About the deterministic simulation of nondeterministic (log n)-tape bounded Turing machines

BURKHARD MONIEN
Universität Dortmund
Abteilung Informatik

1. Introduction

One of the oldest problem in the theory of automata and languages is the so-called LBA problem, that is the question whether deterministic linear bounded automata are as powerful as nondeterministic linear bounded automata. This problem can be formulated also in the terminology of computational complexity. Let us denote by TAPE(f(n)) the class of all languages which are acceptable by deterministic multi-tape Turing machines operating with tape bound f(n). Correspondingly NTAPE(f(n)) is defined taking the nondeterministic Turing machine as the underlying machine model. Then the LBA problem is just the question whether TAPE(n) is equal to NTAPE(n).

It is not difficult to see that equality results which hold for some tape function do hold also for every tape function which grows more rapidely. That means:

Let $g\colon \mathbb{N} \to \mathbb{N}$ be some function such that NTAPE(log n) $\subset$ TAPE(g(log n)) holds. Then NTAPE(f(n)) $\subset$ TAPE(g(f(n))) holds for all functions $f\colon \mathbb{N} \to \mathbb{N}$ such that $f(n) \geq \log n$ for all n.

In particular we have: If NTAPE(log n) = TAPE(log n) then also NTAPE(n) = TAPE(n).

Therefore we can restrict our study to the behaviour of (log n)-tape bounded nondeterministic Turing machines. This has been done before by J. Hartmanis [2] and I.H. Sudborough [8] . First we define a subclass of NTAPE(log n).

Definition: Let C be the class of all languages accepted by nondeterministic one-way one-counter automata. Such an automaton consists of a finite control, an input tape where one head is moving from the left to the right and of a counter. The next move function is nondeterministic.

It is not difficult to see that C $\subset$ NTAPE(log n) because every string accepted by such an automaton is accepted also by a sequence of moves such that the numbers stored by the counter are always lineary bounded by the lenght of the input. Furthermore it is clear that all elements of C are context-free languages. In section 2 we prove the following theorem.

Theorem 1: Let $\alpha \geq 1$ be some rational number. Then NTAPE(log n) $\subset$ TAPE($(\log n)^{\alpha}$) is equivalent to C $\subset$ TAPE($(\log n)^{\alpha}$).

Because of the results of P.M.Lewis, R.E. Stearns and J. Hartmanis [4] we know that every context-free language can be accepted by a deterministic Turing machine with tape bound $(\log n)^2$. Therefore we get the result of W.J. Savitch [6] as a corollary of our theorem.

Corollary 1: $\mathrm{NTAPE}(\log n) \subset \mathrm{TAPE}((\log n)^2)$

Furthermore theorem 1 shows that the simulation of (log n)-tape bounded nondeterministic Turing machines by deterministic Turing machines is simpler (that means that the deterministic Turing machine needs at most the same amount of space) than the deterministic acceptance of context-free languages. As an immediate consequence of theorem 1 we get the following result.

Corollary 2: $\mathrm{NTAPE}(\log n) = \mathrm{TAPE}(\log n) \Leftrightarrow C \subset \mathrm{TAPE}(\log n)$

In order to get an idea of the difference between determinism and nondeterminism, we now consider the usual closure operations. It is obvious that both classes, TAPE(log n) and NTAPE(log n), are closed under union, intersection, concatenation and inverse homomorphism. To the author's knowledge nothing is known about the closure of either of these classes against ε-free homomorphism, but it is easy to show that TAPE(log n) is equal to NTAPE(log n) if TAPE(log n) is closed under ε-free homomorphism. The most interesting operation in this context is Kleene's $*$-operator. The $*$-operator is defined in the following way: Let L be some language. Then $L^* = \{v_1 \ldots v_k \mid k \in \mathbb{N} \cup \{0\}\ ,\ v_i \in L \text{ for all } i = 1,\ldots,k\}$. It is obvious that NTAPE(log n) is closed under the application of the $*$-operator and we show in section 3 that this property is characteristic for the nondeterminism. We prove the following result.

Theorem 2: NTAPE(log n) = TAPE(log n) holds if and only if TAPE(log n) is closed under the application of the $*$-operator.

This theorem makes clear that in the case of the tape function log n the difference between nondeterministic and deterministic machines is just the ability to compute an admissible decomposition of the given input string. We even get the following, more general result.

