KYBERNETIKA — VOLUME 22 (1986), NUMBER 2

# HIERARCHY OF REVERSAL BOUNDED ONE-WAY MULTICOUNTER MACHINES

JURAJ HROMKOVIČ

We consider one-way partially blind multicounter machines and one-way multicounter machines. We study the reversal bounded version of these machines, where for the input words of the length $n$ the number of reversals is bounded by a function $f(n)$. It is established that time and nondeterminism, as the resources, cannot compensate for a substantial decrease of the number of reversals allowed. Several hierarchy results are consequences of this result.

## 1. INTRODUCTION

A multicounter machine is a multipushdown machine whose pushdown stores operate as counters, i.e. have a single-letter alphabet. Unrestricted multicounter machines accept all recursively enumerable sets [4]. So far various types of restricted multicounter machines have been considered to define proper subclasses [1, 3, 4, 5, 6, 8, 9, 10]. In this paper we shall study one-way deterministic and nondeterministic multicounter machines with reversal number restriction, and one-way partially blind multicounter machines introduced in [5]. The reversal number bound is considered as a function $f(n)$ of the input word length. It is established that time, nondeterminism, and counters, as the resources, cannot compensate for a substantial decrease of the number of reversals allowed. Several hierarchy results are consequences of this result.

This paper consists of four sections. The basic definitions are given in Section 2. The main theorem and its consequences are formulated in Section 3, and the proof of the main theorem is given in Section 4.

## 2. DEFINITIONS

Let us first informally define one-way multicounter machines, and one-way partially blind multicounter machines. The formal definitions can be found in [4, 5].

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 regard a counter as an arithmetic register containing an integer which may be positive or zero. In one step, a multicounter machine may increment or decrement a counter by 1. The action or the choice of actions 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 reaches a final state with all counters empty. The class of one-way multicounter machines without time limitation will be denoted by ***COUNTER***, the deterministic version by ***DCOUNTER***. The class of one-way multicounter machines working in quasirealtime (for each machine there exists such a constant $d$ that the length of each part of any computation, in which the reading head is stationary, is bounded by $d$) will be denoted by ***QR-COUNTER***, the deterministic version by ***QR-DCOUNTER***.

A one-way partially blind multicounter machine is a one-way multicounter machine which has no information about the contents of its counters, i.e. it does not know whether its counter are empty or nonempty. If by the computation should any counter go negative, no further transitions are allowed and the machine does not accept the input word. The machine accepts the input words if it ends the computation in a final state with all "pblind" counters empty. The class of one-way (deterministic) partially blind multicounter machines will be denoted by ***PBLIND*** (***DPBLIND***), the class of quasirealtime (deterministic) partially blind multicounter machines will be denote by ***QR-PBLIND*** (***QR-DPBLIND***).

Let $\boldsymbol{M}$ be a class of multicounter machines introduced. Then $\mathscr{L}(\boldsymbol{M})$ denotes the class of languages accepted by machines in $\boldsymbol{M}$. Let $A$ be a multicounter machine from the class $\boldsymbol{M}$ and $L(A)$ be the language accepted by $A$. Let $f$ be a real function defined on natural numbers. Then $L_{Rf}(A)$ denotes the set of all words in $L(A)$ for which there is an accepting computation containing at most $f(n)$ reversals, i.e. changes from increasing to decreasing contents of a counter or vice versa, where $n$ is the length of the input words. Let $\boldsymbol{G}$ be the class of all functions $q$ from natural numbers to positive real numbers such that for all natural numbers $n$: $f(n) \geqq q(n)$. Then we define the classes of languages

$$\mathscr{L}_{Rf}(\boldsymbol{M}) = \bigcup_{B \in \boldsymbol{M}} L_{Rf}(B) \quad \text{and} \quad \mathscr{L}(\boldsymbol{M} - R(f)) = \bigcup_{q \in G} \mathscr{L}_{Rq}(\boldsymbol{M}).$$

In what follows we shall often consider computations in which a multicounter machine reads a group of identical symbols whose number is greater than the number of states. Clearly, there has to be a state $q$ which will be entered twice or more in different configurations in this part of computation. If these two occurrences of the state $q$ are adjacent (no further state $q$ and no two equal states different from $q$ occur inbetween) we say that this part of the computation is a *cycle* with *state characteristic* $q$, *reading head characteristic* – the number (positive or zero) of symbols over which the reading head moves to the right in this cycle, and *counter characteristic* for each counter, which is the difference between the counter contents at the beginning and at the end of the cycle. Thus, the counter characteristic can be positive, if the machine increases the contents of the counter in this cycle, can be negative if the counter contents is decreased in this part of the computation, and it can obviously be zero.

