R. 97
Reg 1203 Cas

140008597
M-REPORT/357

ABOUT LOG-ON LANGUAGES

Preliminary Version

R.Casas
J.Gabarró

RR 85/02

ABOUT LOG-ON LANGUAGES

Preliminary Version

R.Casas

J.Gabarró(&)

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya,08034 Barcelona

Spain

(&) Aquest treball ha estat realitzat,en part,durant una estada a França,possible gràcies a un ajut concedit per la CIRIT,Diari Oficial de la Generalitat n° 486, 16 de Novembre de 1984.

Abstract : We deal with on-line log-space Turing Machines, with markers in the work tapes.

In this type of machines we prove the existence of a language L satisfing

$$L \in NSPACE_{\not s}(\log n) \ , \ \bar{L} \in NSPACE_{\not s}(\log n)$$

and

$$L \notin DSPACE_{\not s}(\log n)$$

We give a explicit definition of L.The main theorem uses minimisation techniques of deterministic finite automata.

Resum : Tractem amb Màquines de Turing que tenen marcadors en les cintes de treball.

En aquest tipus de màquines demostrem l'existència d'un llenguatge L que satisfà

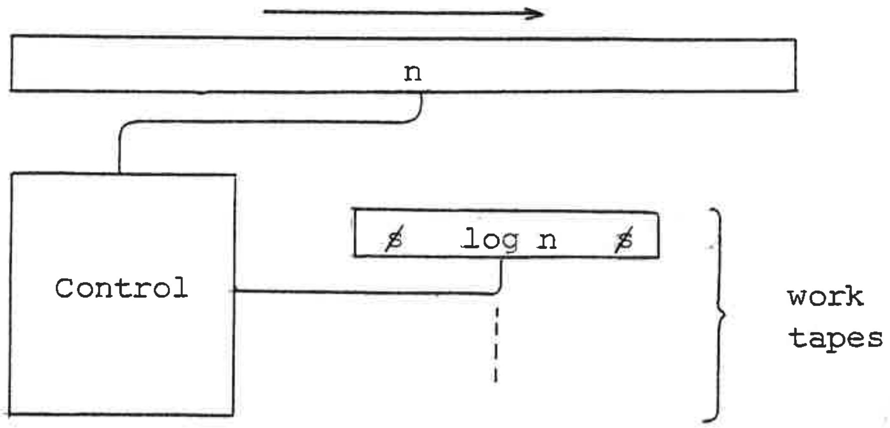
$$L \in NSPACE_{\not s}(\log n) \ , \ \bar{L} \in NSPACE_{\not s}(\log n)$$

i

$$L \notin DSPACE_{\not s}(\log n)$$

Donem la definició explícita de L.El teorema principal utilitza les tècniques de minimització d'autòmats finits.

Our model of machine is the on-line log-space Turing machine. When the machine has markers in the work tape, we note $MT_{\not b}$. More specifically we are interested in on-line log-space TM with markers. Schematically:



We denote the complexity classes defined by these machines by a subindex $\not b$. For example $NSPACE_{\not b}(\log n)$ is the class of languages defined by a non-deterministic on-line log-space Turing machine with markers.

Let us define the language L.

<u>Definition 1</u> : Let us consider the language L over $\{0,1,a\}^*$ which is given by:

$$L = \{g_1 a g_2 a \ldots g_k a \mid k \geq 1, \text{for every } 1 \leq i \leq k \text{ the bloc } g_i \text{ belongs to } \{0,1\}^* \text{ and one half of the blocs have a letter 1 in position } \lfloor \log k \rfloor \}$$

Throughout the paper $\frac{1}{2}k$ or $\log k$ will mean $\lfloor \frac{1}{2}k \rfloor$ or $\lfloor \log k \rfloor$

The language L satisfies

Lemma 1 : The defined language verifies

$$L \in NSPACE_{\not{b}}(\log n)$$

$$\bar{L} \in NSPACE_{\not{b}}(\log n)$$

Proof. Let $w \in L$ be such that $w = g_1 a g_2 a \ldots g_k a$, then $1 \leq k \leq w$.

The following algorithm simulates a non-deterministic $TM_{\not{b}}$ which accepts L in log n space:

(i) The machine guess k.This can be done in log space.

(ii) Find log k.

(iii) Count the number of blocs g in w having 1 in position log k. Let y be that number.

(iv) Test if $y = \frac{1}{2}k$

(v) Test if k is the number of blocs.

