridge [Hr86] between automata and communication complexity says that

$$s(L) \geq 2^{cc_1(f_{2n,L})} \text{ and } ns(L) \geq 2^{ncc_1(f_{2n,L})} \tag{1}$$

for every regular language $L \subseteq \Sigma^*$ and every positive integer n, where $f_{2n,L} : \Sigma^n \times \Sigma^n \to \{0,1\}$, $f_{2n,L}(\alpha, \beta) = 1$ iff $\alpha\beta \in L$. The argument for this lower bound is very simple. Let A be a dfa (nfa) accepting L with $s(L)$ ($ns(L)$) states. Then a one-way protocol can compute $f_{2n,L}$ as follows. For an input α, C_I simulates the work of A on α and sends the name of the state q reached by A after reading α to C_{II}. C_{II} continues in the simulation of the suffix β from the state q. If A accepts $\alpha\beta$, then (C_I, C_{II}) accepts (α, β).

Unfortunately, the lower bound (1) may be arbitrarily bad for both $s(L)$ and $ns(L)$ because this non-uniform approach cannot completely capture the complexity of the uniform acceptance of L. We shall overcome this difficulty in the next section.

3 Communication Complexity and Finite Automata

To improve lower bounds on $s(L)$ and $ns(L)$ by communication complexity, Ďuriš, Hromkovič, Rolim, and Schnitger [DHRS97] (see also [Hr86]) introduced uniform protocols and communication matrices of regular languages as follows. For every regular language $L \subseteq \Sigma^*$, we define the infinite Boolean matrix $M_L = [a_{\alpha\beta}]_{\alpha\in\Sigma^*,\beta\in\Sigma^*}$, where

$$a_{\alpha\beta} = 1 \textit{ iff } \alpha\beta \in L.$$

Since every regular language has a finite index (Myhill-Nerode theorem), the number of different rows of M_L is finite. So, we can again use the protocols as finite devices for accepting L.

Definition 3 *Let Σ be an alphabet and let $L \subseteq \Sigma^*$. A* **one-way uniform protocol over Σ** *is a pair $(C_I,\ C_{II})$, where*

(i) $C_I :\ \Sigma^* \to \{0,1\}^+$ is a function with the prefix freeness property, and $\{C_I(\alpha) \mid \alpha \in \Sigma^*\}$ is a finite set, and

(ii) $C_{II} :\ \Sigma^* \times \{0,1\}^* \to \{accept, reject\}$ is a function.

We say that $D = (C_I,\ C_{II})$ **accepts** *L,* **$L(D) = L$**, *if, for all $\alpha, \beta \in \Sigma^*$:*

$$C_{II}(\beta, C_I(\alpha)) = \text{accept } \textit{iff } \alpha\beta \in L.$$

The **message complexity of the protocol D** *is*

$$\boldsymbol{mc(D)} = |\{C_I(\alpha) \mid \alpha \in \Sigma^*\}|$$

(i.e. the number of the messages used by D), and the **message complexity of L** *is*

$$\boldsymbol{mc(L)} = min\{mc(D) \mid D \textit{ is a one-way uniform protocol accepting } L\}.$$

The **communication complexity of D** *is*

$$\boldsymbol{cc(D)} = max\{|C_I(\alpha)| \mid \alpha \in \Sigma^*\},$$

and the **one-way communication complexity of L** *is*

$$\boldsymbol{cc_1(L)} = min\{cc(D) \mid D \textit{ is a one-way uniform protocol accepting } L\}.$$

□

If one wants to give a formal definition of a **one-way nondeterministic protocol** over Σ, it is sufficient to consider C_I as a function from Σ^* to a finite subset of $\{0,1\}^*$. The acceptance criterion of L changes to

$$(\exists c \in C_I(\alpha) \text{ such that } accept \in C_{II}(\beta, c)) \Leftrightarrow \alpha\beta \in L.$$

Let $nmc_1(L)$ $[ncc_1(L)]$ denote the **one-way nondeterministic message [communication] complexity of L**. We observe that the main difference between uniform protocols and (standard) protocols is the way the input is partitioned between C_I and C_{II}. If a protocol D computes a Boolean function $f : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}$, one can view this as the partition of inputs of f (from $\{0,1\}^{r+s}$) into the prefix of r bits and a suffix of s bits (i.e. assigning the first r bits to C_I and the rest to C_{II}), and a communication between C_I and C_{II} in order to compute the value of f. A uniform protocol over Σ considers, for every input $\alpha = \alpha_1\alpha_2\ldots\alpha_n \in \Sigma^n$, $n+1$ partitions of α $[(\lambda,\alpha),(\alpha_1,\alpha_2\ldots\alpha_n),(\alpha_1\alpha_2,\alpha_3\ldots\alpha_n), \ \ldots \ ,(\alpha_1\ldots\alpha_{n-1},\alpha_n),(\alpha,\lambda)]$ and for each of these partitions it must accept (reject) if $\alpha \in L$ ($\alpha \notin L$). This means, that the matrices $M_L = [a_{\alpha,\beta}]$ are special Boolean matrices with $a_{\lambda,\alpha_1\ldots\alpha_n} = a_{\alpha_1,\alpha_2\ldots\alpha_n} = \ \ldots \ = a_{\alpha_1\ldots\alpha_n,\lambda}$ and a uniform protocol D for L must recognize the membership of α to L for every partition of α between C_I and C_{II}.

