# Complexity Of Calculation On Stack Automatas

by

**David Livshin**

written in 1976
recreated in 2020

## Abstract

In this paper we prove that for an arbitrary stack automata during processing of a word of the length $n$ the following holds:

the time of the processing does not exceed $const|n|^2$

the space consumed during the processing does not exceed $const|n|$

The example of a stack automata is presented for which the above estimates may not be improved.

## $1

1.

We will provide not a formal definition of a stack automata.

One way not-deterministic stack automata ( SA ) is a not deterministic Turing machine with one input and one working ( stack ) tapes with the following restrictions imposed:

a) both tapes are restricted from the left; the leftmost cell is called *'bottom'* and denote *as* $\boldsymbol{O}$.
b) input tape has reading head that can move from left to right ( not necessarily at each cycle )
c) working tape has one head; the context of the cell under this head may be altered only when all the cells right of it are empty
d) at each moment of time ( cycle ) all the cells on the left of the head of the working tape are not empty

In the set of internal states of SA there is initial state and one or more terminating states. At the beginning of the work of SA

- all the cells of the working ( stack ) tape are empty except the most left cell ( the bottom of the stack ) which contains unique symbol that may not be modified during work and may not appear in other cells
- the heads of both tapes are located at the left-most positions and automata is in initial state

2.

Let $\Delta$ = ( $\Delta$(1), $\Delta$(2),…) to be a sequence (finite or infinite ) of instructions of SA $\mathcal{M}$.

We say that the word $x$ allows sequence $\Delta$ if it is possible process execution of the word $x$ by $\mathcal{M}$ such that $\forall i$ at $i$'s cycle $\mathcal{M}$ executes command $\Delta(i)$.

We say that word $x$ is accepted by automata $\mathcal{M}$ if there is a sequence of instructions allowed by the word $x$ after execution of which $\mathcal{M}$ gets into a terminating state. The set of all the words accepted by $\mathcal{M}$ is called a language accepted by $\mathcal{M}$ and denoted by L($\mathcal{M}$ ).

3.

For every word $x \in L(\mathcal{M})$ acand sequence $\Delta$ that accepts $x$ denote

- $t_{\mathcal{M}}(x,\Delta)$ to be number of cycles that were performed while processing $x$ according to $\Delta$ till acceptance of $x$.
- $s_{\mathcal{M}}(x,\Delta)$ to be the maximum size of the stack tape that was visited by the head while processing $x$ according to $\Delta$ till acceptance of $x$.

Denote $t_{\mathcal{M}}(x) = \min t_{\mathcal{M}}(x,\Delta)$, $s_{\mathcal{M}}(x) = \min s_{\mathcal{M}}(x,\Delta)$ where minimum is taken for all sequences $\Delta$ that accepts word $x$.
Denote $t_{\mathcal{M}}(n) = \max t_{\mathcal{M}}(x)$, $s_{\mathcal{M}}(n) = \max s_{\mathcal{M}}(x)$ where maximum is taken for all $x \in L(\mathcal{M})$ such that $|x| = n$.

4.

We call a '*zone*' a consecutive set of cells of the stack tape and '*zone configuration*' at the time $t$ – the word that is written inside the cells of zone at the time $t$. We call '*full configuration*' at the time $t$ the *configuration* of the smallest *zone* at the time $t$ that contains stack *bottom* and all the cell on the right to it are empty.

Let numerate the cells of the stack tape in the natural way from left to right with the *bottom* being cell number 0.
The description of SA at time $t$ is a 3-tuple $\langle i,g,\alpha \rangle$ where

$i$ - is the number of the cell under the head
$g$ – is the state of SA
$\alpha$ - full configuration of SA at the time $t$

5.

Define the important for the future discussion meaning of the '*cutoff function*'. We will need few types of this function.

First define function $L^{\alpha}_{i,j}(T)$, $i$={0;1}, $j$={0,1}, where $\alpha$ is a word of the stack tape, $T$ is a set of words of the input tape. Let $Q=(g_1,\dots,g_m)$ to be a set of the states of SA. Then $L^{\alpha}_{i,j}(T) = \langle L^{1}_{i,j}, \dots, L^{m}_{i,j} \rangle$ where $L^{k}_{i,j}$ is defined as following:

- write in some zone $\mathcal{K}$ of the size $|\alpha|$ the word $\alpha$.
- if $i$=0 put the head at the left-most position of the zone $\mathcal{K}$, if $i$=1 put the head at the right-most position of zone $\mathcal{K}$.
- put SA in the state $g_k$.
- if while provided an input word $x \in T$ there is a process of SA execution that doesn't change configuration $\alpha$ in the zone $\mathcal{K}$ and

  the head first leaves zone $\mathcal{K}$ to the left then add $\langle g,x \rangle$ to $L^{k}_{i,0}$
  the head first leaves zone $\mathcal{K}$ to the right then add $\langle g,x \rangle$ to $L^{k}_{i,1}$

Note that the above definition doesn't depend on zone $\mathcal{K}$.