Theorem 2a: Let $\alpha \geq 1$ be some rational number such that $\mathrm{TAPE}((\log n)^\alpha)$ is closed under the application of the $*$-operator. Then $\mathrm{NTAPE}(\log n) \subset \mathrm{TAPE}((\log n)^\alpha)$.

It is not difficult to see (by means of the methods of [4]) that $\mathrm{TAPE}((\log n)^2)$ is closed under the application of the $*$-operator, and therefore again we get Savitch's result. Furthermore if we could prove an analogon of theorem 2 for any tape function growing faster than log n, then most probably we would get the corresponding analogon of theorem 2a too and this would imply a better result for the deterministic simulation of nondeterministic machines.

The results of this paper are proved by means of transformational methods. These methods have been used extensively by R.V. Book in [1] and in other papers. Further results are proved in [5] .

2. Proof of theorem 1

We use the notion of many-one reducibility as it is defined in recursive function theory (due to D. Knuth [3] we will speak of transformability).

Definition Let $\mathcal{C}$ be a class of functions (on strings).

(i) Let $f: \Sigma^* \to \Gamma$ be a function in $\mathcal{C}$. A set $L_1 \subset \Sigma^*$ is f-transformable to $L_2 \subset \Gamma^*$ if for every $w \varepsilon \Sigma^*$.
$w \varepsilon L_1$ if and only if $f(w) \varepsilon L_2$.

(ii) A class $\mathcal{L}_1$ of sets is $\mathcal{C}$-transformable to a class $\mathcal{L}_2$ of sets if for every $L_1 \varepsilon \mathcal{L}_1$ there exist $L_2 \varepsilon \mathcal{L}_2$ and $f \varepsilon \mathcal{C}$ such that L_1 is f-transformable to L_2.

(iii) A class $\mathcal{L}$ of sets is closed under $\mathcal{C}$-transformabilities if for every set L_1, L_1 is $\mathcal{C}$-transformable to some set $L_2 \varepsilon \mathcal{L}$ implies $L_1 \varepsilon \mathcal{L}$.

The following lemma follows immediately from the definitions above:

Lemma 1: Let $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ be classes of sets and let $\mathcal{C}$ be a class of functions. Suppose that $\mathcal{L}_1$ is $\mathcal{C}$-transformable to $\mathcal{L}_2$ and that $\mathcal{L}_3$ is closed under $\mathcal{C}$-transformabilities. Then $\mathcal{L}_2 \subset \mathcal{L}_3$ implies $\mathcal{L}_1 \subset \mathcal{L}_3$.

The method of trnaformabilities was used explicitly by R.V. Book in [1] and implicitly by J. Hartmanis who showed in [2] that NTAPE(log n) $\subset$ TAPE(log n) is equivalent to $N_3 \subset$ TAPE(log n). N_3 is defined below.

Definition: A k-head two-way automaton consists of a finite control and an input tape where k heads may move independently in both directions (k-head two-way finite automata). The input is placed between two endmarkers ($\dashv$ and $\vdash$). The automaton starts in a distinguished starting state with its k heads on the left endmarker. It accepts the input if it stops in an accepting state. The automaton is called deterministic if its nextmove function is deterministic, otherwise it is called nondeterministic. Let $D_k (N_k), k \varepsilon \mathbb{N}$, be the class of all sets accepted by deterministic (nondeterministic) k-head two-way automata.

It is obvious that $\bigcup_{k \varepsilon \mathbb{N}} N_k$ = NTAPE(log n).

In 1973 I.H. Sudborough [8] improved the result of Hartmanis and showed that $\bigcup_{k \varepsilon \mathbb{N}} N_k \subset$ TAPE(log n) is equivalent to $1\text{-}N_2 \subset$ TAPE(log n), where $1\text{-}N_k$, $k \varepsilon \mathbb{N}$, is the class of all languages accepted by nondeterministic k-head one-way finite automata (that means, the k heads are only allowed to move from the left to the right between the two endmarkers). We will show in this paper that it is also equivalent to consider the problem whether C is contained in TAPE(log n).

This result looks similar to Sudborough's result but it seems that this result can't be proved by using his methods, and the fact that C is a subclass of the context-free languages may be useful to get further results. Furthermore we get our results by using only transformational methods whereas Sudborough uses Savitch's language of threadable mazes [7].