The following result from [DHRS97] shows in fact that one-way uniform protocols are nothing else than deterministic finite automata.

Fact 5 *Let Σ be an alphabet. For every regular language $L \subseteq \Sigma^*$,*

$$s(L) = mc(L) = Row(M_L).$$

THE IDEA OF PROOF: $s(L) = Row(M_L)$ is just a reformulation of the Myhill-Nerode theorem. In Section 2 we have already observed that $Row(M_L)$ is exactly the number of different messages used by an optimal one-way protocol.[2] □

Following the idea of the simulation of a finite automaton by a protocol in the nondeterministic case, we have the following obvious fact [Hr97].

Fact 6 *For every alphabet Σ and every regular language $L \subseteq \Sigma^*$,*

$$nmc(L) \leq ns(L).$$

Fact 6 provides the best known lower bound proof technique on the size of minimal nfa's. All previously known techniques like the fooling set approach are special cases of this approach. Moreover the fooling set method, which covers all previous efforts in proving lower bounds on $ns(L)$, can (for some languages) provide exponentially smaller lower bounds than the method based on nondeterministic communication complexity [DHS96].

The first question is therefore whether $nmc(L)$ can be used to approximate $ns(L)$. Unfortunately this is not possible. Note that a result similar to Lemma 1 was also independently established by Jiráskova [Ji99].

Lemma 1 *There exists a sequence of regular languages $\{PART_n\}_{n=1}^{\infty}$ such that*

$$ns(PART_n) \geq 2^{\Omega(\sqrt{nmc(PART_n)})}.$$

[2] The fact that M_L is infinite does not matter because M_L has a finite number of different rows.

PROOF: Let $PART_n = \{xyz : |x| = |y| = |z| = n, \text{ and } x \neq z \vee x = y\}$. For the next considerations it is important to observe that the condition $x \neq z \vee x = y$ is equivalent to the condition $x \neq z \vee x = y = z$. First we describe a nondeterministic uniform protocol (C_I, C_{II}) for $PART_n$ which uses $O(n^2)$ messages.

Players C_I and C_{II} compute the lengths l_I, l_{II} of their inputs. C_I communicates l_I and C_{II} rejects when $l_I + l_{II} \neq 3n$. So we assume that $l_I + l_{II} = 3n$ in the following.

Case 1: $l_I \leq n$.

C_I chooses a position $1 \leq i \leq l_I$ and communicates i, x_i, l_I. C_{II} accepts, if $x_i \neq z_i$. Otherwise C_{II} accepts if and only if $y = z$.

Observe that if $x \neq z$, then there is an accepting computation because there exists i such that $x_i \neq z_i$. If however $x = z$, then C_{II} accepts iff $y = z$, that is iff $x = y$.

Case 2: $n < l_I \leq 2n$.

C_I chooses a position $1 \leq i \leq n$ and communicates i, x_i, l_I. Furthermore, C_I compares $x_1, \ldots, x_{l_I - n}$ with $y_1, \ldots, y_{l_I - n}$ and sends the bit 1, if the strings are equal and the bit 0 if the strings are different. C_{II} accepts if $x_i \neq z_i$. Otherwise (if $x_i = z_i$) C_{II} compares $y_{l_I - n+1}, \ldots, y_n$ with $z_{l_I - n+1}, \ldots, z_n$. If the two strings are equal and the bit 1 was received, then C_{II} accepts and rejects otherwise.

Note that if $x \neq z$ then there is an accepting computation. If not, then C_{II} accepts if and only if $x = y = z$.

Case 3: $2n < l_I \leq 3n$.

C_I chooses a position $l_I - 2n < i \leq n$ and communicates i, x_i, l_I. Furthermore C_I compares x with y. If $x = y$ or $x_j \neq z_j$ for $1 \leq j \leq l_I - 2n$, then C_I accepts. Otherwise C_{II} accepts if and only if $x_i \neq z_i$.

The protocol uses $O(n^2)$ messages, so $nmc(PART_n) = O(n^2)$.

Now, we prove that $ns(PART_N) \geq 2^{\frac{n}{2}}$. Obviously, every nfa B accepting $PART_n$ must have the following properties:

(i) $L_1 = \{xxx \mid x \in \{0,1\}^n\} \subseteq L(B)$, i.e. there is an accepting computation of B on every word xxx or $x \in \{0,1\}^n$, and

(ii) $L(B) \cap L_2 = \emptyset$ for $L_2 = \{xyx \mid x, y \in \{0,1\}^n, x \neq y\}$, i.e. there is no accepting computation of B on any word xyx with $x \neq y,\ x, y \in \{0,1\}^n$.

