数理解析研究所講究録
第 833 巻 1993 年 175-185

A note on one-way multicounter machines and cooperating systems of one-way finite automata

王 躍 (Yue Wang), 井上 克司 (Katsushi Inoue) and 高浪 五男 (Itsuo Takanami)

Department of Computer Science and Systems Engineering,

Faculty of Engineering, Yamaguchi University,

Ube, 755 Japan

1. Introduction and preliminaries

This note is divided in two parts. The first part gives a hierarchy result on *one-way [on-line] multicounter machines* (1MCM's) with polynomial time or space bound, and the second part explores the relationships between 1MCM's and *cooperating systems of one-way finite automata* (CS-1FA's).

A *multicounter machine* consists of a finite state control, a reading head which reads the input from the input tape and a finite number of counters. We can regard a counter as an arithmetic register containing an integer which may be positive or zero. In one step, a multicounter machine may increase or decrease a counter by 1. The action of the machine is determined by the input symbol currently scanned, the state of the machine and the sign of each counter: positive or zero. The machine starts with all counters empty and accepts if it enters a final state and halts. (The reader is referred to [3, 4] for the formal definition of a multicounter machine.)

We assume in this note that all our machines have endmarkers (¢, \$) on the input tape and never fall off the input tape beyond endmarkers. One-way [on-line] machines read the input tape from left to right and can enter accepting states only when reading the right endmarker \$.

A deterministic machine M accepts in time $T(n)$ if each input w accepted by M is accepted within $T(|w|)$ steps.[1] A nondeterministic machine M accepts in time $T(n)$ if for each input w accepted by M there is a computation of M on w which accepts in at most $T(|w|)$ steps. A deterministic machine M accepts in space $S(n)$ if for each input w accepted by M, each counter of M requires space not exceeding $S(|w|)$. A nondeterministic machine

[1] For a word w, $|w|$ is the length of w.

M accepts in space $S(n)$ if for each input w accepted by M there is a computation of M on w in which each counter of M requires space not exceeding $S(|w|)$.

For each $r \geq 1$, let 1DCM(r)-Time($T(n)$) (1NCM(r)-Time($T(n)$)) denote a 1DCM(r) (1NCM (r)) accepting in time $T(n)$.

For each $r \geq 1$, let 1DCM(r)-Space($S(n)$) (1NCM(r)-Space($S(n)$)) denote a 1DCM(r) (1NCM (r)) accepting in space $S(n)$.

We denote by $\mathcal{L}$[1DCM(r)-Time($T(n)$)] the class of languages accepted by 1DCM(r)-Time ($T(n)$)'s , and $\mathcal{L}$[1NCM(r)-Time($T(n)$)], $\mathcal{L}$[1DCM(r)-Space($S(n)$)], and so forth have analogous meanings.

Many investigations about the classes of languages accepted in polynomial time or space by multicounter machines have been made [3, 4]. It was shown in [4] that for each $s > 1$ and each $r \geq 1$, $\mathcal{L}$[1DCM(r)-Time(n^s)] ($\mathcal{L}$[1NCM(r)-Time(n^s)]) $\subsetneq$ $\mathcal{L}$[1DCM($r + s + 1$)-Time(n^s)] ($\mathcal{L}$[1NCM($r + s + 1$)-Time(n^s)]), where $\mathcal{L}$[1DCM(r)-Time(n^s)] ($\mathcal{L}$[1NCM(r)-Time(n^s)]) denotes the class of languages accepted in time n^s by *one-way deterministic (nondeterministic) r-counter machines,* 1DCM(r)'s (1NCM(r)'s). As far as we know, it is unknown whether for each X$\in$ {D,N} and for each $r \geq 1$, $s > 1$, 1XCM(r)'s accepting in time (or space) n^s are less powerful than 1XCM($r + 1$)'s in the same time (or space) bound. [For time (or space) n, it was shown in [3] that 1DCM(r)'s (1NCM(r)'s) are less powerful than 1DCM($r + 1$)'s (1NCM($r + 1$)'s).] In the first part of this note, we will give an affirmative answer to this question.

Recently, several properties of CS-1FA's as recognizers were investigated in [3]. A CS-1FA is a one-dimensional version of cooperating systems of two-dimensional finite automata (CS-2-FA's) [1, 2, 8] (where the maze and labyrinth search problems for CS-2-FA's were studied).