Asimilar non-deterministic machine to accept $\bar{L}$ could be obtained by changing:

(iv)' Test if $y \neq \frac{1}{2}k$.

∎

Remark : By a small change in the last lemma it can be proved that

$$L \in NSPACE(\log n)$$

$$\bar{L} \in NSPACE(\log n)$$

We study a deterministic machine for L.

Lemma 2 : $L \in DSPACE(\log^2 n)$

Proof. Let w be in L with $w = g_1 a g_2 a \ldots g_k a$, then $1 \leq k \leq |w| = n$

as $0 \leq \lfloor \log k \rfloor \leq \lfloor \log n \rfloor$ then:

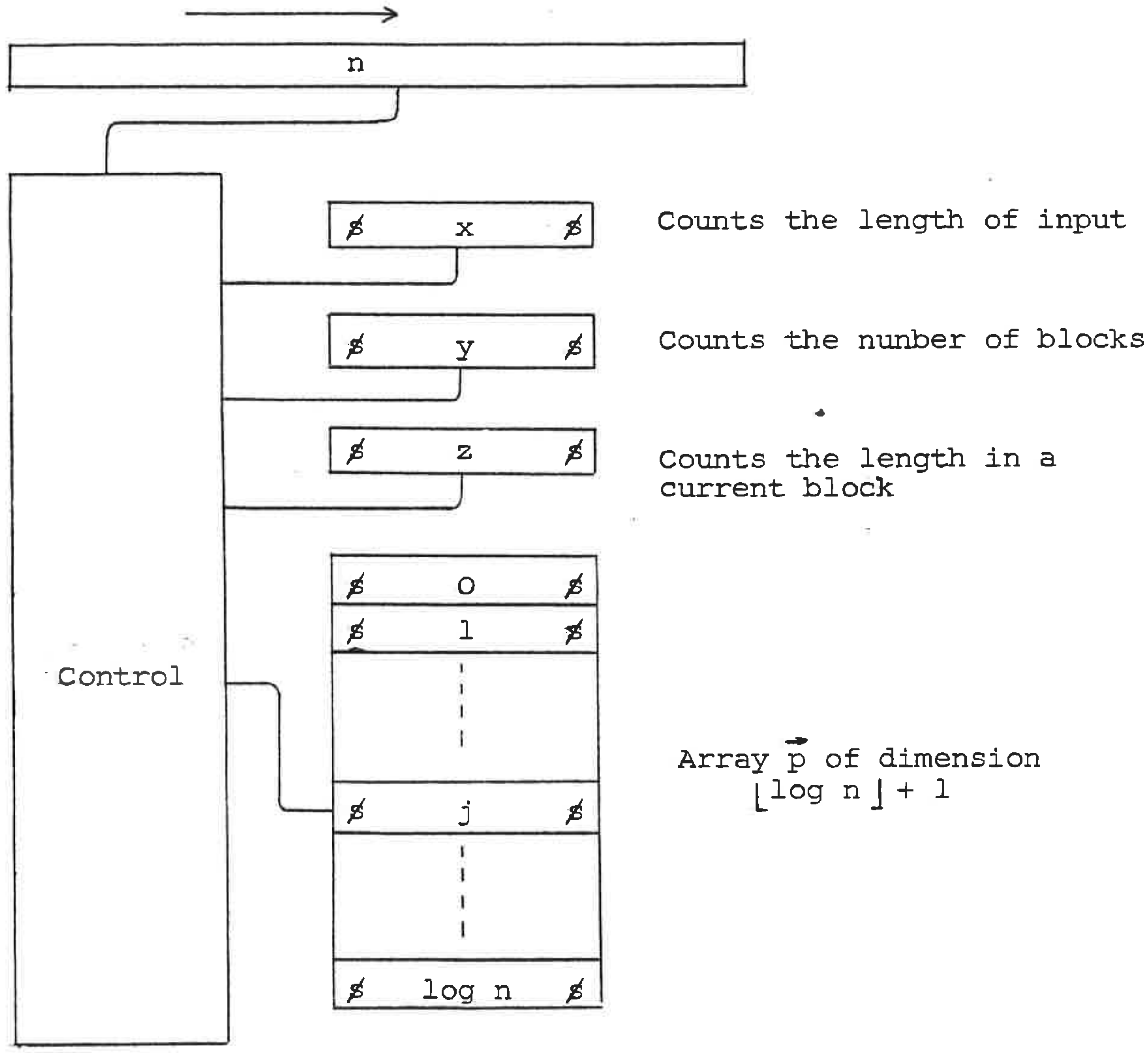
$$\log k \in \{0,1,2,\ldots, \lfloor \log n \rfloor -2, \lfloor \log n \rfloor - 1, \lfloor \log n \rfloor\}$$

Consider a $TM_{\not b}$ which uses a tape as counter for every value

$$\{0,\ldots,\lfloor \log n \rfloor\}$$

When the current bloc g has a 1 in position $t, 1 \leq t \leq \log n$, the $p(t)$ counter increases by 1 the content.

