Michal P. Chytil

Crossing-bounded computations and their relation to the LBA-problem

Kybernetika, Vol. 12 (1976), No. 2, (76)--85

Persistent URL: `http://dml.cz/dmlcz/124638`

KYBERNETIKA — VOLUME *12* (1976), NUMBER 2

Crossing-Bounded Computations and their Relation to the LBA-Problem

MICHAL P. CHYTIL

The space and the crossing complexity measures for one-tape off-line Turing machines are investigated. For nondeterministic machines an equivalence relation between the two measures is established.

1. INTRODUCTION

A relation between two important complexity measures (ammount of tape and maximal number of crossings of bounds between tape squares) will be established in this paper. We shall prove that the class of languages recognizable nondeterministically with the tape bound f is the same as the class of languages nondeterministically recognizable with the crossing bound f, provided $(\forall n)$ $(f(n) \geqq n)$. Since the crossing-bounded computations are at least as powerful as tape bounded ones in the deterministic case, it follows that deterministic crossing-bounded computations are "between" deterministic and nondeterministic computations with the same tape bound. The positive answer to the well-known LBA-problem (i.e. all context-sensitive languages could be recognized by deterministic linear-bounded automata) would therefore imply that tape-bounded and crossing-bounded computations are equally strong also in the deterministic case. That is why a solution of the question about the relation between these two types of deterministic computations would be interesting in any case. Negative answer would obviously imply negative answer to the LBA-problem, positive answer would establish a nontrivial equivalence relation between tape and crossing complexities.

2. BASIC NOTIONS

Machines we are interested in are one-tape off-line Turing machines (deterministic or nondeterministic). We assume that the alphabet (denoted A throughout this paper) of every such a machine M contains the blank symbol b. Suppose for simplicity that A is disjoint with $Q \times A$, where Q is the set of states, for every machine. The *instantaneous description* (ID) of a machine M is then every string of the form $v_1(q, a)\, v_2$, where $v_1, v_2 \in A^*$, $(q, a) \in Q \times A$. Such an ID $v_1(q, a)\, v_2$ denotes the total configuration in which the tape contains the word $v_1 a v_2$ (all other squares are blank), the head is scanning the symbol a and the finite control of M is in the state q.

If w_1 and w_2 are ID and M can change the configuration described by w_1 into the configuration described by w_2 in one step, then we say that the ID w_2 is *next* to the ID $w_1(w_1 \vdash_M w_2)$. Let $\delta : Q \times A \to$ subsets of $(Q \times A \times \{left, right\})$ be the next-state function of M. Then a $q \in Q$ is *final* iff $\delta(q, a)$ is empty for every $(q, a) \in Q \times A$. A machine M *accepts* a word w iff there is a computation beginning in the the initial state q_0 on the leftmost symbol of w and terminating in a final state. A machine M *recognizes* a language $L \subseteq (A - b)^*$ iff for every $w \in (A \setminus \{b\})^*$

$$w \in L \Leftrightarrow M \text{ accepts } w \,.$$

Let $f : N \to N$. We shall say that the language L is nondeterministicically (deterministically) *recognizable with the tape bound* f iff L is recognizable by a nondeterministic (deterministic) machine M such that for every $w \in L$ there is a computation of M accepting w and using not more than $f(|w|)$ tape squares. Replacing the words "using not more than $f(|w|)$ tape squares" by the words "crossing no bound between tape squares more than $f(|w|)$-times", we get the definition of languages nondeterministically (deterministically) *recognizable with the crossing bound* f.

In the sequel, the following notation will be used:

$DS(f)$ is the class of languages which are deterministically recognizable with the tape bound f,

$S(f)$ is the class of languages which are nondeterministically recognizable with the tape bound f,

$DC(f)$ is the class of languages recognizable deterministically with the crossing bound f,

$C(f)$ is the class of languages nondeterministically recognizable with the crossing bound f.

