Theoretical Informatics and Applications
Theoret. Informatics Appl. **37** (2003) 39–49
DOI: 10.1051/ita:2003007

LOWER BOUNDS FOR LAS VEGAS AUTOMATA BY INFORMATION THEORY *

Mika Hirvensalo[1] and Sebastian Seibert[2]

Abstract. We show that the size of a *Las Vegas* automaton and the size of a complete, minimal *deterministic* automaton accepting a regular language are polynomially related. More precisely, we show that if a regular language L is accepted by a Las Vegas automaton having r states such that the probability for a definite answer to occur is at least p, then $r \geq n^p$, where n is the number of the states of the minimal deterministic automaton accepting L. Earlier this result has been obtained in [2] by using a reduction to *one-way Las Vegas communication protocols*, but here we give a direct proof based on information theory.

Mathematics Subject Classification. 68Q19, 68Q10, 94A15.

1. Introduction

A major topic in the theory of computational complexity is to compare the computational powers between nondeterministic and deterministic devices [5]. Nowadays the knowledge on this discipline is far too weak to provide definite solutions to the longstanding open questions, such as: is it true that deterministic polynomial-time Turing machine computation is strictly less powerful that its nondeterministic counterpart? Are there some computational tasks which can be solved probabilistically in polynomial time, but not deterministically?

Keywords and phrases. Las Vegas automata, information theory.

* *Research supported by DAAD and the Academy of Finland under a common grant 864524.*

[1] TUCS-Turku Centre for Computer Science and Department of Mathematics, University of Turku, FIN-20014 Turku, Finland; mikhirve@cs.utu.fi. Supported by the academy of Finland under grant 44087.

[2] Lehrstuhl für Informatik I, RWTH Aachen, Ahornstraße 55, 52074 Aachen, Germany; seibert@I1.Informatik.RWTH-Aachen.DE.

On the other hand, for some models of computation other than Turing machines, separations between determinism and nondeterminism can be established: It is a well-known fact that there are regular languages which can be recognized by a nondeterministic finite automaton having n states, but cannot be recognized by a deterministic automaton having less than 2^n states.

A finite *Las Vegas* automaton is a probabilistic finite automaton (with a single initial state) whose states are divided into three disjoint classes: accepting, rejecting, and *ignorant* states. It is required that for each input word, one of the following two conditions holds: (1) every computation leads to an accepting or to an ignorant state; or (2) every computation leads to a rejecting or to an ignorant state. The interpretation is, as in the case of deterministic and nondeterministic automata, that the language accepted by the Las Vegas automaton consists exactly of those input words which can lead into an accepting state. Intuitively, the above restriction for Las Vegas nondeterminism says that a Las Vegas automaton must always give a correct answer: it is forbidden that some input word has a computation ending at an accepting and a computation ending at a rejecting state. Before calling such an automaton Las Vegas automaton, we also fix another postulate: all input words can result in an ignorant state with a probability of at most some fixed $\epsilon < 1$.

We examine Las Vegas automata, and use argumentation based on information theory to obtain the following result: if a regular language L is accepted by a Las Vegas automaton having r states and ϵ as the highest probability for reaching some ignorant state, then $r \geq n^{1-\epsilon}$, where n is the cardinality of the complete minimal deterministic automaton accepting L. This result has already been obtained in [2] by using a reduction to one-way Las Vegas communication protocols.

The motivation of this article is to give a direct proof for the above result, in order to learn more about randomized computations, and especially, to learn more about the following questions: why Las Vegas automata cannot reach the exponential state reduction (over the deterministic automata) which is possible for nondeterministic automata? The bound $r \geq n^{1-\epsilon}$ can be shown to be tight up to a multiplicative constant [2], but not exactly strict. Why so? Is it true that for each Las Vegas automaton there exists some *normal form* [3] which admits the random choices only at the beginning, and then acts deterministically? In this article, we can give quite evident heuristic argumentation for the first two questions.