Let $s$ be the number of states of a multicounter machine $A$, and $k$ be the number of counters. Then we can bound the number of all cycles with different characteristics by $s \cdot s \cdot (2s + 1)^k$.

Now, we introduce the following notation which we shall use in this paper. Let $d$ be a real number. Then $\{d\}$ is the smallest natural number $k$ such that $d \leqq k$, and $[d]$ is the greatest natural number $m$ such that $d \geqq m$. Let $f$ and $g$ be functions defined on naturals. Then the fact $\lim_{n \to \infty} f(n)/g(n) = 0$ will be denoted by $f(n) = o(g(n))$, and the fact that there exists a constant $c$ such that $\lim_{n \to \infty} f(n)/g(n) = c$ will be denoted by $f(n) = C(g(n))$. Let $N$ be the set of natural numbers.

For $u \in \{a, b\}^*$, $\# a(u)$ denotes the number of occurrences of $a$ in $u$.

## 3. RESULTS

We shall study the hierarchy of reversal bounded multicounter, and partially blind multicounter machines, where the reversal bound is a function of the input word length.

The following results were shown in [1]. Let $r_1 < r_2$ be positive, real numbers. Then[1]

$$\mathscr{L}(\boldsymbol{COUNTER}\text{-}R(0)) \subset \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(1)) \subset \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(\log_2 n))$$
$$\mathscr{L}(\boldsymbol{COUNTER}\text{-}R(n^{r_1})) \subset \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(n^{r_2}))$$
$$\mathscr{L}(\boldsymbol{COUNTER}\text{-}R((\log_2 n)^{r_1})) \subset \mathscr{L}(\boldsymbol{COUNTER}\text{-}R((\log_2 n)^{r_2}))\,.$$

All these results were proved for deterministic multicounter machines, and two-way ones in [1] too. Some stronger results for two-way multicounter machines can be found in [3].

We generalize these assertions for one-way multicounter machines, and extend them for partially blind ones too. In fact, we prove that determinism, one pblind counter, real time, and $g(n)$ reversal number bound can be more powerfull than nondeterminism, arbitrarily large number of counters, unbounded time, and $f(n) = o(g(n))$ reversal number bound. This result is formulated in the following main theorem.

**Theorem 1.** Let $g$, and $f$ be functions from $N$ to $N$ fulfilling the following conditions.

(1) $$n \geqq g(n) \geqq f(n) \quad \text{for all} \quad n \in N$$

(2) $$f(n) = o(g(n))$$

[1] In the sequel for the sets $A$, $B$ symbol $\subset$ means that $A$ is a proper subset of $B$.

Then

$$\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}\text{-}R(g)) - \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(f)) \neq \emptyset\,.$$

Using Theorem 1 we obtain several hierarchy results formulated in the following theorem.

**Theorem 2.** Let $g$, and $f$ be functions from $N$ to $N$ fulfilling the conditions (1),(2), (3), and (4) of Theorem 1. Then

$$\mathscr{L}(\boldsymbol{QR\text{-}PBLIND}\text{-}R(f)) \subset \mathscr{L}(\boldsymbol{QR\text{-}PBLIND}\text{-}R(g))$$
$$\mathscr{L}(\boldsymbol{PBLIND}\text{-}R(f)) \subset \mathscr{L}(\boldsymbol{PBLIND}\text{-}R(g))$$
$$\mathscr{L}(\boldsymbol{QR\text{-}COUNTER}\text{-}R(f)) \subset \mathscr{L}(\boldsymbol{QR\text{-}COUNTER}\text{-}R(g))$$
$$\mathscr{L}(\boldsymbol{COUNTER}\text{-}R(f)) \subset \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(g))\,.$$

Clearly, Theorem 2 does not involve all hierarchy consequences of Theorem 1. The hierarchy results contained in Theorem 2 can be formulated for deterministic machines, and for different types of time restriction too.

Concluding this section we formulate an open problem. What is the relation between $\mathscr{L}(\boldsymbol{QR\text{-}PBLIND}\text{-}R(f))$ and $\mathscr{L}(\boldsymbol{QR\text{-}COUNTER}\text{-}R(f))$, where $f$ is an arbitrary function?

## 4. THE PROOF OF THEOREM 1

The proof technique is a generalization of the proof technique used in [2, 7] for multihead finite automata.