We prove that every nfa satisfying (i) and (ii) must have at least $2^{\frac{n}{2}}$ states. Let us assume the opposite. Let A be a nfa with fewer than $2^{\frac{n}{2}}$ states that satisfies (i) and (ii). Since $L_1 \subseteq L(B)$, there exists an accepting computation C_x on xxx for every $x \in \{0,1\}^n$. Let $Pattern(C_x) = (p, q)$, where p is the state of C_x after reading x and q is the state of C_x after reading xx. Since the number of states is smaller than $2^{\frac{n}{2}}$, the number of different patterns is smaller than $2^n = |\{0,1\}|^n$. So, there exist two words $u, v \in \{0,1\}^n,\ u \neq v$, such that $Pattern(C_u) = Pattern(C_v) = (r, s)$ for some states r, s. This means that starting to work from r on u as well as on v one can reach s after reading u or v. The immediate consequence is that there are accepting computations of B on uvu and vuv as well. Since $u \neq v$, uvu and vuv belong to L_2, a contradiction with condition (ii). □

To find lower bound methods for $ns(L)$ that provide results at most polynomially smaller than $ns(L)$ is one of the central open problems on finite automata. In the following, we concentrate on lower bounds for nfa's with constant ambiguity. Even for unambiguous automata no nontrivial general method for proving lower bounds has been known up to now.

To introduce our method for proving lower bounds on nfa's with bounded ambiguity we have to work with the communication matrices for regular languages. In Fact 5 we have observed that every matrix M_L has a finite number of different rows, which is the index $s(L)$ of the regular language L (this means that there exists a $s(L) \times s(L)$ (finite) submatrix M of M_L such that $Row(M) = Row(M_L)$, $Rank_F(M) = Rank_F(M_L)$ for every field F with neutral elements 0 and 1, $Tiling(M) = Tiling(M_L)$ and $Cover(M) = Cover(M_L)$). Thus, instead of introducing the general two-way uniform communication protocols), we define the **communication complexity of L**, denoted **$cc(L)$**, as the communication complexity of the best protocol for the communication matrix M_L. Because of the definition of M_L, this approach covers the requirement that the protocol correctly decides membership of any input to L for any prefix-suffix partition of the input.

Before formulating the main result of this section we build our intuition about the connection between $cc(L)$ and $uns(L)$. If one simulates an unambiguous automaton by a nondeterministic one-way protocol in the standard way described above, then the resulting protocol is unambiguous, too. This means that every one in M_L is covered by exactly one accepting computation, i.e. the unfa A determines an exact cover of all 1's in M_L of cardinality $size_A$. The similarity to the deterministic communication complexity is that any such protocol determines an exact cover of all elements of the communication matrix by monochromatic submatrices. Some nontrivial results from communication complexity theory [KNSW94] are needed to relate $cc(L)$ and $uns(L)$ via the outlined connection.

Theorem 1 *For every regular language $L \subseteq \Sigma^*$,*
a) $uns(L) \geq Rank_{\mathbb{Q}}(M_L)$
b) $ns_k(L) \geq Rank_{\mathbb{Q}}(M_L)^{1/k} - 1.$
c) $ns_k(L) \geq 2^{\sqrt{cc(L)})/k} - 2.$

PROOF: Let A be an optimal unfa for L. A can be simulated by a one-way nondeterministic protocol as follows: C_I simulates A on its input and communicates the obtained state. C_{II} continues the simulation and accepts/rejects accordingly. Obviously the number of messages is equal to $size_A$ and the protocol works with unambiguous nondeterminism.

It is easy to see that the messages of the protocol correspond to $size_A$ many submatrices of the matrix M_L covering all ones exactly once. Hence the rank is at most $size_A$ and we have shown a), which is the rank lower bound on communication complexity [MS82] (see Fact 3 in Section 2).

For b) observe that the above simulation induces a cover of the ones in M_L so that each one is covered at most k times. By the following fact from [KNSW94] we are done:

Fact 7 *Let $\kappa_r(M)$ denote the minimal size of a set of submatrices covering the ones of a Boolean matrix M so that each is covered at most r times. Then*

$$(1 + \kappa_r(M))^r \geq Rank(M).$$

For the other claim again simulate A by a one-way k-ambiguous nondeterministic protocol with $size_A$ messages.
Results of [KNSW94] (see also [L90], [Y91]) imply that a k-ambiguous nondeterministic one-way protocol with m messages can be simulated by a deterministic two-way protocol with communication $\log(m^k + 1) \cdot k \cdot \log(m + 2)$. Thus

$$cc(L) \leq \log(size_A^k + 1) \cdot k \cdot \log(size_A + 2) \leq \log^2((size_A + 2)^k)$$

and c) follow. □
Before giving an application of the lower bound method we point out that neither $2^{\sqrt{cc(L))}}$ nor $Rank_{\mathbb{Q}}(M_L)$ is a lower bound method capable of proving polynomially tight lower bounds on the minimal size of unfa's for all languages. In the first case this is trivial, in the second case it follows from a modification of a result separating rank from communication complexity (see [KN97]). But the gap between $Rank_{\mathbb{Q}}(M_L)$ and $uns(L)$ may be bounded by a pseudo-polynomial function.
Now we apply Theorem 1 in order to present an exponential gap between $ns(L)$ and $uns(L)$ for a specific regular language. Let, for every positive integer m, $NID_m = \{u \in \{0,1\}^* \mid \exists i : u_i \neq u_{i+m}\}$.