2. Notations and preliminaries

In this section, we represent the basic facts on finite automata and information theory. For the concepts on those topics not represented or mentioned here, references [6] and [1] are recommended.

2.1. Finite automata

Let L be a regular language over an alphabet A. It is a well-known fact [6] that there are only finitely many equivalence classes in A^* with respect to relation $\sim_L$

defined by

$$w_1 \sim_L w_2 \text{ if and only if for each word } x \in A^*$$
$$w_1 x \in L \iff w_2 x \in L. \tag{1}$$

There is also a canonical way to construct the minimal complete deterministic automaton $\mathcal{A}$ accepting L, see [6]. The important thing we here need to know about the construction is that the states of $\mathcal{A}$ are exactly the equivalence classes $[w_1]$, ..., $[w_n]$ of relation $\sim_L$.

If $\{w_1, \ldots, w_n\}$ is a set of representatives of the equivalence classes, we say that a set S of words *separates* the classes (or that S is a *separating set*), if for each pair $w_i \neq w_j$ there exists $s \in S$ such that either $w_i s \in L$ and $w_j s \notin L$ or *vice versa*. By the definition of the relation $\sim_L$, there exists a separating set having cardinality of at most $\binom{n}{2} = n \cdot (n-1)/2$.

For a deterministic automaton $\mathcal{A}$, $\delta_{\mathcal{A}}$ stands for the transition function, and hence $\delta_{\mathcal{A}}(s, w)$ stands for the state which $\mathcal{A}$ enters when the word w is given as an input when $\mathcal{A}$ is initially in state s. We also define a *type function* T from the state set into $\{0, 1\}$ by $T(q) = 1$, if q is an accepting state, and $T(q) = 0$ for a rejecting state (a state which is not accepting) q.

A finite *probabilistic automaton* $\mathcal{P}$ (over A) having a state set R is defined as an ordinary finite automaton $\mathcal{A}$ (over A), but the transition function $\delta_{\mathcal{A}}$ is replaced with a function $\delta_{\mathcal{P}}$, whose value $\delta_{\mathcal{P}}(r, a)$ for any fixed pair $(r, a) \in R \times A$ is a probability distribution on the set R. That is, for any pair $(r, a) \in R \times A$ and any state r', $p(\delta_{\mathcal{P}}(r, a) = r')$ is the probability to enter into the state r' when letter a is read in state r. It is easy to see that $\delta_{\mathcal{P}}$ can be uniquely extended in such a way, that for any pair $(r, w) \in R \times A^*$, $\delta_{\mathcal{P}}(r, w)$ is a probability distribution on R. Hence for any pair $(r, w) \in R \times A^*$, $\delta_{\mathcal{P}}(r, w)$ can be interpreted (and will be interpreted) as a random variable which has all the states of R as its potential values.

Definition 1. A probabilistic automaton $\mathcal{P}$ with initial state l_0 is called a *Las Vegas automaton*, if there is a fixed $\epsilon \in [0, 1)$ and a type function $T : R \to \{0, 1, I\}$ which satisfies the following: first, there is no word $w \in A^*$, for which events $T(\delta_{\mathcal{P}}(l_0, w)) = 0$ and $T(\delta_{\mathcal{P}}(l_0, w)) = 1$ could *both* occur, and secondly, that for each $w \in A^*$, event $T(\delta_{\mathcal{P}}(l_0, w)) = I$ occurs with a probability of at most ϵ.

Thus the Las Vegas condition for a probabilistic automaton means that when reading any word $w \in A^*$, the automaton must either enter into a state of type 0 or I, or into a state of type 1 or I. Moreover, the probability of entering into a state having type I must be at most ϵ for each word $w \in A^*$. If $T(r) = 1$, state r is called *accepting*, if $T(r) = 0$, then r is *rejecting*, and r is said to be *ignorant*, if $T(r) = I$.