3. RELATIONS BETWEEN THE MEASURES

1. Theorem. Let f be an arithmetic function such that $(\forall n)\, (f(n) \geqq n)$. Then

1. $$DS(f) \subseteq DC(f)\,,$$

2. $$S(f) \subseteq C(f)\,;$$

Proof. Let M be a Turing machine (deterministic or nondeterministic) recognizing a language L with the tape bound f. We can construct a machine M' recognizing the same language with the crossing bound f. Moreover, if M is deterministic, then M' is deterministic, too. We give only an outline of the algorithm performed by M'.

1. M' begins with an input $v = x_1 \ldots x_n$ on its tape. It rewrites it to $v_1 = (q_0, x_1) x_2 \ldots x_n$ (i.e. the initial ID for M);
2. M' rewrites a copy of v_1 to the right end of the input word, so that these two identical IDs are separated by a marker (e.g. $*$) and one blank. (The tape contains the word $v_1 * bv_1$, now.)
3. M' rewrites the right ID v_1 to an ID v_2 next to v_1. (Simulation of one step of M.)
4. M' writes a copy of the rightmost ID (v_2 now) separated by $*$ and one blank behind the rightmost nonempty symbol.
5. M' rewrites it to a next ID.

And so on.

M' can obviously simulate every computation of M. It stops iff there is no ID next to the rightmost one, i.e. it accepts the input word iff the simulated computation accepts it. Every "simulating step" needs no more than 2 crossings. Most crossings are due to the "copying part" of the computation. If M' is constructed so that it can "remember" blocks of k symbols, then every ID w can be copied within $2 \,. [|w|/k] - 1$ crossings. The "copying part" of the computation is deterministic; if M is a deterministic machine then the "simulating part" of the computation is also deterministic.

An open problem is whether a converse of the first assertion of Theorem 1 holds, i.e. whether $DC(f) = DS(f)$. The negative answer would imply the negative answer to the LBA-problem, as will be shown in the fourth paragraph of this paper. The positive answer would e.g. imply the linear speed-up result for the crossing measure (namely: $DC(f) = DC(\varepsilon \,. f)$ for every $\varepsilon > 0, f(n) \geqq n$), because of the linear speed-up for the tape measure (cf. [1], p. 137).

Let us turn our attention to the nondeterministic case, now. We shall use a slightly modified notion of M-matrix, introduced by M. S. Paterson [2].

2. Definition. Let M be a Turing machine with the state-space Q and the alphabet A. A matrix $\mathscr{M}$ formed by elements of $(Q \times A) \cup A$ will be called *M-matrix* iff the following condition holds:
if u is the i-th row and v the $i + 1$-th row of $\mathscr{M}$, then there are $\alpha_1, \beta_1, \alpha_2, \beta_2 \in (A \cup (Q \times A))^*$ such that $\alpha_1 u \beta_1$ and $\alpha_2 v \beta_2$ are IDs of M and $\alpha_1 u \beta_1 \vdash_M \alpha_2 v \beta_2$. Apparently, every row of $\mathscr{M}$ is either an ID of M or a word in the alphabet A.

3. Definition. Let $\mathscr{M}$ be an M-matrix. We shall say that a row of $\mathscr{M}$ is

1) of *type a* (absence of the head) iff it is a word in the alphabet A,
2) of *type p* (the head is present), otherwise.

It may easily be seen that M-matrix is a convenient tool for describing the full history of a part of a computation restricted to a tape segment.

4. Definition. Let $\mathcal{M}$ be an M-matrix. Let R_1 and R_2 be its first and last row, respectively. Let S_1 and S_2 be its first and last column, respectively. We shall say that $\mathcal{M}$ is

1) a *computation M-matrix* (notation: $\mathrm{Comp}_M(\mathcal{M})$) iff $S_1 = S_2 \in \{b\}^* \,\&\, R_2$ is a terminal ID,

2) *over* $w = x_1 \dots x_n$ iff $R_1 = b^m(q_0, x_1)\, x_2 \dots x_n b^p$ for some $m, n, p \geqq 0$.