Schematically:



When the machine has read a left factor $w = g_1 a g_2 a \ldots g_m a$

for every $0 \leq i \leq \lfloor \log n \rfloor$, $p(i) = t$ iff there exists exactly t

blocs g with 1 in position i.

When we have read the entiere input word, the acceptance condition

is

$$p(\lfloor \log y \rfloor) = \left\lfloor \frac{y}{2} \right\rfloor$$

We shall study memory bounds. The counters x,y,z are bounded

by log n. As $p(i) \leq \log n$ then

$$\sum_{i=0}^{\log n} p(i) \sim (\lfloor \log n \rfloor)^2$$

In the next lemmas we shall consider configurations like

$$c = (x, y, z, \vec{p})$$

Therefore our interest lyes in some components of vector $\vec{p}$.

We cut $\vec{p}$ in three parts. Look at the next figure.

We shall study components between $1 + \frac{1}{2}\log n$ and $1 + \frac{3}{4}\log n$.

If a word is accepted using the counter $1 + \frac{1}{2}\log n$ then

$$p(1 + \tfrac{1}{2}\log n) = \tfrac{1}{2} . 2^{1+\frac{1}{2}\log n} = \sqrt{n}$$

In general for every counter

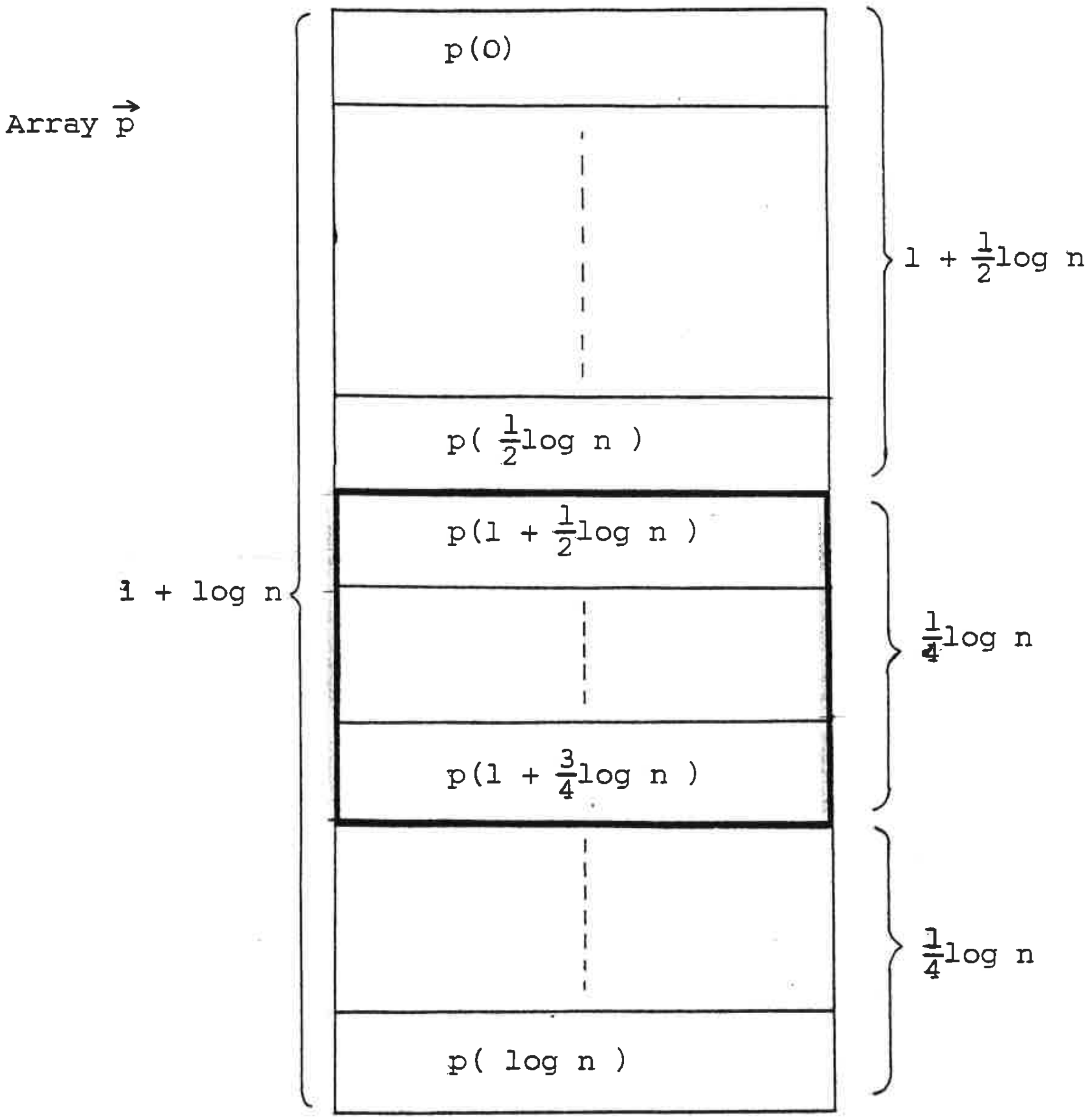
$$1 + i + \tfrac{1}{2}\log n \quad , \quad 0 \leq i \leq \tfrac{1}{4}\log n$$