Definition 2. The *language accepted by a Las Vegas automaton* is defined to consist exactly of those words $w \in A^*$, for which $T(\delta_{\mathcal{P}}(r_0, w)) = 1$ can occur.

If $\mathcal{P}$ is a Las Vegas automaton, one can even ignore the probabilities and consider only the underlying *nondeterministic automaton* $\mathcal{N}$ that has the same state

set and the same accepting states as $\mathcal{P}$, but the transition function $\delta_{\mathcal{P}}$ is replaced with transition relation $\delta_{\mathcal{N}}$ defined as $\delta_{\mathcal{N}}(r,a) \to q$ if and only if $\delta_{\mathcal{P}}(r,a)$ has value q with a nonzero probability. Clearly $\mathcal{N}$ and $\mathcal{P}$ accept the same language, hence all languages accepted by Las Vegas automata are regular.

2.2. Information theory

For a set $X = \{x_1, x_2, \ldots, x_n\}$ we can assign probabilities $p(x_i)$, and treat X as a random variable. The *(binary) entropy* of the random variable X is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i).$$

In the above sum, $0 \cdot \log_2 0$ is defined to be 0. Using the basic properties of the logarithm function, such as concavity, it is easy to show that $0 \leq H(X) \leq \log_2 n$. It is also worth noticing that in the case $p_i = \frac{1}{n}$ for each i, we have $H(X) = \log_2 n$. A usual interpretation of the binary entropy $H(X)$ is that it measures the *uncertainty* about random variable X [1]. That is, the value of $H(X)$ is the number of bits needed (in average) to encode the elements of X.

If $Y = \{y_1, \ldots, y_m\}$ is another random variable, the *joint entropy* of X and Y is defined naturally as

$$H(X,Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 p(x_i, y_j),$$

where $p(x_i, y_j)$ is the usual joint probability distribution of X and Y. In a similar way, the joint entropy can be defined for more than two random variables. The *conditional entropy of X provided that the value of Y is known to be y_k*, is defined as

$$H(X \mid y_k) = -\sum_{i=1}^{n} p(x_i \mid y_k) \log_2 p(x_i \mid y_k),$$

where $p(x_i \mid y_k)$ is the usual conditional probability. *The conditional entropy of X provided Y is known* is then defined as an expected value

$$H(X \mid Y) = \sum_{k=1}^{m} p(y_k) H(X \mid y_k).$$

It is easy to see that in the case $Y = X$ we have $H(X \mid X) = 0$, an equation whose intutitive meaning is clear.

The *information about X when Y is known* is defined as

$$I(X;Y) = H(X) - H(X \mid Y). \tag{2}$$

The information about X when Y is known thus tells us how much our certainty about X increases when the knowledge about Y is provided. In the case $Y = X$ we have obviously $I(X;X) = H(X)$, which also has an intuitively clear meaning.

The proofs of the following propositions, as well as some of their interpretations can be found in [1].

Proposition 1. $H(X \mid Y) = H(X,Y) - H(Y)$ *(relation between conditional and joint entropy).*

The above proposition has two interesting consequences. The first one is straightforward and tells us that

$$I(X;Y) = I(Y;X), \tag{3}$$

meaning that the information is a symmetric property. For the second consequence, we notice that evidently $H(X \mid X,Y) = 0$, but also $H(X \mid X,Y) = H(X,X,Y) - H(X,Y)$, so $H(X,X,Y) = H(X,Y)$, which means that duplicating a variable does not affect the uncertainty.

Proposition 2. $H(X \mid Y) \leq H(X)$ *(condition Y cannot increase the uncertainty about X).*

Notice that the above proposition implies that the information is always non-negative.

Proposition 3. *If X, Y, and Z are random variables having distributions $p(x)$, $p(y) = \sum_x p(y\,|\,x)p(x)$, and $p(z) = \sum_y p(z\,|\,y)p(y)$ respectively, then*

$$I(X;Z) \leq I(X;Y).$$

(The Data Processing Inequality.)