The cooperating systems of finite automata may be considered as one of the simplest models of parallel computation: there are more than one finite automata and an input tape where these finite automata operate simultaneously (in parallel) and can communicate with each other on the same cell of the input tape. More precisely, A *cooperating system of k finite automata*, $M = (\mathrm{FA}_1, \mathrm{FA}_2, \cdots, \mathrm{FA}_k)$, consists of k finite automata $\mathrm{FA}_1, \mathrm{FA}_2, \cdots, \mathrm{FA}_k$, and a read-only input tape where these finite automata independently (in parallel) work step by step. Each step is assumed to require exactly one time for its completion. Those finite automata whose input heads scan the same cell of the input tape can communicate with each other, that is, every finite automaton is allowed to know the internal states of other finte automata on the cell it is scanning at the moment. The system M starts with each FA_i on the left endmarker ¢ in its initial state and accepts the input tape if each FA_i enters an accepting state and halts (when reading the right endmarker \$ of the input tape). (The reader is referred to [11] for the formal definition of a cooperating system of [one-way] finite automata.)

For each $k \geq 1$, we denote a *cooperating system of k one-way deterministic (nondeterministic) finite automata* by CS-1DFA(k) (CS-1NFA(k)). In the second part of this note, we continue to investigate the properties of this model. It is shown that $\mathcal{L}$[CS-1DFA(2)] ($\mathcal{L}$[CS-1NFA(2)]) $=$ $\mathcal{L}$[1DCM(1)] ($\mathcal{L}$[1NCM(1)]). It is also shown that for each $k \geq 2$, $\mathcal{L}$[1DCM(k)-Time(n)] $\subsetneq$ $\mathcal{L}$[CS-1DFA($k+1$)] and $\mathcal{L}$[CS-1DFA($k+1$)] ($\mathcal{L}$[CS-1NFA($k+1$)]) $\subsetneq$ $\mathcal{L}$[1DCM(k)-Time(cn)] ($\mathcal{L}$[1NCM(k)-Time(cn)]), where c is a positive constant.

2. Hierarchies based on the number of counters for 1MCM's with polynomial time or space dound

It seems not so easy to find a particular language L for proving that $L \in \mathcal{L}$[1DCM($r+1$)-Time(n^s)] ($\mathcal{L}$[1NCM($r+1$)-Time(n^s)]) $-$ $\mathcal{L}$[1DCM(r)-Time(n^s)] ($\mathcal{L}$[1NCM(r)-Time(n^s)]) or $L \in \mathcal{L}$[1DCM($r+1$)-Space(n^s)] ($\mathcal{L}$[1NCM($r+1$)-Space(n^s)]) $-$ $\mathcal{L}$[1DCM(r)-Space(n^s)] ($\mathcal{L}$[1NCM (r)-Space(n^s)]) for $s > 1$ and $r \geq 1$. We will use another approach to derive the desired results.

Let $L(1) = \{0^i 2 0^i \mid i \geq 1\}$, and for each $k \geq 1$, let $L(k+1) = \{0^i 1 w 1 0^i \mid i \geq 1 \ \& \ w \in L(k)\}$. Given $s, k \geq 1$, let $f_s(k)$ ($f'_s(k)$) denote the minimum number of counters required for deterministic 1MCM's to accept $\{a\}^*L(k)\{a\}^*$ in time n^s (in space n^s), and $g_s(k)$ ($g'_s(k)$) denote the minimum in n^s (in time n^s), and $g_s(k)$ ($g'_s(k)$) denote the minimum number of counters required for nondeterministic 1MCM's to accept $\{a\}^*L(k)\{a\}^*$ in time n^s (in space n^s). Furthermore, for each $r, s \geq 1$, let $\mathrm{D}_s(r) = \max\{k \mid f_s(k) = r\}$, $\mathrm{N}_s(r) = \max\{k \mid g_s(k) = r\}$, $\mathrm{D}'_s(r) = \max\{k \mid f'_s(k) = r\}$ and $\mathrm{N}'_s(r) = \max\{k \mid g'_s(k) = r\}$.

Lemma 2.1. For each $r, s \geq 1$, $\mathrm{D}'_s(r) \leq \mathrm{N}'_s(r) \leq r \cdot s$, and thus $\mathrm{D}_s(r) \leq \mathrm{N}_s(r) \leq r \cdot s$.

Proof: It follows from the definitions that $\mathrm{D}_s(r) \leq \mathrm{N}_s(r)$ and $\mathrm{D}'_s(r) \leq \mathrm{N}'_s(r)$ for each $r, s \geq 1$. We below establish by a contradiction that $\mathrm{N}'_s(r) \leq r \cdot s$ for each $r, s \geq 1$.