we have

$$p(1 + i + \tfrac{1}{2}\log n) = 2^i \sqrt{n}$$

We are interested in an accessibility problem for a non-

polynomial number of configurations.



To study that problem, we fill the vector $\vec{p}$ in such a way that:

(i) $p(i) = 0 \quad , \quad 0 \leq i \leq \frac{1}{2}\log n$

(ii) $p(i) = x \quad , \quad 1 + \frac{1}{2}\log n \leq i \leq 1 + \frac{3}{2}\log n \quad , \quad x \leq \sqrt{n}$

(iii) $p(i) = 0 \quad , \quad 2 + \frac{3}{2}\log n \leq i \leq \log n$

The values of $\vec{p}$ and the accessibility of these configurations induce us to define the set Q and the words $g_{\vec{q}}$.

<u>Definition 2</u> For every n, let Q be the set of vectors

$$Q = \{0\}^{1+\frac{1}{2}\log n} \cdot \{0,1,\ldots,\sqrt{n}-1\}^{\frac{1}{4}\log n} \cdot \{0\}^{\frac{1}{4}\log n}$$

for every $\vec{q} \in Q$ consider the word

$$g_{\vec{q}} = g_1 a g_2 a \ldots g_i a \ldots g_{\sqrt{n}} a$$

such that for each $1 \leq i \leq \sqrt{n}$ we have

$$g_i = 0^{1+\frac{1}{2}\log n} \; u_1^i u_2^i \ldots u_m^i \ldots u^i_{\frac{1}{4}\log n}$$

with

$$u_m^i = \begin{cases} 1 \text{ if } q(1+m+\frac{1}{2}\log n) \geq i \\ 0 \text{ otherwise} \end{cases}$$

Given any $j, 0 \leq j \leq \sqrt{n}$ we denote by $g_{\vec{q},j}$ the left factor of g_q containing j bloks, formally

$$g_{\vec{q},j} = g_1 a g_2 a \ldots g_j a$$

notice that

$$g_{\vec{q},\sqrt{n}} = g_{\vec{q}}$$

We are interested in values for n such that $\frac{1}{2}\log n, \frac{1}{4}\log n, \sqrt{n}$ are naturals, let us consider

$$n = 2^{4x} \quad , \quad x \geq 1$$

then $\sqrt{n} = 2^{2x}$, $\frac{1}{2}\log n = 2x$, $\frac{1}{4}\log n = x$

This we obtain a second version of the last definition

Definition 3 : Let $n = 2^{4x}$, $x \geq 1$ then consider the set

$$Q = \{0\}^{1+2x} \cdot \{0,1,\dots,2^{2x}-1\}^{x} \cdot \{0\}^{2x}$$

for every $\vec{q} \in Q$ let us consider the word

$$g_{\vec{q}} = g_1 a g_2 a \dots g_i a g_{2^{2x}} a$$

such that for each $i, 1 \leq i \leq 2^{2x}$ we have

$$g_i = 0^{2x+1} u_1^i u_2^i \dots u_m^i \dots u_x^i$$

with

$$u_m^i = \begin{cases} 1 \text{ if } q(1+2x+m) \geq i \\ 0 \text{ otherwise} \end{cases}$$