3. Relating the automata sizes *via* information

Let L be a regular language with some fixed set $\{w_1, \ldots, w_n\}$ of representatives of classes of $\sim_L$ as defined in (1). Thus the minimal complete deterministic automaton $\mathcal{A}$ accepting L has n states. Let also q_0 be the initial state of this automaton, and $S = \{s_1, \ldots, s_t\}$ a fixed set of words which separates the equivalence classes. Recall that the type function T defined on the state set has value 1 for accepting states and value 0 for the rejecting ones.

Definition 3. The *characteristic vectors* $\boldsymbol{x}^{(1)}$, …, $\boldsymbol{x}^{(n)} \in \{0,1\}^t$ of a regular language L with respect to representatives w_1, …, w_n and separating set S are defined as

$$\boldsymbol{x}_j^{(i)} = T(\delta_{\mathcal{A}}(q_0, w_i s_j)). \tag{4}$$

Thus $\boldsymbol{x}_j^{(i)} = 1$ if and only if $w_i s_j$ is in the language L. Because set S separates the equivalence classes, it is clear that the vectors $\boldsymbol{x}^{(i)}$ are distinct. In fact, since we assume the representatives w_1, …, w_n and the separating set S to be fixed, there is even a one-to-one correspondence between the representatives w_i and the vectors $\boldsymbol{x}^{(i)}$.

To simplify the construction of the vectors $\boldsymbol{x}^{(i)}$ a little bit, we notice that (4) can be also written as

$$\boldsymbol{x}_j^{(i)} = T(\delta_{\mathcal{A}}(\delta_{\mathcal{A}}(q_0, w_i), s_j)). \tag{5}$$

Thus, in order to determine the coordinates of $\boldsymbol{x}^{(i)}$, it suffices only to *once* find out the state $q_i = \delta_{\mathcal{A}}(q_0, w_i)$, and then, for each j, to discover if $\delta_{\mathcal{A}}(q_i, s_j)$ is an accepting state or not. It will turn out that this very simple idea behind representation (5), suitably adjusted for Las Vegas automata can be used to derive a lower bound for the cardinality of a Las Vegas automaton accepting L.

Let now $\mathcal{P}$ be a Las Vegas automaton accepting language L, having r states, and l_0 as the initial state. Assume also that $T(\delta_{\mathcal{P}}(l_0, w)) = I$ occurs with a probability of at most ϵ for any word $w \in A^*$.

Definition 4. We define X as a random variable which has any of the characteristic vectors as its values, each one with a probability of $1/n$.

Definition 5. A random variable Y depending on X is defined as follows: if the value of X is $\boldsymbol{x}^{(i)}$, then set $Y = \delta_{\mathcal{P}}(l_0, w_i)$, where w_i is the representative corresponding to the characteristic vector $\boldsymbol{x}^{(i)}$. Thus Y has the states of the Las Vegas automaton $\mathcal{P}$ as its potential values.

Recall that since $\mathcal{P}$ is a probabilistic automaton, the value of $\delta_{\mathcal{P}}(l_0, w_i)$ is a random variable even if the value $\boldsymbol{x}^{(i)}$ of X is fixed.

Definition 6. For each $j \in \{1, \ldots, t\}$ we define a random variable Z_j depending on Y by

$$Z_j = T(\delta_{\mathcal{P}}(Y, s_j))$$

and finally we define a random variable $Z = Z_1 \times \ldots \times Z_t$ having its potential values in $\{0, 1, I\}^t$.