Suppose that there exists some 1NCM(r)-Space(n^s) M accepting $\{a\}^*L(r \cdot s + 1)\{a\}^*$. For each $m \geq 1$, let

$$V(m) = \{a^j 0^{i_1} 1 0^{i_2} 1 \cdots 1 0^{i_{r\cdot s+1}} 2 0^{i_{r\cdot s+1}} 1 0^{i_{r\cdot s}} 1 \cdots 1 0^{i_1} a^j \mid 1 \leq i_1, i_2, \cdots, i_{r\cdot s+1} \leq m \;\&\; (j + i_1 + i_2 + \cdots + i_{r\cdot s+1}) = (r \cdot s + 1)m\}.$$

Clearly, $V(m) \subseteq \{a\}^*L(r \cdot s + 1)\{a\}^*$ and $|V(m)| = m^{r\cdot s+1}$.[2] With each $w \in V(m)$, we associate one fixed accepting computation, $c(w)$, of M on w which accepts in space $|w|^s$. Since the number of distinct *memory configurations* of M just after reading the symbol 2 during $c(x2x^R)$ for words $x2x^R \in V(m)$ cannot exceed $O(m^{r\cdot s})$,[3,4] it follows that for large m, there exist two different words $x2x^R$, $y2y^R \in V(m)$ such that the memory configration of M just after reading 2 during $c(x2x^R)$ is the same as that of M after reading the symbol 2 during $c(y2y^R)$. Clearly, from $c(x2x^R)$ and $c(y2y^R)$, we can construct an accepting computation (in space $|x2y^R|^s$) of M on $x2y^R$. This is a contradiction, because $x2y^R$ is not in $\{a\}^*L(r \cdot s + 1)\{a\}^*$. Thus the lemma follows. □

---

[2]For a finite set A, $|A|$ denotes the number of elements in A.

[3]A memory configuration of M is an $(r+1)$-tuple $(q, c_1, \cdots, c_r)$, where q is the current internal state of M and c_i is the contents of the i-th counter of M for $1 \leq i \leq r$,

[4]For a word w, w^R denotes the reversal of word w.

Theorem 2.1. For each $r, s \geq 1$ and each X$\in$ {D,N}, $\mathcal{L}$[1XCM(r)-Time(n^s)] $\subsetneq$ $\mathcal{L}$[1XCM(r+1)-Time(n^s)].

Proof: For each X$\in$ {D,N}, let M be a 1XCM(r)-Time(n^s) accepting $\{a\}^*L(\mathrm{X}_s(r))\{a\}^*$. By Lemma 2.1, $\mathrm{X}_s(r) \leq r \cdot s$. We conside a 1XCM($r+1$)-Time($n^s$) M' which acts as follows.

Suppose that an input word

$$a^p 0^q 1 w 1 0^{q'} a^{p'},$$

where $w \in \{0^{i_1}10^{i_2}1 \cdots 10^{i_{\mathrm{X}_s(r)}}20^{i'_{\mathrm{X}_s(r)}}10^{i'_{\mathrm{X}_s(r)-1}}1 \cdots 10^{i'_1} \mid \forall j(1 \leq j \leq \mathrm{X}_s(r))[i_j, i'_j \geq 1]\}$ and $p, p' \geq 0$, $q, q' \geq 1$. [Input words in the form different from the above can be easily rejected by M'.] M' simulates the action of M on w by using its r counters, and checks by using the remaining counter whether $q = q'$. M' enters an accepting state only if it finds out that (1) M accepts w (i.e. $w \in L(\mathrm{X}_s(r))$) and (2) $q = q'$. Noting that for each $w \in L(\mathrm{X}_s(r))$ and each $w' = a^p 0^q 1 w 1 0^q a^{p'} \in \{a\}^*L(\mathrm{X}_s(r)+1)\{a\}^*$, $|w|^s + p + p' + 2(q+1) \leq |w'|^s$, it will be obvious that M' accepts $\{a\}^*L(\mathrm{X}_s(r)+1)\{a\}^*$ in time n^s. From this and the fact that $\{a\}^*L(\mathrm{X}_s(r)+1)\{a\}^* \notin \mathcal{L}$[1XCM($r$)-Time($n^s$)], it follows that $\{a\}^*L(\mathrm{X}_s(r)+1)\{a\}^* \in \mathcal{L}$[1XCM($r+1$)-Time($n^s$)] $-$ $\mathcal{L}$[1XCM(r)-Time(n^s)]. $\square$

Using a similar technique, we can get the following theorem.