By Definition 2 and Definition 4 it immediately follows that all rows of a computation M-matrix are IDs of M and if u, v are two subsequent rows, then $u \vdash_M v$. Therefore, the following fact is only a reformulation of definitions introduced in the second paragraph of this paper.

5. Fact. A word w is accepted by a machine M iff there is an M-matrix $\mathcal{M}$ over w such that $\mathrm{Comp}_M(\mathcal{M})$.

Recall the notion of crossing sequence due to Trachtenbrot, Hennie and Rabin (cf. [1], p. 143). Every computation of a machine M with an input w determines for every bound between two tape squares a crossing sequence $\sigma = q(1) \dots q(n)$. The i-th member of the sequence is the state in which the head crosses the bound for the i-th time. We shall modify the notion so that every member of a crossing sequence will also contain the information about in what direction the bound was crossed.

6. Definition. For a given Q, let $Q^{\rightarrow}$ and $Q^{\leftarrow}$ be disjoint sets of pairwise distinct symbols such that card $Q^{\rightarrow}$ = card $Q^{\leftarrow}$ = card Q. A *crossing sequence* is then every sequence $s_1 \dots s_n \in (Q^{\rightarrow} \cup Q^{\leftarrow})^*$ such that

$$s_i \in Q^{\rightarrow} \Rightarrow s_{i+1} \in Q^{\leftarrow} \quad \text{and} \quad s_i \in Q^{\leftarrow} \Rightarrow s_{i+1} \in Q^{\rightarrow} \quad \text{for every} \quad 1 \leqq i < n\,.$$

A one-one correspondence between $Q^{\rightarrow}$ and Q (and between $Q^{\leftarrow}$ and Q) can be chosen. In the sequel, we shall assume that such underlying correspondences are fixed and $q^{\leftarrow}$ and $q^{\rightarrow}$ will denote the elements corresponding with $q \in Q$ in $Q^{\leftarrow}$ and $Q^{\rightarrow}$, respectively.

7. Definition. Let $\mathcal{M} = (a_{ij})$ be an M-matrix with n columns, $n \geqq 2$. We shall say that $\mathcal{M}$ has

1) *left entry point* i iff the i-th row of $\mathcal{M}$ is of type a and $a_{i+1,1} \in \{q\} \times A$, for some q. $\{q^{\rightarrow}\}$ is then called the *set adjacent to the left entry point* i;

2) *right entry point* i iff the i-th row of $\mathcal{M}$ is of type a and $a_{i+1,n} \in \{q\} \times A$ for some q. $\{q^{\leftarrow}\}$ is then called the *set adjacent to the right entry point* i;

3) *left exit point* i iff $a_{i,1} \in Q \times A$ and the $i+1$-st row is of type a. $\{q^{\leftarrow}; (\{q\} \times \times A \times \{left\}) \cap \delta(a_{i1}) \neq \emptyset\}$ is then called the *set adjacent to the left exit point* i;

4) *right exit point* i iff $a_{i,n} \in Q \times A$ and the $i+1$-st row is of type a. $\{q^{\rightarrow}; (\{q\} \times \times A \times \{right\}) \cap \delta(a_{in}) \neq \emptyset\}$ is then called the *set adjacent to the right exit point* i.

All points introduced sub 1)–4) are *limiting points* of $\mathscr{M}$.

Let $p_1, \ldots, p_k$ are the all limiting points of $\mathscr{M}$, in the natural ordering. We shall say that $\mathscr{M}$ is *normal* iff the following two conditions hold:

a) p_i is a left exit point $\Rightarrow p_{i+1}$ is a left entry point,

b) p_i is a right exit point $\Rightarrow p_{i+1}$ is a right entry point, for $1 \leqq i < k$.

Normal M-matrices apparently reflect the fact that when the head leaves the left end of a tape segment, it cannot come back from the right and vice versa.