The value of Z can be seen as an attempt to reconstruct the value of X: if X has value $\boldsymbol{x}^{(i)}$, we first take the word w_i corresponding to the vector $\boldsymbol{x}^{(i)}$ and give the word w_i as an input to automaton $\mathcal{P}$ (beginning at the initial state). The computation ends in some (randomly chosen) state l_i of $\mathcal{P}$, which is defined as the value of Y. After this, for each $j \in \{1, 2, \ldots, t\}$ we find the value for the j-th coordinate of Z by running computation on $\mathcal{P}$ with input word s_j, now beginning in state l_i. In symbols: if X has value $\boldsymbol{x}^{(i)}$, then

$$Z_j = T(\delta_{\mathcal{P}}(Y, s_j)) = T(\delta_{\mathcal{P}}(\delta_{\mathcal{P}}(l_0, w_i), s_j)). \tag{6}$$

Notice the similarity between equations (5) and (6).

Let us now suppose that X assumes some particular value $\boldsymbol{x}^{(i)} \in \{0, 1\}^t$ and Z value $\boldsymbol{z} \in \{0, 1, I\}^t$. Since a Las Vegas automaton can never give an erratic answer, we must have $\boldsymbol{z}_j = \boldsymbol{x}_j^{(i)}$ for all those coordinates for which $\boldsymbol{z}_j \neq I$. But the probability that a single coordinate $\boldsymbol{z}_j$ has value I is at most ϵ, so we should learn something about the value of X when the value of Z is known. The following lemma provides a lower bound for the information that Z gives about X.

Lemma 1. *Let X be a random variable with potential values $\boldsymbol{x}^{(1)}$, ..., $\boldsymbol{x}^{(n)}$ in $\{0,1\}^t$. Also, let Z be a random variable whose value is a vector $\boldsymbol{z} \in \{0,1,I\}^t$ obtained from the value $\boldsymbol{x}$ of X such that if $\boldsymbol{x}_i = 0$ (resp. $\boldsymbol{x}_i = 1$), then $\boldsymbol{z}_i$ is either 0 (resp. 1) or I, but for each i, $\boldsymbol{z}_i = I$ occurs with a probability of at most ϵ. Then $I(X;Z) \geq (1-\epsilon)I(X;X)$.*

Proof. Since $I(X;Z) = H(X) - H(X \mid Z)$ and $I(X;X) = H(X)$, the claim is equivalent to $H(X \mid Z) \leq \epsilon H(X)$. First we decompose X into its coordinates as follows: we introduce random variables X_1, ..., X_t having their potential values in $\{0,1\}$ with a joint probability distribution

$$p(x_1, \ldots x_t) = \begin{cases} p(\boldsymbol{x}), & \text{if } \boldsymbol{x} = (x_1, \ldots, x_t) \text{ is} \\ & \text{a characteristic vector of } L, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Analogously to (7), we define the conditional probabilities as

$$p(x_1, \ldots, x_t \mid z_1, \ldots, z_t) = p(\boldsymbol{x} \mid z_1, \ldots, z_t), \tag{8}$$

if $\boldsymbol{x} = (x_1, \ldots, x_t)$ is a characteristic vector, and

$$p(x_1, \ldots, x_t \mid z_1, \ldots, z_t) = 0 \tag{9}$$

otherwise.

According to (7–9), we can write the entropy as

$$H(X \mid Z) = H(X_1, X_2, \ldots, X_t \mid Z_1, Z_2, \ldots, Z_t). \tag{10}$$

Expression (10) can be written as

$$\begin{aligned} & H(X_1, X_2, \ldots, X_t \mid Z_1, Z_2, \ldots, Z_t) \\ = \;& p(Z_1 = 0) H(X_1, X_2, \ldots, X_t \mid 0, Z_2, \ldots, Z_t) \\ + \;& p(Z_1 = 1) H(X_1, X_2, \ldots, X_t \mid 1, Z_2, \ldots, Z_t) \\ + \;& p(Z_1 = I) H(X_1, X_2, \ldots, X_t \mid I, Z_2, \ldots, Z_t). \end{aligned} \tag{11}$$