Theorem 2 *For every positive integer m*
(i) *NID_m can be recognized by an nfa A with ambiguity $O(m)$ and size $O(m)$*
(ii) *Any nfa with ambiguity k for NID_m has size at least $2^{m/k} - 1$, and in particular any unfa for NID_m must have $2^m - 1$ states.*
(iii) *No nfa with ambiguity $o(m/\log m)$ for NID_m has polynomial size in m.*

Proof:
(i) First the nfa guesses a residue i modulo m, and then checks whether there is a position $p \equiv i \bmod m$ with $u_p \neq u_{p+m}$.
(ii) Observe that the submatrix spanned by all words u and v with $u, v \in \{0,1\}^m$ is the "complement" of the $2^m \times 2^m$ identity matrix. The result now follows from assertions a) and b) of Theorem 1.
(iii) is an immediate consequence of (ii). □
We see that the proof of Theorem 2 is a substantial simplification of the proofs of similar results presented in [Sc78], [SH85].

4 Degrees of Nondeterminism in Finite Automata

It is easy to see that $advice_A(n) \leq leaf_A(n) \leq 2^{O(advice_A(n))}$ and also that $ambig_A(n) \leq leaf_A(n)$ for every nfa A. The aim of this section is to investigate whether stronger relations between these measures hold.

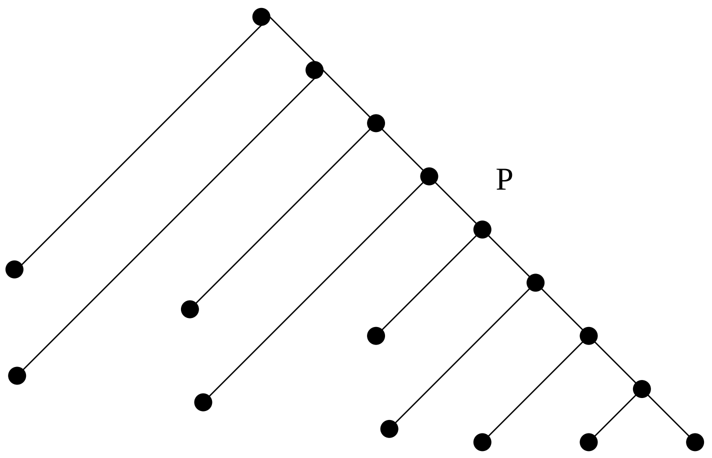

Figure 2:

Lemma 2 *For all nfa A either*
a) $advice_A(n) \leq size_A$ *and* $leaf_A(n) \leq size_A^{size_A}$ *or*
b) $advice_A(n) \geq n/size_A - 1$ *and* $leaf_A(n) \geq n/size_A - 1$.

PROOF: If some reachable state q of A belongs to a cycle in A and if q has two edges with the same label originating from it such that one of these edges belongs to the cycle, then $advice_A(n) \geq (n - size_A)/size_A \geq n/size_A - 1$. Otherwise for all words all states with a nondeterministic decision are traversed at most once. □

Our next lemma relates the *leaf* function to ambiguity. The initial idea is that a computation tree of any minimal unfa A on any input w could look like the tree from Figure 2. There is exactly one path P from the root to a leaf (a computation) with several nondeterministic guesses and all paths having only one vertex in common with P do not contain any nondeterministic branching. In other words, if a computation branches into two computations P_1 and P_2, then at least one of P_1 and P_2 should be completely deterministic. We are not able to verify this nice structure, but the next result shows that any computation tree of a minimal unfa A is very thin because every level of this tree can contain at most $size_A + 1$ different computations.

In what follows a state q of an nfa A is called *terminally rejecting*, if there is no word and no computation of A, such that A accepts when starting in q, i.e., $\delta^*(q, v)$ contains no accepting state for any word v. Clearly there is at most one terminally rejecting state in a minimal automaton, because otherwise these states can be joined reducing the size. Call all other states of A *undecided.*

Lemma 3 *Every nfa A with at most one terminally rejecting state satisfies*

$$leaf_A(x) \leq ambig_A(|x| + size_A) \cdot |x| \cdot size_A + 1$$

for all x.

PROOF: Let $k = ambig_A(|x| + size_A)$. If the computation tree consists only of nodes marked with the terminally rejecting state, then the tree has just one leaf and the claim is trivial. For the general case, consider a level of the computation tree of A on x that is not the root level. Assume that the level contains more that $k \cdot size_A$ nodes labeled with undecided states (called undecided nodes). Then one undecided state q must appear at least $k+1$ times on this level. There are $k+1$ computations of A on a prefix of x such that q is reached. If q is accepting, then the prefix of x is accepted with $k+1$ computations, a contradiction, since $ambig_A$ is monotone. If q is rejecting, but undecided, then there is a word v of length at most $size_A$ such that v is accepted by some computation of A starting in q. But then the prefix of x concatenated with v is accepted by at least $k+1$ computations, a contradiction.