8. Definition. Let $\mathscr{M}$ be a normal M-matrix; $l_1, \ldots, l_m$ all its (naturally ordered) left entry and exit points; $r_1, \ldots, r_n$ all its right entry and exit points, again in natural ordering. Let $\sigma_1 = s_1^1, \ldots, s_l^1$ and $\sigma_2 = s_1^2, \ldots, s_k^2$ be crossing sequences. We shall say that σ_1 and σ_2 are *adjacent* to $\mathscr{M}$ (denote $\mathrm{Adj}(\sigma_1, \mathscr{M}, \sigma_2)$ iff

1) $l = m \,\&\, k = n$,

2) s_i^1 is in the set adjacent to l_i for every $1 \leqq i \leqq m$,

3) s_i^2 is in the set adjacent to r_i for every $1 \leqq i \leqq n$.

9. Lemma. Let $\mathscr{M}_1$ and $\mathscr{M}_2$ be M-matrices with the same number of columns and let $\mathrm{Adj}(\sigma_1, \mathscr{M}_1, \sigma_2)$, $\mathrm{Adj}(\sigma_3, \mathscr{M}_2, \sigma_4)$ for some $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. If the last row u of $\mathscr{M}_1$ and the first row v of $\mathscr{M}_2$ are of type p and such that $u \vdash_M v$, then $\mathrm{Adj}(\sigma_1\sigma_3, \mathscr{M}, \sigma_2\sigma_4)$, where

$$\mathscr{M} = \begin{pmatrix} \mathscr{M}_1 \\ \mathscr{M}_2 \end{pmatrix}.$$

Proof. $\mathscr{M}$ is an M-matrix, by the assumption about u and v. The last limiting point of $\mathscr{M}_1$ is an entry point and the first limiting point of $\mathscr{M}_2$ is an exit points, by the same assumption. $\mathscr{M}$ is therefore normal and also the conditions 1)–3) of Definition 8 obviously hold.

In an M-matrix some blocks of a-rows can occur. If the matrix is interpreted as the history of a computation on a tape segment, then the vertical dimension of such a block denotes the number of steps during which the head is continuously out of the tape segment. In the sequel, we often use only the information that the head is absent and we are not interested in how long it is absent. That is why the equivalence relation $\approx$ is introduced.

10. Definition. Define the relation $\sim$ in the set of M-matrices as follows: for any M-matrices $\mathscr{M}$, $\mathscr{N}$ let $\mathscr{M} \sim \mathscr{N}$ iff

$$\mathscr{M} = \begin{pmatrix} \mathscr{M}_1 \\ w \\ w \\ \mathscr{M}_2 \end{pmatrix}, \quad \mathscr{N} = \begin{pmatrix} \mathscr{M}_1 \\ w \\ \mathscr{M}_2 \end{pmatrix}.$$

where $\mathscr{M}_1$ and $\mathscr{M}_2$ are some M-matrices (may be empty) and w is a row of type $\boldsymbol{a}$.

Then $\approx$ define as the equivalence relation generated by $\sim$ (i.e. the reflexive, symmetric and transitive closure of $\sim$).

11. Lemma. Let $\mathrm{Adj}(\sigma_1, \mathscr{M}_1, \sigma_2)$ & $\mathrm{Adj}(\sigma_2, \mathscr{M}_2, \sigma_3)$ & $\sigma_2 \neq \lambda$ (λ will denote the empty sequence). Then there are $\mathscr{M}'_1$, $\mathscr{M}'_2$ such that $\mathscr{M}'_1 \approx \mathscr{M}_1$, $\mathscr{M}'_2 \approx \mathscr{M}_2$ and $\mathrm{Adj}(\sigma_1, \mathscr{M}, \sigma_3)$ for $\mathscr{M} = (\mathscr{M}'_1 \mathscr{M}'_2)$.

Proof. By induction.