In the above formula and afterwards, entries 0, 1, and I at the i-th condition stand for the conditions $Z_i = 0$, $Z_i = 1$, and $Z_i = I$, respectively. We will denote $x_0 = p(X_1 = 0)$, $x_1 = p(X_1 = 1)$, $\epsilon_0 = p(Z_1 = I \mid X_1 = 0)$, and $\epsilon_1 = p(Z_1 = I \mid X_1 = 1)$. By the assumption, $\epsilon_0 \leq \epsilon$ and $\epsilon_1 \leq \epsilon$. For brevity, we also denote the above three entropies in the sum (11) by $H^{(0)}$, $H^{(1)}$, and $H^{(I)}$, respectively. Notice that since $Z_1 = I$ does not give any information about any X_i, event $Z_1 = I$ will not decrease the entropy, whereas events $Z_1 = 0$ and $Z_1 = 1$ may do so. Therefore $H^{(I)} \geq H^{(0)}$ and $H^{(I)} \geq H^{(1)}$. Notice that now

$$\begin{aligned} p(Z_1 = 0) \;&=\; P(Z_1 = 0 \mid X_1 = 0) P(X_1 = 0) \\ &+\; P(Z_1 = 0 \mid X_1 = 1) P(X_1 = 1) = (1 - \epsilon_0) x_0 + 0 \cdot x_1, \end{aligned}$$

$p(Z_1 = 1) = (1 - \epsilon_1)x_1$, and $p(Z_1 = I) = x_0\epsilon_0 + x_1\epsilon_1$. Thus (11) can be rewritten and estimated as

$$\begin{aligned} & H(X_1, X_2, \ldots, X_t \mid Z_1, Z_2, \ldots, Z_t) \\ = \;& (1-\epsilon_0)x_0H^{(0)} + (1-\epsilon_1)x_1H^{(1)} + (x_0\epsilon_0 + x_1\epsilon_1)H^{(I)} \\ = \;& x_0\epsilon_0(H^{(I)} - H^{(0)}) + x_1\epsilon_1(H^{(I)} - H^{(1)}) + x_0H^{(0)} + x_1H^{(1)} \\ \leq \;& \epsilon H^{(I)} + (1-\epsilon)(x_0H^{(0)} + x_1H^{(1)}). \end{aligned}$$

Recalling the meanings of all the notations, we result in an inequality

$$\begin{aligned} & H(X_1, X_2, \ldots, X_t \mid Z_1, Z_2, \ldots, Z_t) \\ \leq \;& \epsilon H(X_1, X_2, \ldots, X_t \mid I, Z_2, \ldots, Z_t) \\ + \;& (1-\epsilon)H(X_1, X_2, \ldots, X_t \mid X_1, Z_2, \ldots, Z_t). \end{aligned} \tag{12}$$

Choosing a collective notation $\widehat{X}$ for random variables X_3, …, X_t, and notation $\widehat{Z}$ for Z_3, …, Z_t, the above inequality can be written as

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid Z_1, Z_2, \widehat{Z}) \\ \leq \;& \epsilon H(X_1, X_2, \widehat{X} \mid I, Z_2, \widehat{Z}) + (1-\epsilon)H(X_1, X_2, \widehat{X} \mid X_1, Z_2, \widehat{Z}). \end{aligned}$$

Analogously, we conclude that

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid I, Z_2, \widehat{Z}) \\ \leq \;& \epsilon H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) + (1-\epsilon)H(X_1, X_2, \widehat{X} \mid I, X_2, \widehat{Z}), \end{aligned}$$

and that

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid X_1, Z_2, \widehat{Z}) \\ \leq \;& \epsilon H(X_1, X_2, \widehat{X} \mid X_1, I, \widehat{Z}) + (1-\epsilon)H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}), \end{aligned}$$

so (12) gives

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid Z_1, Z_2, \widehat{Z}) \\ \leq \;& \epsilon^2 H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) \\ + \;& \epsilon(1-\epsilon)H(X_1, X_2, \widehat{X} \mid I, X_2, \widehat{Z}) \\ + \;& \epsilon(1-\epsilon)H(X_1, X_2, \widehat{X} \mid X_1, I, \widehat{Z}) \\ + \;& (1-\epsilon)^2 H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}). \end{aligned}$$