Theorem 2.2. For each $r, s \geq 1$ and each X$\in$ {D,N}, $\mathcal{L}$[1XCM(r)-Space(n^s)] $\subsetneq$ $\mathcal{L}$[1XCM ($r+1$)-Space(n^s)].

3. Relationship between 1MCM's and CS-1FA's

In this section, we establish a relation between 1MCM's and CS-1FA's. We first show that CS-1DFA(2)'s (CS-1NFA(2)'s) and 1DCM(1)'s (1NCM(1)'s) are equivalent in accepting power.

Theorem 3.1. (1) $\mathcal{L}$[CS-1DFA(2)]=$\mathcal{L}$[1DCM(1)], and (2) $\mathcal{L}$[CS-1NFA(2)]=$\mathcal{L}$[1NCM(1)].

Proof: (1) Let M be a 1DCM(1) with s internal states. We will construct a CS-1DFA(2) $M' = (\mathrm{FA}_1, \mathrm{FA}_2)$ to simulate M. If a finite automaton moves its input head one cell to the right every m steps, we say that the *speed* of its input head is $1/m$. M' acts as follows:

1. FA_1 and FA_2 store the internal state of M in their finite controls.
2. For each cell of the input tape:
   (a) If M reaches the cell with the memory configuration (q_0, c_0), $c_0 \leq s$, then
      (i) if M leaves the cell with the contents of the counter $s + c$ $(1 \leq c < s)$, then FA_2 moves at speed $1/(1 + c)$ on the cell and FA_1 moves at speed 1 on the cell;
      (ii) otherwise FA_1 (FA_2) simulates the action of M on the cell, and if the contents of the counter of M on the cell exceeds $2s$, then FA_1 (FA_2) rejects the input tape (because M enters a loop, that is, M never leaves the cell). In this case, FA_1 and FA_2 are on the same cell.
   (b) If M reaches the cell with the memory configuration (q_0, c_0), $c_0 > s$, (in this case, FA_1 and FA_2 are on the different cells) then
      (i) if M leaves the cell with memory configuration (q_x, c_0) in s steps, then FA_1 and FA_2 move at speed 1 on the cell;
      (ii) if M leaves the cell with memory configuration $(q_x, c_0 + \delta)$, $1 \leq \delta < s$, in s steps, then FA_2 moves at speed $1/(\delta + 1)$ on the cell and FA_1 moves at speed 1 on the cell;
      (iii) if M leaves the cell with memory configuration $(q_x, c_0 - \delta)$, $1 \leq \delta < s$, in s steps, then FA_1 moves at speed $1/(\delta + 1)$ on the cell and FA_2 moves at speed 1 on the cell;

(iv) otherwise there is a sequence of memory configurations, $(q_0, c_0), (q_1, c_1), \cdots,$ (q_j, c_j), $j \leq s$, of M on the cell such that $q_i = q_j$ for some i $(0 \leq i < j)$. If $c_i \leq c_j$ (it means that M enters a loop), then FA_1 (FA_2) rejects the input tape; If $c_i > c_j$, then FA_1 stays on the cell until FA_2 reaches the cell.

Note that (A) if the contents of the counter of M exceeds s when M leaves a cell of the input tape, then M' stores it by using the difference between the times at which FA_1 and FA_2 leave the cell, and (B) otherwise M' can simulate the action of M using the finite control. It is easy to verify that M' is able to simulate M. (1) of the theorem follows from this and (2) of Lemma 3.1 below.

(2) It is shown in [5] that every 1NCM(1) is equivalent to some 1NCM(1)-Time(n) (i.e. some 1NCM(1) which accepts in real-time). Let M be a 1NCM(1)-Time(n). We will construct a CS-1NFA(2) $M' = (FA_1, FA_2)$ to simulate M. M' acts as follows:

1. FA_2 moves at the same speed 1/2 on each cell of the input tape.
2. FA_1 stores the internal state of M in its finite control.
3. For each cell of the input tape:
   (a) If M does not change the counter on the cell, then FA_1 moves at speed 1/2 on the cell.
   (b) If M increases the counter by 1 on the cell, then FA_1 moves at speed 1 on the cell.
   (c) If M decreases the counter by 1 on the cell, then FA_1 moves at speed 1/3 on the cell.