I. Let $|\sigma_2| = 1$, e.g. $\sigma_2 = q^{\leftarrow}$ for some q. σ_2 determines the single right entry point of $\mathscr{M}_1$. This point cuts the matrix $\mathscr{M}_1$ into matrices $\mathscr{M}_1^{(1)}$ and $\mathscr{M}_1^{(2)}$

$$\left(\text{i.e. } \mathscr{M}_1 = \begin{pmatrix} \mathscr{M}_1^{(1)} \\ \mathscr{M}_1^{(2)} \end{pmatrix}\right).$$

All rows of $\mathscr{M}_1^{(1)}$ are of type a, by Definition 5. Similarly, σ_2 determines the single left exit point of $\mathscr{M}_2$ and consequently matrices $\mathscr{M}_2^{(1)}$, $\mathscr{M}_2^{(2)}$ such that

$$\mathscr{M}_2 = \begin{pmatrix} \mathscr{M}_2^{(1)} \\ \mathscr{M}_2^{(2)} \end{pmatrix}$$

and $\mathscr{M}_2^{(2)}$ contains rows of type a only. By adding or omitting a-rows matrices $\tilde{\mathscr{M}}_1^{(1)} \approx \mathscr{M}_1^{(1)}$ and $\tilde{\mathscr{M}}_2^{(2)} \approx \mathscr{M}_2^{(2)}$ can be constructed such that $\mathscr{M}' = (\tilde{\mathscr{M}}_1^{(1)} \mathscr{M}_2^{(1)})$ and $\mathscr{M}'' = (\mathscr{M}_1^{(2)} \tilde{\mathscr{M}}_2^{(2)})$ are M-matrices. Moreover, $\mathrm{Adj}(\sigma_1, \mathscr{M}_1^{(2)}, \lambda)$ and $\mathrm{Adj}(\lambda, \mathscr{M}_2^{(1)}, \sigma_3)$. Therefore $\mathrm{Adj}(\sigma_1, \mathscr{M}'', \lambda)$ and $\mathrm{Adj}(\lambda, \mathscr{M}', \sigma_3)$. For the last row w' of $\mathscr{M}'$ and the first row w'' of $\mathscr{M}''$ the condition $w' \vdash_M w''$ holds, by Definitions 7 and 8. This implies $\mathrm{Adj}(\sigma_1, \mathscr{M}, \sigma_3)$ for

$$\mathscr{M} = \begin{pmatrix} \mathscr{M}' \\ \mathscr{M}'' \end{pmatrix}$$

by Lemma 9. Let us define

$$\mathscr{M}'_1 = \begin{pmatrix} \tilde{\mathscr{M}}_1^{(1)} \\ \mathscr{M}_1^{(2)} \end{pmatrix} \quad \text{and} \quad \mathscr{M}'_2 = \begin{pmatrix} \mathscr{M}_2^{(1)} \\ \tilde{\mathscr{M}}_2^{(2)} \end{pmatrix}.$$

Then obviously $\mathscr{M}'_1 \approx \mathscr{M}_1$, $\mathscr{M}'_2 \approx \mathscr{M}_2$ and $\mathscr{M} = (\mathscr{M}'_1 \mathscr{M}'_2)$.

The case $\sigma_2 = q^{\rightarrow}$ is quite analogous.

II. Let $|\sigma_2| > 1$, e.g. $\sigma_2 = q^{\rightarrow}\sigma_2'$ for some q and $|\sigma_2'| \geqq 1$. The first left entry point of $\mathcal{M}_2$ determines matrices $\mathcal{M}_2^{(1)}$, $\mathcal{M}_2^{(2)}$ such that

$$\mathcal{M}_2 = \begin{pmatrix} \mathcal{M}_2^{(1)} \\ \mathcal{M}_2^{(2)} \end{pmatrix},$$

$\mathcal{M}_2^{(1)}$ contains rows of type a only and $\mathrm{Adj}(\sigma_2', \mathcal{M}_2^{(2)}, \sigma_3)$. The first right exit point of $\mathcal{M}_1$ determines matrices $\mathcal{M}_1^{(1)}$, $\mathcal{M}_1^{(2)}$ and crossing sequence $\sigma_1^{(1)}$, $\sigma_1^{(2)}$ such that

$$\mathcal{M}_1 = \begin{pmatrix} \mathcal{M}_1^{(1)} \\ \mathcal{M}_1^{(2)} \end{pmatrix},$$