The inequality thus obtained can also be written as

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid Z_1, Z_2, \widehat{Z}) \\ \leq\ & \epsilon H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) + (1-\epsilon) H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}) \\ +\ & \epsilon(1-\epsilon) H(X_1, X_2, \widehat{X} \mid I, X_2, \widehat{Z}) + \epsilon(1-\epsilon) H(X_1, X_2, \widehat{X} \mid X_1, I, \widehat{Z}) \\ -\ & \epsilon(1-\epsilon) H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) - \epsilon(1-\epsilon) H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}). \end{aligned}$$

If we can show that the contribution of the last four terms in the above sum is at most 0, we would have an estimate

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid Z_1, Z_2, \widehat{Z}) \\ \leq\ & \epsilon H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) + (1-\epsilon) H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}). \end{aligned} \tag{13}$$

Then, continuing the same reasoning (*cf.* inequalities (12) and (13)), we eventually get

$$\begin{aligned} & H(X_1, X_2, \ldots, X_t \mid Z_1, Z_2, \ldots, Z_t) \\ \leq\ & \epsilon H(X_1, X_2, \ldots, X_t \mid I, I, \ldots, I) \\ +\ & (1-\epsilon) H(X_1, X_2, \ldots, X_t \mid X_1, X_2, \ldots, X_t). \end{aligned}$$

But the knowledge that $Z_i = I$ for each i cannot reduce the uncertainty about the variables X_i, which is to say that $H(X_1, \ldots, X_t \mid I, \ldots, I) = H(X_1, \ldots, X_t)$. Moreover, it is clear that $H(X_1, \ldots, X_t \mid X_1, \ldots, X_t) = 0$, so the claim would follow.

It is still left to show that

$$\begin{aligned} & H(X_1, X_2, \widehat{X} \mid I, X_2, \widehat{Z}) + H(X_1, X_2, \widehat{X} \mid X_1, I, \widehat{Z}) \\ -\ & H(X_1, X_2, \widehat{X} \mid I, I, \widehat{Z}) - H(X_1, X_2, \widehat{X} \mid X_1, X_2, \widehat{Z}) \leq 0. \end{aligned}$$

Events $Z_1 = I$ and $Z_2 = I$ cannot give any additional information about X_i's, so the they do not affect the uncertainties. Removing those conditions and utilizing the identities $H(X \mid Y) = H(X, Y) - H(Y)$ and $H(X, X, Y) = H(X, Y)$, we can write the above sum as

$$\begin{aligned} & H(X_1, X_2, \widehat{X}, \widehat{Z}) - H(X_2, \widehat{Z}) + H(X_1, X_2, \widehat{X}, \widehat{Z}) - H(X_1, \widehat{Z}) \\ -\ & H(X_1, X_2, \widehat{X}, \widehat{Z}) + H(\widehat{Z}) - H(X_1, X_2, \widehat{X}, \widehat{Z}) + H(X_1, X_2, \widehat{Z}) \\ =\ & H(X_1 \mid X_2, \widehat{Z}) - H(X_1 \mid \widehat{Z}) \\ \leq\ & H(X_1 \mid \widehat{Z}) - H(X_1 \mid \widehat{Z}) = 0. \end{aligned}$$

The latest estimate is due to the fact that deleting a condition cannot decrease the entropy. □

Theorem 1. *Let L be a regular language whose minimal complete deterministic automaton has n states. Let $\mathcal{P}$ be an r-state Las Vegas automaton accepting language L. If for each input word $w \in A^*$ the probability that the computation ends at an ignorant state is at most ϵ, then $r \geq n^{1-\epsilon}$.*