So the contents of the counter of M on each cell of the input tape corresponds to the difference between the times at which FA_1 and FA_2 leave the cell. It is easy to verify that M' is able to simulate M. (2) of the theorem follows from this and (2) of Lemma 3.1 below.

□

Remark 3.1. If a problem is undecidable for CS-1DFA(2)'s (CS-1NFA(2)'s), then it is undecidable for CS-1DFA(k)'s (CS-1NFA(k)'s), for all $k \geq 2$. From this observation and the fact [9, 10] that the containment problem is undecidable for 1DCM(1)'s and the equivalence and universe problems are undecidable for 1NCM(1)'s, it follows by Theorem 3.1 that for all $k \geq 2$, the containment problem is undecidable for CS-1DFA(k)'s, and the equivalence and universe problems are undecidable for CS-1NFA(k)'s.

Remark 3.2. It is well known [6] that context-free languages over a one-letter alphabet are regular (so are the lanugages in $\mathcal{L}$[1NCM(1)]). From Theorem 3.1, it follows that the languages in $\mathcal{L}$[CS-1NFA(2)] over a one-letter alphabet are regular. On the other hand, we can easily prove that there exists a nonregular language over a one-letter alphabet (e.g, $\{0^{2^n} | n \geq 1\}$) in $\mathcal{L}$[CS-1DFA(3)]. Hence it follows that over a one-letter alphabet, CS-1DFA(2)'s (CS-1NFA(2)'s) are less powerful than CS-1DFA(3)'s (CS-1NFA(3)'s). It is unknown whether over a one-letter alphabet, CS-1DFA(k)'s (CS-1NFA(k)'s) are less powerful than CS-1DFA($k+1$)'s (CS-1NFA($k+1$)'s) for $k \geq 3$.

Remark 3.3. It is an important open problem in the computing theory whether the classes of languages accepted by determinstic and nondeterministic $L(n)$ tape-bounded Turing machines are the same for $L(n) \geq \log(n)$. Combining Theorem 3.1 with the result in [7], we can give another possibility to investigate the above problem. That is, $\mathcal{L}$[CS-1NFA(2)] is contained in the class of languages accepted by deterministic $\log(n)$ tape-bounded Turing machines if and only if the classes of languages accepted by determinstic and nondeterministic $L(n)$ tape-bounded Turing machines are the same for $L(n) \geq \log(n)$.

Lemma 3.1. For each $k \geq 1$, (1) every 1DCM(k)-Time(n) can be simulated by a CS-1DFA($k+1$), and (2) every CS-1DFA($k+1$) (CS-1NFA($k+1$)) can be simulated by a

1DCM(k)-Time(cn) (1NCM(k)-Time(cn)), where c is some positive constant.

Proof: (1) The proof is very similar to that of (2) of Theorem 3.1, and we leave it to the reader.

(2) For each X$\in$ {D,N} and each $k \geq 1$, let $M = (\text{FA}_1, \text{FA}_2, \cdots, \text{FA}_{k+1})$ be a CS-1XFA($k+1$). We will construct a 1XCM(k)-Time(cn) M' to simulate M, where c is some positive constant dependent only on M. Let $c_1, c_2, \cdots, c_k$ denote k counters of M'. M' acts as follows:

1. M' stores the internal states of FA_1, FA_2, $\cdots$, FA_{k+1} in its finite control.
2. For each cell of the input tape:
   (a) M' stores in its finite control the internal state of each FA_i ($1 \leq i \leq k+1$) when FA_i leaves the cell, and the order $\langle t_1, t_2, \cdots, t_{k+1}\rangle$ in which FA_1, FA_2, $\cdots$, FA_{k+1} leave the cell subsequently (i.e., FA_{t_1} firstly leaves the cell, FA_{t_2} secondly leaves the cell, and so on).[5]
   (b) Furthermore, for each i ($1 \leq i \leq k$), the interval between the times at which FA_{t_i} and $\text{FA}_{t_{i+1}}$ leave the cell is stored by counter c_i.

It was shown in [11] that if M accepts its input tape, it can do so in linear time. Thus, it is easy to verify that M' can simulate M. □

Lemma 3.2. (1) $\mathcal{L}$[CS-1DFA(2)] $- \cup_{1\leq k<\infty}\mathcal{L}$[1DCM($k$)- Time($n$)] $\neq \emptyset$, and (2) $\mathcal{L}$[1DCM (2)-Time(cn)] $- \cup_{1\leq k<\infty}\mathcal{L}$[CS- 1NFA($k$)]$\neq \emptyset$ for some positive constant c.