$\sigma_1 = \sigma_1^{(1)}\sigma_1^{(2)}$, $\mathrm{Adj}(\sigma_1^{(1)}, \mathcal{M}_1^{(1)}, \lambda)$, $\mathrm{Adj}(\sigma_1^{(2)}, \mathcal{M}_1^{(2)}, \sigma_2)$. An M-matrix $\tilde{\mathcal{M}}_2^{(1)} \approx \mathcal{M}_2^{(1}$ such that

$$\mathcal{M}' = (\mathcal{M}_1^{(1)} \tilde{\mathcal{M}}_2^{(1)})$$

is an M-matrix can be formed simply by adding or omitting a-rows. Then $\mathrm{Adj}(\sigma_1^{(1)}, \mathcal{M}', \lambda)$. By induction hypothesis an M-matrix $\mathcal{M}''$ exists such that $\mathrm{Adj}(\sigma_1^{(2)}, \mathcal{M}'', \sigma_3)$. Moreover, $\mathcal{M}''$ is of the form

$$\mathcal{M}'' = (\mathcal{M}_1'' \mathcal{M}_2''),$$

where $\mathcal{M}_1'' \approx \mathcal{M}_1^{(2)}$, $\mathcal{M}_2'' \approx \mathcal{M}_2^{(2)}$.

As the first row of $\mathcal{M}_1^{(2)}$ is of type p and as $\mathcal{M}_1'' \approx \mathcal{M}_1^{(2)}$ w is also the first row of $\mathcal{M}_1''$. It is easy to see that for

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}' \\ \mathcal{M}'' \end{pmatrix}$$

$\mathrm{Adj}(\sigma_1, \mathcal{M}, \sigma_3)$ holds by Lemma 9. At last, let us define

$$\mathcal{M}_1' = \begin{pmatrix} \mathcal{M}_1^{(1)} \\ \mathcal{M}_1'' \end{pmatrix}, \quad \mathcal{M}_2' = \begin{pmatrix} \tilde{\mathcal{M}}_2^{(1)} \\ \mathcal{M}_2'' \end{pmatrix}$$

and the theorem follows, because the case $\sigma_2 = q^{\leftarrow}\sigma_2'$ is analogous.

The preceding lemma will be used in the proof of the converse of the second part of Theorem 1. For this purpose the space requirements of the recognition of the predicate Adj are estimated in the following lemma.

12. Lemma. Let M be a Turing machine ($* \notin Q^{\rightarrow} \cup Q^{\leftarrow} \cup A$). Then the language $L = \{\sigma_1 * \sigma_2 * w;\ \sigma_1, \sigma_2$ are crossing sequences such that there is an M-matrix $\mathcal{M}$ for which $\mathrm{Adj}(\sigma_1, \mathcal{M}, \sigma_2)$ and w is its first row$\}$
can be recognized by a LBA.

Proof. For the notion of linear bounded automaton (LBA) we refer to Hopcroft, Ullman [1]. The reader acquainted with it can immediately see that the following nondeterministic algorithm for recognizing L can be performed by a LBA. w-segment denotes the segment of tape containing the input word w at the beginning of the computation. Let