We shall consider the language $L(g) = \{w = u_1 u_2 \dots u_p \mid \# a(w) = \# b(w)$, $u_j \in a^+b^+$ for $j = 1, \dots, p$, $p \leqq \{g(n)/2\}$, and for all $v$, $y$ in $\{a, b\}^*$ such that $w = vy$ it follows $\# a(v) \geqq \# b(v)\}$. This language belongs to $\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}\text{-}R(g))$ because there is a ***QR-DPBLIND*** machine $B$ (with one pblind counter) which accepts $L(g)$ within the reversal bound $g$. The machine $B$ reading the symbol $a$ increases the content of its pblind counter by 1 and reading the symbol $b$ decreases the content of its pblind counter by 1. If $B$ ends the computation with empty pblind counter it will accept the input word. Let $w = u_1 u_2 \dots u_k$ be an input word for $k > \{g(n)/2\}$, $u_i$ in $a^+b^+$. Then $B$ according to its definition has to reverse its pblind counter at least $2k - 1 > g(n)$ times what means that the computation on $w$ cannot be accepting.

Now, we shall show by contradiction that $L(g)$ does not belong to $\mathscr{L}(\boldsymbol{COUNTER}$-$R(f))$. Let $h$ be a function from naturals to positive reals such that $h(n) \leqq f(n)$. Let there is a COUNTER machine $C$ with $k$ counters, for a natural $k$, such that $L_{Rh}(C) = L(g)$. Let us consider an input word

$$x = (a^{s+1}b^{s+1})^{d_1 d_2 (k+1) h(n)}\, a^t b^t$$

of the length $n = (2s + 2)\, d_1 d_2 (k + 1)\, h(n) + 2t$, where $d_1, d_2$. and $t$ are some constants such that $d_1 > s$, and $d_2 > s^2(2s + 1)^k$ (the number of cycles with different characteristics). It is clear that $x$ belongs to $L(g)$ because $2(s+1)\,d_1 d_2(k+1)\,h(n) \leqq \leqq n$ for sufficiently large $n$ $(h(n) \leqq f(n) = o(g(n))$ i.e. $h(n) = o(n))$.

In the following we shall construct a word $x'$ not in $L(g)$ such that $A$ accepts $x'$ if $A$ accepts $x$. Clearly, this will prove our assertion.

It is easy to see that there exists such a subword

$$x_1 = (a^{s+1} b^{s+1})^{d_1 d_2 (k+1)}$$

of $x$ that the machine $A$ when computing on $x_1$ does not reverse its counters. The fact that no reversal of counters is done in the part $y_1$ of the accepting computation $y$ on the subword $x_1$ implies that each counter can be emptied at most once in the computation $y_1$ on $x_1$. So we can assume that there exists such subword

$$x_2 = (a^{s+1} b^{s+1})^{d_1 d_2}$$

of the word $x_1$ that no counter is reversed or emptied in the part $y_2$ of the accepting computation $y$ on $x_2$.

Now, we shall consider all cycles with reading head characteristic 0 in the computation $y_2$ on $x_2$. Let $y = u_1 y_2 u_2 = u_1 v_1 r_1 v_2 r_2 v_3 u_2$, where $r_1$ and $r_2$ are some cycles with the same state characteristic and the reading head characteristic 0, and $v_1$, $v_2$, $v_3$, $u_1$, $u_2$ are some parts of the computation. Considering the fact no counter is reversed or emptied in $y_2$ it can be easily seen that if $y = u_1 v_1 r_1 v_2 r_2 v_3 u_2$ is an accepting computation on $x = w_1 x_2 w_2$ then $y' = u_1 v_1 r_1 r_2 v_2 v_3 u_2$ is an accepting computation on $x$ too. So we can assume that the accepting computation $y = u_1 y_2 u_2 = = u_1 z_1 o_1 z_2 o_2 \dots z_c o_c z_{c+1} u_2$, where $c \leqq s$, and $o_j$ consists of all cycles with the reading head characteristic 0 and some state characteristic $q_j$ (obviously the reading head is stationary in the part $o_j$ of the computation) and $z_j$ involves no cycle with reading head characteristic 0 for $j = 1, \dots, c+1$.

Using this assumption we obtain that there is such a subword

$$x_3 = (a^{s+1} b^{s+1})^{d_2}$$

of the word $x_2$ that the part $y_3$ of the accepting computation $y$ on $x_3$ involves no cycle with reading head characteristic 0, and no counter is reversed or emptied in the computation $y_3$ on $x_3$. There exists at least one cycle on each subword $a^{s+1}$ of $x_3$ and since the number of cycles with different characteristics is bounded by $s^2(2s+1)^k < d_2$ there exist some cycles $p_1$ and $p_2$ with the same characteristics which are situated in two parts of the computation $y_3$ on two different groups of $a$'s of the word $x_3$. Let the reading head characteristic of $p_1$ and $p_2$ be $m$. Clearly $m > 0$.

Choosing $m$ symbols $a$ from the first group of $a$'s and pumping $a^m$ to the second group of $a$'s we obtain a word $x'$ which does not belong to $L(g)$. Now, we shall construct an accepting computation on the word $x'$ what proves our assertion for $g(n) = C(n)$.