Given any $j, 1 \leq j \leq 2^{2x}$ let $g_{\vec{q},j}$ denote the left factor of g_q with j blocks

$$g_{\vec{q},j} = g_1 a g_2 a \dots g_j a$$

notice that

$$g_{\vec{q},2^{2x}} = g_{\vec{q}}$$

In the next lemma we shall see that after reading $g_{\vec{q}}$ the $MT_{\not\leq}$ satisfies

$$\vec{p} = \vec{q}$$

<u>Lemma 3</u> : Take $\vec{q} \in Q$ and $1 \leqslant m \leqslant \frac{1}{4}\log n$.

(a) When the $TM_{\not{s}}$ has read the prefix $g_{\vec{q},j}$, the counter $1 + \frac{1}{2}\log n + m$ satisfies

$$p(1 + \tfrac{1}{2}\log n + m) = \begin{cases} j \text{ if } u_m^j = 1 \\ q(1 + \tfrac{1}{2}\log n + m) \text{ otherwise} \end{cases}$$

(b) When the $TM_{\not{s}}$ has read the complete word $g_{\vec{q}}$ it in the configuration

$$c_{\vec{q}} = (\sqrt{n}(2 + \tfrac{3}{4}\log n) \quad , \sqrt{n} \quad , \quad 0 \quad , \quad \vec{q})$$

<u>Proof</u> We denote $r = 1 + \frac{1}{2}\log n + m$.

(a) When the $TM_{\not{s}}$ has read the block $g_{\vec{q},j}$ there are two possibilities.

If $1 \leqslant j \leqslant r$ the counter $p(r)$ has value j.

By construction the blocks $g_{\vec{q},1}, g_{\vec{q},2}, \ldots, g_{\vec{q},r}$ satisfy:

$$u_m^1 = u_m^2 = \ldots = u_m^r = 1$$

If $r < j \leqslant \sqrt{n} - 1$ the counter $p(r)$ has value $q(r)$.

The blocks $g_{\vec{q},r+1}, g_{\vec{q},r+2}, \ldots, g_{\vec{q},\sqrt{n}-1}$ satisfy

$$u_m^{r+1} = u_m^{r+2} = \ldots = u_m^{\sqrt{n}-1} = 0$$

(b) We denote $c_{\vec{q}} = (x,y,z,p)$

(i) After reading the complete word $g_{\vec{q}}$ we have processed an exact number of blocks.

That means $z = 0$.

(ii) By construction the number of blocks is $\sqrt{n}$, then $y = \sqrt{n}$.

(iii) Every block $g_i a$ has a length of:

$$1 + 1 + \frac{1}{2}\log n + \frac{1}{4}\log n = 2 + \frac{3}{4}\log n$$

As there are $\sqrt{n}$ blocks we come to

$$x = |g_{\vec{q}}| = \sqrt{n}(2 + \frac{3}{4}\log n)$$

(iv) Considering that $g_{\vec{q},\sqrt{n}} = g_{\vec{q}}$ we come to $\vec{p} = \vec{q}$

■

Lemma 4 : For every $\vec{q} \in Q$ we have $g_{\vec{q}} \notin L$.

Proof The word $g_{\vec{q}}$ has $\sqrt{n}$ blocks. If $g_{\vec{q}} \in L$ we need $q(\log n) = \frac{1}{2}\sqrt{n}$. By construction $q(\frac{1}{2}\log n) = 0$ then $g_{\vec{q}} \notin L$.

■

Lemma 5 : Consider $\vec{q}, \vec{q}'$ in Q, with $\vec{q} \neq \vec{q}'$, we cannot merge the configurations $c_{\vec{q}}$ and $c_{\vec{q}'}$.