We use a class π of functions which is defined in the following way:

(i) Let Σ be an alphabet, let $\dashv, \vdash$ be elements not in Σ and let k be a natural number. Let $f_{\Sigma,k}: \dashv \Sigma^* \vdash \;\rightarrow\; ((\Sigma \cup \{\dashv, \vdash\})^k)^*$
be the following function. For all $m \in \mathbb{N}$ and all $a_i \in \Sigma$, $i = 1,\ldots,m$,
$f_{\Sigma,k}(\dashv a_1 \ldots a_m \vdash) = \alpha_0\alpha_1\ldots\alpha_{n^k-1}$ where $n = m+2$ and
$\alpha_j = (a_{i_1}, a_{i_2}, \ldots, a_{i_k})$ for $j = i_1 + i_2 n + \ldots + i_k n^{k-1}$ with
$0 \leq i_\nu \leq n-1$ for all $\nu = 1,\ldots,k$. For the sake of simplicity we set $a_0 = \dashv$ and $a_{n-1} = \vdash$.
As an example let us consider the case $k = 3$ and $n = 3$.
Then $f_{\Sigma,3}(a_0a_1a_2) = \alpha_0\alpha_1\ldots\alpha_{26}$, where

$$\begin{array}{lclcll}
\alpha_0 = (a_0,a_0,a_0) & \rightarrow & \alpha_3 = (a_0,a_1,a_0) & \rightarrow & \ldots & \rightarrow \alpha_9 = (a_0,a_0,a_1) \\
\alpha_1 = (a_1,a_0,a_0) & & \alpha_4 = (a_1,a_1,a_0) & & \ldots & \alpha_{10} = (a_1,a_0,a_1) \\
\alpha_2 = (a_2,a_0,a_0) & \lrcorner & \alpha_5 = (a_2,a_1,a_0) & \lrcorner & \ldots & \lrcorner\ \alpha_{11} = (a_2,a_0,a_1)
\end{array}$$

Note that $(\dashv, \dashv, \ldots, \dashv)$ and $(\vdash, \vdash, \ldots, \vdash)$ enclose the new string and don't occur inside, therefore they can be regarded as endmarkers.

(ii) Let d be a natural number and let
$g_{\Sigma,k,d}: \dashv \Sigma^* \vdash \;\rightarrow\; \dashv((\Sigma \cup \{\dashv, \vdash\})^k)^*$ be defined by
$g_{\Sigma,k,d}(\dashv w \vdash) = \dashv (f_{\Sigma,k}(\dashv w \vdash) \cdot f_{\Sigma,k}(\dashv w \vdash)^R)^{d \cdot 1(\dashv w \vdash)^k} \vdash \quad \forall w \in \Sigma^*$.

We denote by π the class of all functions which are defined in (ii).

<u>Lemma 2:</u> $\bigcup_k N_k$ is π-transformable to

$\{L \mid \exists\, L_1 \in C,\ L_2 \in \mathrm{TAPE}(\log n)$ such that $L = L_1 \cap L_2\}$.

<u>Proof:</u> Let $L \subset \Sigma^*$ be an arbitrary element of N_k for some k. We will show that there exists a $d \in \mathbb{N}$ such that $g_{\Sigma,k,d}(L) = \tilde{L} \cap g_{\Sigma,k,d}(\dashv \Sigma^* \vdash)$ where $\tilde{L} \in C$. Therefore we first have to show that $g_{\Sigma,k,d}(\dashv \Sigma^* \vdash) \in \mathrm{TAPE}(\log n)$. This is not difficult to be proved and we refer the reader to [5] where a detailed proof of a slightly stronger result is given.

Now we will construct a nondeterministic 1-counter automaton $\tilde{M}$ which accepts an input string of the form $g_{\Sigma,k,d}(\dashv w \vdash)$ if and only if $\dashv w \vdash \in L$. We don't care about the behavior of $\tilde{M}$ on input strings which are not of this form.