Proof: (1) It is shown in [3] that $L_1 = \{0^p1^m \mid p \geq m \geq 1\}^*$ is not in $\cup_{1\leq k<\infty}\mathcal{L}$[1DCM($k$)-Time($n$)]. On the other hand, it is easy to prove that L_1 can be accepted by some CS-

[5]If FA_{i_1}, FA_{i_2}, $\cdots$, FA_{i_r} ($1 \leq i_1 < i_2 < \cdots < i_r \leq k+1$) leave the cell simultaneously, we refer the order on them as $\langle i_1, i_2, \cdots, i_r\rangle$

1DFA(2).

(2) For a word w in $\{1\}\{0,1\}^*$, let $n(w)$ be the integer represented by w as a binary number. It is shown in [4] that $L_2 = \{w20^{n(w)} \,|w \in \{1\}\{0,1\}^*\}$ is accepted by a 1DCM(2)-Time(cn), where c is a positive constant. On the other hand, using the technique in the proof of (2) of Lemma 2 in [11], we can prove that L_2 is not in $\cup_{1\leq k<\infty}\mathcal{L}$[CS-1NFA($k$)]. □

Lemma 3.3. For each $k \geq 1$: (1) $\mathcal{L}$[1DCM(k)-Time(n)] $-$ $\mathcal{L}$[CS-1NFA(k)] $\neq \emptyset$, and (2) $\mathcal{L}$[CS-1DFA($k+1$)] $-$ $\mathcal{L}$[1NCM($k-1$)-Time(cn)] $\neq \emptyset$ for any positive constant c.

Proof: It is obvious that $\{a\}^*L(k)\{a\}^*$ (defined in Section 2) is accepted by some 1DCM(k)-Time(n), but not accepted by any 1NCM($k-1$)-Time(cn) (c is positive constant). Furthermore, by using the technique in the proof of Lemma 2 in [11], we can prove that $\{a\}^*L(k)\{a\}^*$ is accepted by some CS-1DFA($k+1$), but not accepted by any CS-1NFA(k). Hence the lemma follows. □

We get the following theorem from Lemmas 3.1 and 3.2, and we know by Lemma 3.3 that this result cannot be "tightened".

Theorem 3.2. For each $k \geq 2$, (1) $\mathcal{L}$[1DCM(k)-Time(n)] $\subsetneq$ $\mathcal{L}$[CS-1DFA($k+1$)], and (2) $\mathcal{L}$[CS-1DFA(k+1)] ($\mathcal{L}$[CS-1NFA(k+1)]) $\subsetneq$ $\mathcal{L}$[1DCM(k)-Time(cn)] ($\mathcal{L}$[1NCM(k)-Time(cn)]), where c is a positive constant.

REFERENCES

1. M. Blum and C. Hewitt, Automata on a two-dimensional tape, in *IEEE Symp. Switching Automata Theory,* (1967)155–160.

2. M. Blum and D. Kozen, On the power of the compass, in *Proceedings of the* **19th** *Annual Symposium on*

Foundations of Computer Science, (1978).

3. P. C. Fischer, A. R. Meyer and A. L. Rosenberg, Counter machines and counter languages, in *Math. Systems Theory* **2** (1968)265–283.

4. S. A. Greibach, Remarks on the complexity of nondeterministic counter languages, in *Theoret. Comput. Sci.* **1** (1976)269–288.

5. S. A. Greibach, Erasable context-free languages, in *Information and Control* **29** (1975)301–326.

6. Ginsburg, S. and H. G. Rice, Two families of languages related to ALGOL, in *J. ACM* **9** (1962)350–371.

7. Z. G, Some open problems in the theory of computation as questions about two-way deterministic pushdown automaton languages, in *Math. Systems Theory* **10** (1977)211–228.

8. A. Hemmerling, Normed two-plane trapes for finite systems of cooperating compass automata, in *EIK* **23(8/9)**, (1978)453–470.

9. O. H.Ibarra, Restricted one-counter machines with undecidable universe problems, in *Math. Systems Theory* **13**, (1979)181–186.

10. L. G. Valiant, Deterministic one-counter automata, in *J. Computer and System Sciences* **10**, (1975)340–350.

11. Y. Wang, K. Inoue and I. Takanami, Some properties of cooperating systems of one-way finte automata (in Japanese), in *IEICE Jan. Trans.* **D-I No.7**, (1992)391–399.