Thus each level of the tree that is not the root level contains at most $k \cdot size_A$ undecided nodes. Overall there are at most $|x| \cdot k \cdot size_A + 1$ undecided nodes.

Observe that each node has at most one terminally rejecting child. Thus the number of terminally rejecting leaves is equal to the number of undecided nodes that have a terminally rejecting child. Hence the number of terminally rejecting leaves is at most the number of undecided nodes minus the number of undecided leaves. Thus the overall number of leaves is at most the number of terminally rejecting leaves plus the number of undecided leaves which is at most the number of undecided nodes. So overall there are at most $k \cdot |x| \cdot size_A + 1$ leaves. □

Theorem 3 *Every nfa A with at most one terminally rejecting state satisfies*

$$advice_A(n), ambig_A(n) \leq leaf_A(n) \leq O(ambig_A(n) \cdot advice_A(n)).$$

Especially for any such unfa: $advice_A(n) = \Theta(leaf_A(n))$.

PROOF: Observe that for all n: $ambig_A(n) = \Omega(ambig_A(n + O(1)))$, since $ambig_A$ is monotone and at most exponential. □

Next we further investigate the growth of the leaf function. Lemma 4 is a variation of a result in [IR86].

Lemma 4 *For every nfa A, either $leaf_A(n) \leq (n \cdot size_A)^{size_A}$ or $leaf_A(n) \geq 2^{\Omega(n)}$.*

PROOF: Assume that an nfa A contains some state q, such that q can be reentered on two different paths starting in q, where each path is labeled with the same word w. It is not hard to show that in this case there are two different paths from q to q labeled with a word w of length $size_A^2 - 1$. Then the computation tree of uw^m (where u leads from the starting state to q) has at least $2^m \geq 2^{(n-size_A)/size_A^2}$ leafs, where $n = |uw^m|$.

Now assume that A does not contain such a state. Then, for each nondeterministic state q (i.e., a state with more than one successor for the same letter) and any computation tree, the following holds: If q is the label of a vertex v, then q appears in each level of the subtree of v at most once.

We prove by induction over the number k ($k \leq size_A$) of different nondeterministic states in a computation tree that the number of leafs is at most $(n \cdot size_A)^k$. The claim is certainly true if there are no nondeterministic states.

Assume that there are k nondeterministic states, with some state q_1 appearing first in the tree. Observe that no level in the entire computation tree contains q_1 more than once.

For each occurrence of q_1 in the computation tree fix some child, so that the overall number of leaves is maximized. We get a tree with one nondeterministic state less, and by the induction hypothesis this tree has at most $(n \cdot size_A)^{k-1}$ leaves.

Since q_1 appears at most once on each level and since there are at most $size_A$ children of q_1 on each level, there are at most $(n \cdot size_A)^k$ leaves. □

Lemma 2 and 4 give us

Theorem 4 *For every nfa A: $leaf_A(n)$ is bounded by a constant, or is between linear and polynomial in n, or is $2^{\Theta(n)}$.*

Now, we consider the difference between polynomial and exponential ambiguity resp. polynomial and exponential leaf number. We show that languages which have small automata of polynomial ambiguity are related to the concatenation of languages having small unfa's. If the language is a Kleene closure, then one unfa accepts a large subset. Compare this to [GKW90], where Kleene closures are shown to be recognizable as efficient by nfa's with constant advice as by dfa's.

Theorem 5 a) *Let L be an infinite regular language and A some nfa for L with polynomial ambiguity. Then there are $d \leq size_A$ languages L_i such that $L_1 \cdots L_d \subseteq L$, L_i is recognizable by an unfa with $O(size_A)$ states, and*

$$\frac{|L_1 \cdots L_d \cap \Sigma^n|}{|L \cap \Sigma^n|} = \Omega(1)$$

for infinitely many n.

b) *Let $L = (K\#)^*$ for a regular language K not using the letter $\#$ and let A be some nfa for L with polynomial ambiguity. Then, for all m, there is an unfa A' with $O(size_A)$ states that decides $L' \subseteq L$ such that for infinitely many n*

$$\frac{|L' \cap (\Sigma^m \cap K)\#)^n|}{|((\Sigma^m \cap K)\#)^n|} = \Omega(1/poly(n)).$$

Proof: **a)** Define the ambiguity graph of A in the following way: the nodes are the (reachable) states of A and there is an edge from q_i to q_j if there are two paths from q_i to q_j in A, with the same label sequence. Note that the ambiguity graph is acyclic iff the ambiguity of A is polynomially bounded as we have seen in the proof of Lemma 4.

Now we construct a unfa $A_{i,j,k}$ which accepts those words that lead in A from q_i to q_j and then via one edge to q_k. Here, we assume that the longest path from q_i to q_k in the ambiguity graph consists of one edge and q_j is reachable from q_i in A, but not in the ambiguity graph. Moreover, we demand that there is an edge in A from q_j to q_k.

The states of $A_{i,j,k}$ are the states reachable in A from q_i, but not reachable in the ambiguity graph from q_i, plus the state q_k. The edges are as in A except that the only edges to q_k come from q_j. q_i is the start. Accepting state is q_k. $L_{i,j,k}$ is the language accepted by $A_{i,j,k}$.