Let M_k be a nondeterministic k-head two-way automaton accepting L. Then there exists a $d \in \mathbb{N}$ such that every computation of M on an input of lenght n needs at most $2 \cdot d \cdot n^k$ moves. In the following let d be this number. M simulates on the input string $g_{\Sigma,k,d}(\dashv w \vdash)$ all the moves performed by M_k on the input string $\dashv w \vdash$. When M is simulating the t-th step, $1 \leq t \leq 2 \cdot d \cdot n^k$, of M_k, then its head is located in the t-th block (a block is a string $f_{\Sigma,k}(\dashv w \vdash)$ or $f_{\Sigma,k}(\dashv w \vdash)^R$, respectively) of $g_{\Sigma,k,d}(\dashv w \vdash)$. Let $i_1, i_2, \ldots, i_k$ be the positions of the k heads of M_k before M_k performs its t-th step. Then the head position of $\tilde{M}$ is given by

$i = i_1 + i_2 n + \ldots + i_k n^{k-1} + tn^k$ if $t \equiv 1 \bmod 2$, and

$i = (t+1)n^k - (i_1 + i_2 n + \ldots + i_k n^{k-1})$ if $t \equiv 0 \bmod 2$.

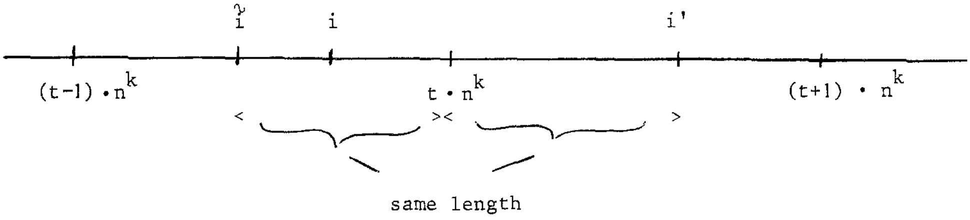
Note that the head of $\tilde{M}$ is located in a block of the form $f_{\Sigma,k}(\dashv w \vdash)$ if $t \equiv 1 \bmod 2$ and in a block of the form $f_{\Sigma,k}(\dashv w \vdash)^R$ if $t \equiv 0 \bmod 2$.

Therefore the i-th symbol of $g_{\Sigma,k,d}(\dashv w \vdash)$ is just $(a_{i_1}, a_{i_2}, \ldots, a_{i_k})$ when $\dashv w \vdash = a_0 \ldots a_{n-1}$. That means that $\tilde{M}$ reads with its single head just the symbols read by the k heads of M_k and so $\tilde{M}$ has all the information necessary to simulate the next move of M_k.

Let us suppose that $\tilde{M}$ is simulating the t-th step of M_k and let i be the head position of $\tilde{M}$. Then M has to move its head to the position $i' = t \cdot n^k + (t \cdot n^k - \tilde{i})$, where

$$\tilde{i} = i + \sum_{j=0}^{k-1} \beta_j n^j \text{ and } \beta_j \in \{-1,0,+1\}, \quad j = 1, \ldots, k,$$

are determined by the moves performed by the k heads of M_k in its t-th step.



$\tilde{M}$ performs the following operations (note that the head of $\tilde{M}$ is not allowed to move to the left).

Its head moves to the right and the number of these moves are stored by its counter. During this process $\tilde{M}$ subtracts all numbers n^j such that $\beta_j = +1$. That means, if $\tilde{M}$ reads a symbol of the form $(\underbrace{\dashv, \dashv, \ldots, \dashv}_{m}, a, \ldots)$, $a \varepsilon \Sigma$, on the input tape, it looks for the greatest $j \leq m$ such that it still has to subtract n^j. Then $\tilde{M}$ moves to the next cell containing a symbol of the form $(\underbrace{\dashv, \dashv, \ldots, \dashv}_{j}, a, \ldots), a \varepsilon \Sigma$. The distance of these two cells is just n^j. During these moves the counter remains unchanged. Therefore the counter stores the number

$(t \cdot n^k - i) - \sum_{j, \beta_j = 1} n^j$ when the head reaches the $(t \cdot n^k)$-th cell.

Now $\tilde{M}$ adds to its headposition all numbers n^j such that $\beta_j = -1$ one after the other beginning with the largest one. Again the movement of the head is controlled by symbols of the form $(\underbrace{\dashv, \dashv, \ldots, \dashv}_{j}, a, \ldots)$, $a \varepsilon \Sigma$. During this process the head

moves to the position $t \cdot n^k + \sum_{j, \beta_j = -1} n^j$. Now $\tilde{M}$ reaches the head position i' by adding the contents of the counter.