$$\sigma_1 = s_1^1 \dots s_k^1\,, \quad \sigma_2 = s_1^2 \dots s_l^2$$

```
0: $i := 0$;
   $j := 0$;
   if $w$ is not a row of any $M$-matrix then REJECT;
   if $w$ is a row of type $a$ then go to 1 else simulate $M$ until $M$ leaves the $w$-segment;
   if $M$ leaves the left end of the segment in a state $q_1$ then $i := i + 1$ and go to 4;
   if $M$ leaves the right end of the $w$-segment in a state $q_1$ then $j := j + 1$ and go to 6;
   if $M$ enters a final state without leaving the $w$-segment then go to END;

1: if $s_1^1 = q^{\rightarrow}$ for some $q$ then go to 3 else go to 2;

2: if $s_1^2 = q^{\leftarrow}$ for some $q$ then go to 5 else REJECT;

3: if $s_i^1 \notin Q^{\rightarrow}$ then go to END else (i.e. $s_i^1 = q^{\rightarrow}$ for some $q$)
   beginning on the leftmost square of the $w$-segment in the state $q$, simulate $M$
   until $M$ leaves the w-segment;
   if $M$ leaves the left end of the $w$-segment in a state $q_1$ then $i := i + 1$ and go to 4;
   if $M$ leaves the right end of the $w$-segment in a state $q_1$ then $j := j + 1$ and go to 6;
   if $M$ enters a final state not leaving the $w$-segment then go to END;

4: if $i > k$ then REJECT;
   if $s_i^1 \neq q_1^{\leftarrow}$ then REJECT
   if $i < k$ then $i := i + 1$ and go to 3
            else go to END

5: if $s_j^2 \notin Q^{\leftarrow}$ then go to END else (i.e. $s_j^2 = q^{\leftarrow}$ for some $q$)
   beginning on the rightmost square of the $w$-segment in the state $q$, simulate $M$
     until $M$ leaves the $w$-segment;
   if $M$ leaves the left end of the $w$-segment in a state $q_1$ then $i := i + 1$ and go to 4;
   if $M$ leaves the right end of the $w$-segment in a state $q_1$ then $j := j + 1$ and go to 6;
   if $M$ enters a final state not leaving the $w$-segment then go to END;

6: if $j > l$ then REJECT;
   if $s_j^2 \neq q_1^{\rightarrow}$ then REJECT;
   if $j < l$ then $j := j + 1$ and go to 5
            else go to END

END: if $i = k$ and $j = l$ then ACCEPT
                             else REJECT.
```

13. Lemma. Let M be a nondeterministic Turing machine, f an arithmetic function such that $(\forall n)\,(f(n) \geqq n)$ and let $L(M)$ denote the language recognized by M. If

1) for every $w \in L(M)$ there is an M-matrix $\mathcal{M}$ over w which can be written in the form

$$\mathcal{M} = (\mathcal{M}_{-s}\mathcal{M}_{-s+1}, \ldots, \mathcal{M}_{-1}\mathcal{M}_0\mathcal{M}_1, \ldots, \mathcal{M}_r)$$

such that $\mathcal{M}_0$ is over w and every matrix $\mathcal{M}_i\,(-s \leqq i \leqq r)$ is of type $m \times n_i$ for some m and $n_i \leqq f(|w|)$ and there are crossing sequences $\sigma_{-s}, \sigma_{-s+1}, \ldots, \sigma_0, \sigma_1, \ldots, \sigma_{r+1}$ such that

$$|\sigma_i| \leqq f(|w|)\ (-s \leqq i \leqq r + 1)\,,$$

$$\mathrm{Adj}(\sigma_i, \mathcal{M}_i, \sigma_{i+1})\ (-s \leqq i < r + 1)\,,$$

$$\sigma_s = \sigma_{r+1} = \lambda \quad\text{and}\quad \sigma_i \neq \lambda \quad\text{for}\quad -s < i \leqq r\,;$$

2) there is no such a matrix for any $w \notin L(M)$,
then $L(M) \in S(f)$.

Proof. Let L be the language defined in Lemma 12. To verify the existence of a matrix $\mathcal{M}$ of the described properties for a given $w = x_1 \ldots x_n$ it is sufficient to test gradually whether

$(-s)$ $\sigma_{-s} * \sigma_{-s+1} * b^{n-s} \in L$ for an $n_{-s} \leqq f(n) = f(|w|)$,

. . . .

(-1) $\sigma_{-1} * \sigma_0 * b^{n-1} \in L$ for an $n_{-1} \leqq f(n)$,

(0) $\sigma_0 * \sigma_1 * b^i(q_0, x_1)\, x_2 \ldots x_n b^j \in L$, where q_0 is the initial state of M, $i + j + + n \leqq f(n)$,

(1) $\sigma_1 * \sigma_2 * b^{n_1} \in L$ for an $n_1 \leqq f(n)$,

. . . .