Proof As $\vec{q} \neq \vec{q}'$ there exists $m, 1 \leq m \leq \frac{1}{4}\log n$, such that $\vec{q}$ and $\vec{q}'$ differ in component

$$r = 1 + \frac{1}{2}\log n + m$$

suppose

$$x = q(r) > q'(r) = y$$

Take

$$g_q = g_1 a g_2 a \ldots g_{\sqrt{n}} a \quad \text{with } g_i = 0^{1 + \frac{1}{2}\log n} u_1^i u_2^i \ldots u_t^i$$

$$g_{q'} = f_1 a f_2 a \ldots f_{\sqrt{n}} a \quad \text{with } f_i = 0^{1 + \frac{1}{2}\log n} v_1^i v_2^i \ldots v_t^i$$

with $t = \frac{1}{4}\log n$

we have

$$g_x = 0^{1+\frac{1}{2}\log n} u_1^x \ldots u_{m-1}^x 1 u_{m+1}^x \ldots u_t^x$$

$$f_x = 0^{1+\frac{1}{2}\log n} v_1^x \ldots v_{m-1}^x 0 v_{m+1}^x \ldots v_t^x$$

When the machine has read the left factor $g_{\vec{q},r}$. That counter r has his maximum value x. That means $p(r) = x$.

When the machine has read $g_{\vec{q},y}$ the counter satisfies $p(r) = y$.

We need to complete $g_{\vec{q}}$ with a word g, such that $w = g_{\vec{q}}\, g$ satisfies:

(1) The word w has length n.

(2) The word w has 2^r blocks.

(3) The word w has exactly $\frac{1}{2}2^r$ blocks with a value 1 in position r.

Considering (1), (2) and (3) we conclude $w \in L$.

We proceed step by step.

(a) Consider a word g of the form

$$g = h_1 a h_2 a \ldots h_i a$$

with $h_j \in \{0,1\}^*$, $1 \leq j \leq i$. It is easy to see that

$$i = \sqrt{n}(2^{m-1} - 1)$$

(b) We need a total of

$$\frac{1}{2}\, 2^r = 2^m \sqrt{n}$$

occurrences of 1 in position r. As $g_{\vec{q}}$ contain x occurrences, g needs to contain

$$t = 2^m . \sqrt{n} - x$$

occurrences of 1 in position r.

(c) We define

$$h = 0^{1+\frac{1}{2}\log n} 0^{m-1} 1 0^{\frac{1}{4}\log n - m}$$

then $|h| = 1 + \frac{3}{4}\log n$ and h has only one occurrence of 1 in position m.

Take $h_1 = h_2 = \ldots = h_t = h$, when $t = 2^m \sqrt{n} - x$.

(d) In this moment we have

$$w = g_q (ha)^t h_{t+1} a \ldots h_i a$$

it is easy to see that $r - i = \sqrt{n}(2^m - 1) + x = s$

as $1 \leq m \leq \frac{1}{4}\log n$, $0 \leq x \leq \sqrt{n} - 1$ we come to $r - i \geq \sqrt{n} + x > 0$

(e) We take $h_{t+1} = \ldots = h_{i-1} = \lambda$, where λ is the empty word and $h_i = 0^y$, taking y to obtain a word of length n.

Finally

$$w = g_{\vec{q}} (ha)^t a^s 0^y a$$

with

$$n = |w| = (2^m \sqrt{n} + \sqrt{n} - x)(2 + \frac{3}{4}\log n) + 2^m \sqrt{n} - \sqrt{n} + y - 1$$

as $m \leq \frac{1}{4}\log n$, $2^m \sqrt{n} \leq n^{3/4}$,for increasing values of n,y becomes positive,and the word w exists.

(f) Consider the word $w' = g_{\vec{q}'} \cdot g$,with the same g as in w.We have $w' \notin L$,then $c_{\vec{q}}$ and $c_{\vec{q}'}$ cannot be merget.

∎

<u>Corollary 1</u> : The language L has a deterministic space lower bound given by $\Omega(\log^2 n)$ infinitely often.

<u>Proof</u> The cardinal of the set Q is given by

$$\| Q \| = (\sqrt{n} + 1)^{\frac{1}{4}\log n} \sim n^{\frac{1}{8}\log n}$$

As for $\vec{q} \neq \vec{q}'$ we cannot merge configurations of deterministic

$TM_{\not\beta}$, this machine has at least $\| Q \|$ configurations.

To code these configuartions we need at least

$$\log(n^{\frac{1}{8}\log n}) = \log^2 n$$

bits.

■

We can conclude:

<u>Theorem 1</u> : $L \notin DSPACE_{\not\beta}(\log n)$

In connection with initial index(Ga,83) it is easy to prove that:

$$a_L(n) \sim \theta(n^{\frac{1}{8}\log n})$$

<u>References</u>

(Ga,83) Gabarro,J.1983."Initial Index:A New Complexity Function For Languages.ICALP,83,LNCS 154,226-236.