Now consider the words $w \in L \cap \Sigma^n$. Each such word is accepted on some path in A leading from q_0 to some accepting state q_a. Fix one such accepting state so that a constant fraction of all words w is accepted and make the other accepting states rejecting. On an accepting path for w the states appear without violating the topological ordering of the ambiguity graph. So, we may fix a sequence of states $q_0, q_{i_1}, \ldots, q_a$ such that $w \in L_{0,i_1,i_2} L_{i_2,i_3,i_4} \cdots L_{i_{2k-2},i_{2k-1},a}$. Since there are only finitely many such sequences we are done.

b) Similar to a), we get k languages $L_1, \ldots, L_k$ decidable by small unfa's A_i, such that

$$\frac{|L_1 \cdots L_k \cap ((\Sigma^m \cap K)\#)^n|}{|((\Sigma^m \cap K)\#)^n|} = \Omega(1)$$

for infinitely many n.

A partition of the letters of words in $(\Sigma^m \#)^n$ is given by mapping the nm letters to the k unfa's. There are at most $\binom{n}{k-1} \cdot (m+1)^{k-1}$ possible partitions. So some partition must be consistent with accepting paths for a fraction of $1/poly(n)$ of $((\Sigma^m \cap K)\#)^n$. Fix one such partition. Then for each words $w \in (\Sigma^m \#)^n$ an unfa is responsible for some prefix u, followed by a concatenation of words of the form $\#\Sigma^m$, and finally a word of the form $\#v$. For all i we fix a prefix u_i, a suffix v_i, and states q_i, q_i' entered when reading the first and final occurrence of $\#$, such that as many words from $((\Sigma^m \cap K)\#)^n$ as possible are accepted under this fixing. At least a fraction of $size^{-k}/2^{O(mk)} = 1/poly(n)$ of $((\Sigma^m \cap K)\#)^n$ has accepting paths consistent with this fixing.

If any A_i accepts less than a polynomial fraction (compared to the projection of $(\Sigma^m \cap K)\#)^n$ to the responsibility region of A_i) then overall less than a polynomial fraction is accepted. Hence one A_i can be found, where from q_i a polynomial fraction of words in $(\Sigma^m \cap K)\#)^{n/k}$ leads to non-terminally rejecting states in A_i. Making one non-terminally rejecting state reached by a $\#$ edge accepting and removing the original accepting states yields an unfa that accepts the desired subset for infinitely many n.

□

Applying Theorem 5 we can prove an exponential gap between nfa's and nfa's with polynomial ambiguity. This proof is also substantially simpler[3] than the proof of an exponential gap between polynomial ambiguity and exponential ambiguity for the language $(0 + (01^*)^{n-1}0)^*$ in [HL98].

Theorem 6 *There is a family of languages KL_m such that KL_m can be recognized by an nfa with advice $\Theta(n)$, leaf $2^{\Theta(n)}$ and size $poly(m)$, while every nfa with polynomial leaf number/ambiguity needs size at least $2^{\Omega(m)}$ to recognize KL_m.*

Proof: Let $LNDISJ_m = \{x_1 \cdots x_m \cdot y_1 \cdots y_m | x_i, y_i$ encode elements from a size m^{32} universe and the sets $\cup_i x_i$ and $\cup_i y_i$ intersect non-trivially$\}$. Moreover, let $KL_m = (LNDISJ_m\#)^*$.

Given a polynomial ambiguity nfa for KL_m, we get an unfa accepting a fraction of $1/poly(n)$ of $(LNDISJ_m\#)^n$ for infinitely many n by Theorem 4b). Then we simulate

[3] If the known results about communication complexity are for free (i.e., not included in the measurement of the proof difficulty).

the unfa by a nondeterministic communication protocol, where player C_I receives all x and player C_{II} all y inputs. The protocol needs $O(n \cdot \log size_A)$ bits to work correctly on a $1/poly(n)$ fraction of $(LNDISJ_m\#)^n$ and has unambiguous nondeterminism. A result from [HS96] implies that this task needs communication $\Omega(nm)$ and thus $size_A \geq 2^{\Omega(m)}$. □

Thus, we have another strong separation between the size of automata with polynomial ambiguity and the size of automata with exponential ambiguity. The situation seems to be more complicated, if one compares constant and polynomial ambiguity. Ravikumar and Ibarra [RI89] and Hing Leung [HL98] considered it as the central open problem related to the degree of ambiguity of nfa's. Here, we can only show that there is a family KON_m of languages with small size nfa's of polynomial ambiguity, while nfa's of ambiguity $\sqrt{m}$ are exponentially larger. In the following theorem we describe a candidate for a language that has efficient nfa's only when ambiguity is polynomial. Furthermore the language exhibits an almost optimal gap between the size of unfa's and polynomial ambiguity nfa's. In the proof the rank of the communication matrix of KON_m is shown to be large by a reduction from the disjointness problem.