$\tilde{M}$ accepts the input string when it notices that M_k reaches a final state. Therefore $\tilde{M}$ accepts $g_{\Sigma,k,d}(\dashv w \vdash)$ if and only if M_k accepts $\dashv w \vdash$. Let $\tilde{L}$ be the language accepted by $\tilde{M}$, then $g_{\Sigma,k,d}(L) = \tilde{L} \cap g_{\Sigma,k,d}(\dashv \Sigma^* \vdash)$. q.e.d.

It is not difficult to see that the class TAPE($(\log n)^\alpha$) is closed under π-transformabilities.

Lemma 3: For any rational number $\alpha \geq 1$ the class TAPE($(\log n)^\alpha$) is closed under π-transformabilities.

Proof: We have to show that $L \varepsilon$TAPE($(\log n)^\alpha$) implies

$f^{-1}(L) = \{w \mid f(w) \varepsilon L\} \varepsilon$TAPE($(\log n)^\alpha$) for all $f \varepsilon \pi$.

Let M be some deterministic $(\log n)^\alpha$-tape bounded Turing machine accepting L. We will define a Turing machine $\tilde{M}$ accepting $f^{-1}(L)$. $\tilde{M}$ simulates on the input w all moves performed by M on the input f(w). Since there exists a $k \varepsilon \mathbb{N}$ such that $l(f(w)) \leq l(w)^k$, $\tilde{M}$ needs not more that $(\log n)^\alpha$ cells to store in each step of the simulation the contents of the working tape of M. Furthermore $\tilde{M}$ has to store the head position of M. This can be done in log n cells. It is clear that $\tilde{M}$ can decode the head position of M in order to read just the symbols necessary to simulate one step of M.

Our transformational lemma 1 implies together with lemma 2 and lemma 3 our first main result.

Theorem 1: Let $\alpha \geq 1$ be some rational number. Then
$NTAPE(\log n) \subset TAPE((\log n)^{\alpha})$ is equivalent to $C \subset TAPE((\log n)^{\alpha})$.

Proof: Because of lemma 1,2,3 the following relation holds:
$NTAPE(\log n) \subset TAPE((\log n)^{\alpha})$
$\Leftrightarrow \mathcal{L} = \{L_1 \cap L_2 \mid L_1 \varepsilon C, L_2 \varepsilon TAPE(\log n)\} \subset TAPE((\log n)^{\alpha}$.
Since $C \subset TAPE((\log n)^{\alpha})$ implies $\mathcal{L} \subset TAPE((\log n)^{\alpha})$ the theorem follows.

3. Proof of theorem 2

Let $L \subset \Sigma^*$ be an arbitrary element of NTAPE(log n). We showed in the proof of lemma 2 that there exist $k,d \varepsilon \mathbb{N}$ and a set $\tilde{L} \varepsilon C$ such that $g_{\Sigma,k,d}(L) = \tilde{L} \cap g_{\Sigma,k,d}(\dashv \Sigma^* \vdash)$. Furthermore we constructed a nondeterministic 1-counter automaton $\tilde{M}$ accepting $\tilde{L}$ whose head is moving one cell to the right in each step and which has the following property:

If its counter stores a number not equal to zero then the next move of $\tilde{M}$ is determined deterministically.

That means that $\tilde{M}$ can act nondeterministerically only if its counter stores zero. Now let $S = \{s_o,\ldots,s_r\}$ be the set of states, $F \subset S$ the set of final states and $s_o \varepsilon S$ the starting state of $\tilde{M}$. We define sets $L_{ij} \subset \Sigma^*$, $0 \leq i,j \leq r$.

Let $\tilde{M}$ start its computation with the state s_i, $v \varepsilon \Sigma^*$ on its input tape and its counter storing zero. v is an element of L_{ij} if and only if the following two condition are fulfilled: (1) After $\tilde{M}$ has read the whole string v, that means after its head has left the input string, the state of $\tilde{M}$ is s_j and its counter stores zero. (2) During the whole computation (except of the first step) the contents of the counter are always greater than zero.

Because of this definition $\tilde{M}$ can act nondeterministically only in its first step when it is accepting an element of L_{ij}. Therefore each L_{ij}, $0 \leq i,j \leq r$, is a finite union of elements of C_D, where C_D is the class of all languages accepted by deterministic one-way one-counter automata. This implies $L_{ij} \varepsilon TAPE(\log n)$ for all $0 \leq i,j \leq r$, because $C_D \subset TAPE(\log n)$ and TAPE(log n) is closed under union.