Let $y = u y_3 v = u s_1 p_1 s_2 p_2 s_3 v$ be the accepting computation on the word $x = = w x_3 w'$, where $y_3 = s_1 p_1 s_2 p_2 s_3$ is the part of the computation on $x_3$ and $u$, $v$, $s_1$, $s_3$ are some parts of the computation. Then the accepting computation on $x'$ will be

$us_1s_2p_1p_2s_3v$, because the machine $A$ is after the initial part of the computation $us_1p_1s_2p_2s_3$ on $x$ in the same configuration (it means in the same state, with the same contents of all counters, and with the same postfix $w'$ of the words $x$ and $x'$ on the input tape) as after the initial part of the computation $us_1s_2p_1p_2s_3$ on $x'$.

We call attention to the fact that no counter can be emptied in the computation $s_1s_2p_1p_2s_3$ what is the essential point in our consideration (no counter can reverse in the computation $s_1s_2p_1p_2s_3$ and no counter can be emptied in the computation $s_1p_1s_2p_2s_3$ imply this fact).

*Remark.* For $g$ fulfilling the additional conditions
(i) $g(n) = C(n)$,
(ii) $g$ is an increasing function,
the language

$L(g) = \{w \in \{a, b\}^* \mid \# a(w) = \# b(w)$ and for each prefix $u$ of $w$ $\# a(u) \geqq \# b(u)\}$

is a witness language for $\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}\text{-}R(g)) - \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(f))$. Clearly $L(g)$ is in $\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND})$. Since it is known [9] that $\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}) \subseteq \mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}\text{-}R(g))$ for $g$ fulfilling (i) and (ii) we see that $L(g)$ is in $\mathscr{L}(\boldsymbol{QR\text{-}DPBLIND}\text{-}R(g))$. The proof of the fact $L(g) \notin \mathscr{L}(\boldsymbol{COUNTER}\text{-}R(f))$ is as above.


## ACKNOWLEDGEMENTS

I am indebted to Pavol Ďuriš for several interesting and helpful discussions. I would like to thank Branislav Rovan, and the referee for their comments concerning this work.

(Received March 29, 1984.)


## REFERENCES


[1] T. Chan: Reversal complexity of counter machines. In: Proc. IEEE Symposium on Theory of Computing 1981, IEEE, New York, 146—157.

[2] P. Ďuriš and J. Hromkovič: One-way simple multihead finite automata are not closed under concatenation. Theoret. Comput. Sci. *27* (1983), 121—225.

[3] P. Ďuriš and Z. Galil: On reversal-bounded counter machines and on pushdown automata with a bound on the size of the pushdown store. Inform. and Control *54* (1982), 3, 217—227.

[4] S. Ginsburg: Algebraic and Automata — Theoretic Properties of Formal Languages. North-Holland Publ. Comp., Amsterdam 1975.

[5] S. A. Greibach: Remarks on blind and partially blind one-way multicounter machines. Theoret. Comput. Sci. *7* (1978), 311—324.

[6] M. Hack: Petri Net Languages, Computation Structures. Group Memo 124, Project MAC, MIT, 1975.

[7] J. Hromkovič: Closure properties of the family of languages recognized by one-way two-head deterministic finite state automata. In: Mathematical Foundations of Computer Science 1981 — Proc. 10th Symposium, Štrbské pleso, Czechoslovakia, August 31 — September 4, 1981 (J. Gruska, M. Chytil, eds.). (Lecture Notes in Computer Science 118.) Springer-Verlag, Berlin—Heidelberg—New York 1981, 304—313.

[8] J. Hromkovič: Hierarchy of reversal and zerotesting bounded multicounter machines. In: Mathematical Foundations of Computer Science 1984 — Proc. 11th Symposium, Prague, Czechoslovakia, September 3—7, 1984 (M. P. Chytil, V. Koubek, eds.). (Lecture Notes in Computer Science 176.) Springer-Verlag, Berlin—Heidelberg—New York—Tokyo 1984, 312—321.

[9] J. Hromkovič: Reversal bounded multicounter machines. Computers and Artificial Intelligence *4* (1985), 4, 361—366.

[10] O. H. Ibarra: Reversal-bounded multicounter machines and their decision problems. J. Assoc. Comput. Mach. *25* (1978), 116—133.

*RNDr. Juraj Hromkovič, Katedra teoretickej kybernetiky a matematickej informatiky Matematicko-fyzikálnej fakulty Univerzity Komenského (Department of Theoretical Cybernetics — Comenius University), 842 15 Bratislava. Czechoslovakia.*