Theorem 7 *Let $KON_m = \{0,1\}^*0M_m0\{0,1\}^*$, where M_m contains all words in $\{0,1\}^*$ with a number of ones that is divisible by m. KON_m can be recognized by an nfa A with $ambig_A(n), leaf_A(n) = \Theta(n)$ and size $m+2$, while any nfa with ambiguity k for KON_m needs at least $2^{(m-1)/k} - 2$ states.*

PROOF: Since the upper bound of theorem 7 is obvious, we focus on proving the lower bound.

Consider the communication problem for the complement of the disjointness predicate $NDISJ_l$. The inputs are of the form $x, y \in \{0,1\}^l$, where x and y are interpreted as incidence vectors of subsets of a size l universe. The goal is to find out, whether the two sets have a nontrivial intersection. Note that the rank of the communication matrix M_{NDISJ_l} is $2^l - 1$. We reduce $NDISJ_{m-1}$ to KON_m, i.e., identify a submatrix of M_{KON_m} that is the communication matrix $M_{NDISJ_{m-1}}$.

Consider inputs to KON_m of the form $01^{r_1}\ldots01^{r_t}$ with $t < m$, $0 < r_i$, and $\{r_t, r_t + r_{t-1}, \ldots, r_t + \cdots + r_1\} = s \subseteq \{1, \ldots, m-1\}$ with addition over $\mathbb{Z}_m$. For any subset $s \subseteq \{1, \ldots, m-1\}$ one can find such an input x_s. These 2^{m-1} inputs correspond to the rows of our submatrix.

For each subset $s = \{s_1, \ldots, s_t\} \subseteq \{1, \ldots, m-1\}$ fix an input y_s of the form $01^{r_1}\ldots01^{r_t}$ with $t < m, 0 < r_i$, and $\{r_1, r_1 + r_2, \ldots, r_1 + \cdots + r_t\} = \{m - s_1, \ldots, m - s_t\}$. These 2^{m-1} inputs correspond to the columns of our submatrix.

Now consider the obtained submatrix: if s and r intersect non-trivially, then $x_s y_r \in KON_m$. On the other hand, if s and r are disjoint, then there is no sub-word $01\ldots10$ of $x_s y_r$ which has a number of ones divisible by m. So $x_s y_r$ is not in KON_m. We have identified a submatrix of rank $2^{m-1} - 1$. Applying Theorem 1(b) we obtain our lower bound. □

For every constant m, the language KON_{m^2} of Theorem 7 can be recognized with size $O(m^2)$, leaf number and ambiguity $\Theta(n)$, and advice $\Theta(n)$, while every $m-$ambiguous nfa has size $2^{\Omega(m)}$. We conjecture that the language KON_m cannot be computed by nfa's with constant ambiguity and size $poly(m)$.

5 Conclusions and Open Problems

We have shown that communication complexity can be used to prove lower bounds on the size of nfa's with small ambiguity. This approach is limited, because for nontrivial bounds ambiguity has to be smaller than the size of a minimal nfa. Is it possible to prove lower bounds for automata with arbitrarily large, but constant ambiguity, when equivalent automata of small size and polynomial ambiguity exist?
In this context it would be also of interest to investigate the fine structure of languages with regard to constant ambiguity. At best one could show exponential differences between the number of states for ambiguity k and the number of states for ambiguity $k+1$. Observe however, that such an increase in power is impossible provided that the size of unfa's does not increase substantially under complementation [K00]. Analogous questions apply to polynomial and exponential ambiguity.
Are there automata with non-constant but sub-linear ambiguity? A negative answer establishes Theorem 3 also for ambiguity as complexity measure.
Other questions concern the quality of communication as a lower bound method. How far can *Rank* resp. $2^{\sqrt{cc(L)}}$ be from the actual size of minimal unfa's? Note that the bounds are not polynomially tight. Are there alternative lower bound methods?
Finally, what is the complexity of approximating the minimal number of states of an nfa?

References

[Ab78] Abelson, H.: Lower bounds on information transfer in distributed computations. *Proc. 19th IEEE FOCS*, IEEE 1978, pp. 151–158.

[AUY83] Aho, A.V., Ullman, J.D., Yannakakis, M.: On notions of informations transfer in VLSI circuits. *Proc. 15th ACM STOC*, ACM 1983, pp. 133–139.

[DHRS97] Duriš,P. , Hromkovič, J. , Rolim, J.D.P., Schnitger, G.: Las Vegas versus determinism for one-way communication complexity, finite automata, and polynomial-time computations. *Proc. STACS'97*, LNCS 1200, Springer-Verlag 1997, 117–128.

[DHS96] Dietzfelbinger, M., Hromkovič, J., Schnitger, G.: A comparison of two lower bound methods for communication complexity. *Theoretical Computer Science* 168 (1996), 39–51.

[GKW90] Goldstine, J., Kintala, C.M.R., Wotschke, D.: On measuring nondeterminism in regular languages. *Information and Computation* 86 (1990), 179–194.

[GLW92] Goldstine, J., Leung, H., Wotschke, D.: On the relation between ambiguity and nondeterminism in finite automata. *Information and Computation* 100 (1992), 261–270.