Proof. Let X, Y, and Z be as in Definitions 4–6. Because X has n potential values with uniform distribution, $I(X;X) = H(X) = \log_2 n$. By Lemma 1 the information that Z gives about X can be estimated as

$$I(X;Z) \geq (1-\epsilon) I(X;X) = (1-\epsilon)\log_2 n.$$

On the other hand, the number of the states of the automaton $\mathcal{P}$ introduces a "bottleneck" for the amount of information we can learn about X when Z is known: first, Proposition 3 and equation (3) give us that $I(X;Z) \leq I(X;Y) = I(Y;X)$. By equation (2), $I(Y;X) = H(Y) - H(Y \mid X)$, and therefore $I(Y;X) \leq H(Y)$. Finally, $H(Y) \leq \log_2 r$, since there are only r potential values of Y (states of $\mathcal{P}$). Combining all the inequalities we have that

$$\begin{aligned} (1-\epsilon)\log_2 n &\leq I(X;Z) \leq I(X;Y) \\ &= I(Y;X) = H(Y) - H(Y \mid X) \\ &\leq H(Y) \leq \log_2 r, \end{aligned}$$

and hence the lower bound $r \geq n^{1-\epsilon}$ follows immediately. □

4. Open questions

In addition to the questions in the introduction, we can consider the following ones. Two of the above inequalities are of special interest: if inequality $I(X;Z) \leq I(X;Y)$ is even equality, then the variable Z would give us as much information about X as Y does, which somehow refers to the idea that the random choices (if the automaton makes any such) when "feeding" the separating words, do not increase the uncertainty about X.

On the other hand, if inequality $I(Y;X) \leq H(Y)$ is even equality, then necessarily $H(Y \mid X) = 0$, which means that Y is fully determined by X. But this would mean that when a representative w_i is given to the probabilistic automaton as input, there is a unique state of $\mathcal{P}$ where the computation ends. It should also be noted that the above bound can be derived by using *any* representatives of the equivalence classes, and, unless L is finite, there is necessarily an infinite equivalence class with an arbitrarily long representative.

Problem A) How should the representatives $w_1, \ldots, w_n$ be chosen to guarantee the maximality of $I(Y;X) - H(Y)$. What is the value of $H(Y) - \log r$ then?

Problem B) For which choice of S, $I(X;Z) - I(X;Y) = H(X \mid Y) - H(X \mid Z)$ becomes as small as possible? How does this choice affect the inequality $I(X;Z) \geq (1-\epsilon)\log n$?

In [2], there is given an example family of automata that shows the bound of Theorem 1 to be tight up to an multiplicative constant, *i.e.* $r = c \cdot n^{1-\epsilon}$ in this case. Interestingly, those automata do random steps only at the very beginning, which gave raise to the normal form conjecture by Hromkovič [3].

Relating that to our inequalities above, it means $(1-\epsilon)\log_2 n + \log_2 c = \log_2 r$. Especially, we have $H(Y \mid X) \leq \log_2 c$ and $I(X;Z) = I(X;Y)$ in that example. The question is whether this can be achieved for all languages by choosing appropriate representatives.

Acknowledgements. We thank one of the referees for pointing out that an analogous result to Lemma 1 was provided in [4].

References

[1] T.M. Cover and J.A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc. (1991).
[2] P. Ďuris, J. Hromkovič, J.D.P. Rolim and G. Schnitger, *Las Vegas Versus Determinism for One-way Communication Complexity, Finite Automata, and Polynomial-time Computations*. Springer, *Lecture Notes in Comput. Sci.* **1200** (1997) 117-128.
[3] J. Hromkovič, personal communication.
[4] H. Klauck, On quantum and probabilistic communication: Las Vegas and one-way protocols, in *Proc. of the ACM Symposium on Theory of Computing* (2000) 644-651.
[5] C.H. Papadimitriou, *Computational Complexity*. Addison-Wesley (1994).
[6] S. Yu, *Regular Languages*, edited by G. Rozenberg and A. Salomaa. Springer, *Handb. Formal Languages* **I** (1997).

Communicated by J. Hromkovič.
Received December, 2001. Accepted March, 2003.