Now let $w \varepsilon \Sigma^*$ be an arbitrary element of $\tilde{L}$. Then there exists a decomposition $w = v_1 v_2 \ldots v_t$ such that the contents of the counter of $\tilde{M}$ are zero if and only if the head of $\tilde{M}$ is located on the first symbol of one of the v_i, $i \varepsilon \{1,\ldots,t\}$. This shows us that $w \varepsilon \tilde{L}$ holds if and only if there exist a $t \varepsilon \mathbb{N}$ and $i_1,\ldots,i_t \varepsilon \{0,\ldots,r\}$

such that $w \varepsilon L_{oi_1} \circ L_{i_1 i_2} \circ \dots \circ L_{i_{t-1} i_t}$ and $s_{i_t} \varepsilon F$

We set now

$$L^k_{ij} = \bigcup_{t \varepsilon \mathbb{N}} \bigcup_{i_1,\dots,i_t \varepsilon \{0,\dots,k-1\}} L_{ij_1} \circ L_{i_1 i_2} \circ \dots \circ L_{i_t j} \ .$$

The L^k_{ij} are also defined by the wellknown recursive formula

$$L^o_{ij} = L_{ij}$$
$$L^{k+1}_{ij} = L^k_{ij} \cup L^k_{i,k+1} \circ \left(L^k_{k+1,k+1}\right)^{*} \circ L^k_{k+1,j}$$

Intuitively L^k_{ij} consists of all words which lead from the state s_i to the state s_j such that during this computation only states s_ν with $0 \leq \nu \leq k-1$ are reached.

We have seen above that $\tilde{L} = \bigcup_{j, s_j \varepsilon F} L^r_{oj}$.

Now let $\alpha \geq 1$ be some rational number such that TAPE($(\log n)^\alpha$) is closed under the application of the $*$-operator. We have shown already that $L_{ij} \varepsilon$TAPE(log n) for all $i,j \varepsilon \{0,\dots,r\}$ and therefore the recursion formula above and the closure of TAPE(log n) against π-transformabilities (Lemma 3) lead to the following conclusions:

$$L_{ij} \varepsilon \text{TAPE}(\log n) \qquad \forall i,j \varepsilon \{0,\dots,r\}$$
$$\Rightarrow L^k_{ij} \varepsilon \text{TAPE}((\log n)^\alpha) \qquad \forall i,j,k \varepsilon \{0,\dots r\}$$
$$\Rightarrow \tilde{L} = \bigcup_{j, s_j \varepsilon F} L^r_{oj} \varepsilon \ \text{TAPE}(\log n)^\alpha)$$
$$\Rightarrow g_{\Sigma,k,d}(L) = \tilde{L} \cap g_{\Sigma,k,d}(\dashv \Sigma^* \vdash) \varepsilon \ \text{TAPE}((\log n)^\alpha).$$
$$\Rightarrow L \varepsilon \text{TAPE}((\log n)^\alpha)$$

Because L was arbitrarely chosen this completes the proof of theorem 2 and theorem 2a.

References

1. Book, R.V.: On the structure of complexity classes, 2 nd Colloquium: Automata, Languages and Programming, 1974, 437-445.

2. Hartmanis, J., On Non-Determinacy in Simple Computing Devices, Acta Informatica 1 (1972), 336-344.

3. Knuth, D.E., Postscript about NP-hard problems, SIGACT News 6, 2 (1974), 15-16.

4. Lewis, P.M. Stearns, R.E. and Hartmanis, J., Memory bounds for recognition of context-free and context-sensitive languages, IEEE conf.rec. 7-th ann. symp. Switch.Cir.Th.Log. Des., 1965, 191-202.

5. Monien, B., Transformational Methods and their Application to Complexity Problems, submitted for publication

6. Savitch, W.J., Relationships between nondeterministic and deterministic tape complexity, J. Comp. Syst. Sci. 4 (1970), 177-192.

7. Savitch, W.J. Maze Recognizing Automata and Nondeterministic Tape Complexity, J. Comp. Syst. Sci. 7 (1973), 389-403.

8. Sudborough, I.H., On Tape-Bounded Complexity Classes and Multi-Head Finite Automata, IEEE conf. rec. 14-th ann.symp. Switch.Aut.Th., 1973, 138-144.