[Hr86] Hromkovič, J.: Relation between Chomsky Hierarchy and Communication Complexity Hierarchy. *Acta Math. Univ. Com.*, vol. 48–49, 1986, 311–317.

[Hr97] Hromkovič, J.: *Communication Complexity and Parallel Computing.* Springer, 1997.

[Hr00] Hromkovič, J.: Communication protocols: An exemplary study of the power of randomness. In: *Handbook of Randomized Computing* (P. Pardalos, S. Rajarekaran, J. Reif, J. Rolim (Eds.)), Kluwer Publ., to appear.

[HKK00] Hromkovič, J., Karhumiki, J., Klauck, H., Seibert, S., Schnitger, G.: Measures of nondeterminism in finite automata. In: *Proc. ICALP* '00, Lecture Notes in Computer Science 1853, Springer-Verlag 2000, pp. 199–210.

[HL98] Hing Leung: Separating exponentially amgigous finite automata from polynomially ambigous finite automata. *SIAM J. Computing* 27 (1998), 1073–1082.

[HS96] Hromkovič, J., Schnitger, G.: Nondeterministic communication with a limited number of advice bits. *Proc. 28th ACM STOC*, ACM 1996, pp. 451–560.

[HSW97] Hromkovič, J., Seibert, S., Wilke, T. Translating regular expressions into small ϵ-free nondeterministic finite automata. *Proc. STACS'97*, LNCS 1200, Springer-Verlag 1997, pp. 55–66.

[IR86] Ibarra, O., Ravikumar, B.: On sparseness, ambiguity and other decision problems for acceptors and tranducers. *Proc. 3rd STACS* '86, Lecture Notes on Computer Science 210, Springer-Verlag 1986, pp. 171–179.

[Ji99] Jirásková, G.:Finite automata and communication protocols. In: *Words, Sequences, Grammars, Languages: Where Biology, Computer Science, Linguistics and Mathematics Meet II* (C. Martin-Vide, V. Mitrana, Eds.), to appear.

[KNSW94] Karchmer, M., Saks, M., Newman, I., Wigderson, A.: Non-deterministic communication complexity with few witnesses. *Journal of Computer and System Sciences* 49 (1994), 247–257.

[K98] Klauck, H.: Lower bounds for computation with limited nondeterminism. *Proc. 13th IEEE Conference on Computational Complexity*, IEEE 1998, pp. 141–153.

[K00] Klauck, H.: On automata with constant ambiguity. Manuscript.

[KN97] Kushilevitz, E., Nisan, N.:*Communication Complexity.* Cambridge University Press, 1997.

[L90] Lovász, L.: Communication Complexity: A survey. In: *Paths, Flows, and VLSI Layout*, Springer 1990.

[MS82] Mehlhorn, K., Schmidt, E.: Las Vegas is better than determinism in VLSI and distributed computing. *Proc. 14th ACM STOC*, ACM 1982, pp. 330–337.

[MF71] Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars and formal systems. *Proc. 12th Annual Symp. on Switching and Automata Theory*, 1971, pp. 188–191.

[Mo71] Moore, F.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic and two-way finite automata. *IEEE Trans. Computing* 20 (1971), 1211–1214.

[M80] Micali, S.: Two-way deterministic finite automata are exponentially more succinct than sweeping automata. *Information Processing Letters* 12 (1981).

[NW95] Nisan, N., Wigderson, A.: On ranks vs. communication complexity. *Combinatorica* 15 (1995), 557–565.

[PS82] Papadimitriou, C., Sipser, M.: Communication complexity. *Proc. 14th ACM STOC*, ACM 1982, pp. 196–200.

[PS84] Papadimitriou, C., Sipser, M.: Communication Complexity. *Journal of Computer and System Sciences* 28 (1984), pp. 260–269.

[RI89] Ravimkumar, B., Ibarra, O.: Relating the type of ambiguity of finite automata to the succinctness of their representation. *SIAM J. Computing* 19 (1989), 1263–1282.

[RS59] Rabin, M., Scott, D.: Finite automata and their decision problems. *IBM J. Res. Development* 3 (1959), 114–125.

[S80] Sipser, M.: Lower Bounds on the Size of Sweeping Automata. *Journal of Computer and System Sciences* 21(2) (1980), pp. 195–202.

[Sc78] Schmidt, E.: Succinctness of descriptions of context-free, regular and finite languages. *Ph. D. thesis*, Cornell University, Ithaca, NY, 1978.

[SH85] Stearns, R., Hunt, H.: On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Computing* 14 (1985), 598–611.

[Th79] Thompson, C.D.: Area-time complexity for VLSI. *Proc. 11th ACM STOC*, ACM 1979, pp. 81–88.

[Th80] Thompson, C.D.: A complexity theory for VLSI. Doctoral dissertation. CMU-CS-80-140, Computer Science Department, Carnagie-Mellon University, Pittsburgh, August 1980, 131 p.

[Y91] Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences* 43 (1991), pp. 223–228.

[Y79] Yao, A.: Some complexity questions related to distributed computing. *Proc. 11th ACM STOC*, ACM 1979, pp. 209–213.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'