(r) $\sigma_r * \sigma_{r+1} * b^{n_r} \in L$ for an $n_r \leqq f(n)$

This can be done nondeterministically by choosing all σ_i and n_i arbitrarily. Moreover, by Lemma 12, for every $-s \leqq i \leqq r$ the space $3.\,f(n)$ is sufficient to verify the condition (i), provided $|\sigma_i|$, $|\sigma_{i+1}|$, $n_i \leqq f(n)$. For some time it is necessary to keep the input word w in the memory. No other extra memory is necessary and so the space $4.\,f(n)$ will do. Then by linear speed-up (c.f. [1], p. 137) the space $f(n)$ is sufficient.

By Lemma 13, the converse of the second part of Theorem 1 immediately follows.

14. Theorem.* Let f be an arithmetic function such that $(\forall n)\,(f(n) \geqq n)$. Then

$$C(f) = S(f)\,.$$

* The results stated as Theorem 1 and Theorem 14 in this paper were announced in [3]. Theorem 14 was known to R. Freivalds independently [4], as the author has recently been informed.

Proof. I. $S(f) \subseteq C(f)$ by Theorem 1.

II. Let $L \in C(f)$. Then there is an M such that $L(M) = L$ and for every $w \in L$ there is an M-matrix $\mathscr{M}$ satisfying conditions of Lemma 13 and for $w \notin L$ there is no such a matrix. By Lemma 13 then $L \in S(f)$.

4. RELATION TO THE LBA-PROBLEM

Let us list several open problems:

(LBA): Does $DS(id) = S(id)$ hold? (id is the identity function: $id(n) = n$.)
(DS − S): Does $DS(f) = S(f)$ hold for every f such that $f(n) \geqq n$?
(DS − DC): Does $DC(f) = DS(f)$ hold, provided $f(n) \geqq n$?
(DC − C): Does $DC(f) = C(f)$ hold, provided $f(n) \geqq n$?

The problem (LBA) is the well-known long opened LBA problem. By a result of Hartmanis and Hunt III [5] the problems (LBA) and (DS − S) are equivalent, i.e. answers to them are either both positive or both negative. Let us denote this fact thus:

$$(\mathrm{LBA}) \equiv (\mathrm{DS}-\mathrm{S})\,.$$

Theorem 1 and Theorem 14 of this paper imply that the answer to the problem (DS − S) is positive iff the answers to both of the problems (DS − DC) and (DC − C) are positive. Thus symbolically:

$$(\mathrm{DS}-\mathrm{S}) \equiv (\mathrm{DS}-\mathrm{DC})\,\&\,(\mathrm{DC}-\mathrm{C})\,.$$

The nature of the problem (DC − C) resembles the deterministic-versus-nondeterministic nature of the LBA-problem. The problem (DS − DC), on the contrary, concerns deterministic machines only. The study of the relation between deterministic tape and crossing complexities therefore could be of some importance also for the solution of the LBA-problem.

(Received November 27, 1975.)

REFERENCES

[1] J. E. Hopcroft, J. D. Ullman: Formal languages and their relation to automata, Addison-Wesley, 1969.

[2] M. S. Paterson: Tape bounds for time-bounded Turing machines. JCSS *6*, (1972), 116—124.

[3] M. P. Chytil: Crossing-bounded automata and their relation to the LBA-problem. In "Tagung über Automatentheorie und Formale Sprachen, 1974", Universität Dortmund.

[4] R. V. Freivalds, oral communication.

[5] J. Hartmanis, H. B. Hunt III: The LBA Problem and its Importance in the Theory of Computing. TR 73—171, May 1973, Cornell University.

RNDr. Michal Chytil, Matematicko-fyzikální fakulta UK (Faculty of Mathematics and Physics — Charles University), Malostranské nám. 25, 110 00 Praha 1. Czechoslovakia.