Now define function $M^{\alpha}_{i,j}(T)$ as a vector $\langle M^{1}_{i,j}, \dots, M^{m}_{i,j} \rangle$ where

$$M^{k}_{i,j} = \{ g \mid \exists x \in T: \langle g,x \rangle \in L^{k}_{i,j} \}$$

6.

Assume that while processing $\boldsymbol{x}$ according to $\boldsymbol{\Delta}$ in some zone $\mathcal{K}$ configuration $\boldsymbol{\alpha}$ appears, $\boldsymbol{D}$ being a point from $\mathcal{K}$. Denote by $\boldsymbol{g}_1^{\boldsymbol{D}}(\boldsymbol{\alpha})$ the state during creation of $\boldsymbol{\alpha}$ before the last exit right from $\boldsymbol{D}$ when all the cells right from $\boldsymbol{D}$ are empty; if this wouldn't happen then define $\boldsymbol{g}_1^{\boldsymbol{D}}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \boldsymbol{g}_{-1}$ where $\boldsymbol{g}_{-1} \notin \boldsymbol{Q}$ is an element that doesn't belong to the set of states of SA. Denote by $\boldsymbol{g}_2^{\boldsymbol{D}}(\boldsymbol{\alpha})$ the state during elimination of $\boldsymbol{\alpha}$ before the last exit left from $\boldsymbol{D}$ when all the cells right from $\boldsymbol{D}$ are empty; if this wouldn't happen then define $\boldsymbol{g}_2^{\boldsymbol{D}}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \boldsymbol{g}_{-1}$ .

# $2

Consider an arbitrary SA $\mathcal{M}$.

<u>Theorem 1:</u>

$$\exists \boldsymbol{c}\ \forall \boldsymbol{x} \in L(\mathcal{M}):\ \boldsymbol{s}_{\mathcal{M}}(\boldsymbol{x}) \leq \boldsymbol{c} * |\boldsymbol{x}|,\ \ \boldsymbol{t}_{\mathcal{M}}(\boldsymbol{x}) \leq \boldsymbol{c} * |\boldsymbol{x}|^{2}.$$

<u>Proof:</u>

Assume that configuration $\boldsymbol{\alpha}$, that is created while processing $\boldsymbol{x} \in L(\mathcal{M})$ according to $\boldsymbol{\Delta}$ in some zone $\mathcal{K}$, satisfies the following conditions:

I. At some points of time $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ : $\boldsymbol{t}_1 > \boldsymbol{t}_2$ descriptions of $\mathcal{M}$ were $<\boldsymbol{x}', |\boldsymbol{\gamma}|, \boldsymbol{g}_1, \boldsymbol{\gamma}>$ and $<\boldsymbol{x}', |\boldsymbol{\gamma\alpha}|, \boldsymbol{g}_2, \boldsymbol{\gamma\alpha}>$ respectfully and during time interval $[\boldsymbol{t}_1, \boldsymbol{t}_2]$ the configuration $\boldsymbol{\gamma}$, that was generated in the zone and located at the left of $\mathcal{K}$, was not affected ( changed ).

II. At some points of time $\boldsymbol{t}_3$ and $\boldsymbol{t}_4$: $\boldsymbol{t}_4 > \boldsymbol{t}_3 > \boldsymbol{t}_2$ descriptions of $\mathcal{M}$ were $<\boldsymbol{x}'', |\boldsymbol{\gamma\alpha}|\boldsymbol{-1}, \boldsymbol{g}_3, \boldsymbol{\gamma\alpha}>$ and $<\boldsymbol{x}'', |\boldsymbol{\gamma}|\boldsymbol{-1}, \boldsymbol{g}_4, \boldsymbol{\gamma}>$ respectfully and during time interval $[\boldsymbol{t}_3, \boldsymbol{t}_4]$ the configuration $\boldsymbol{\alpha}$ was not affected ( changed ).

III. During the stay of the head inside configuration $\boldsymbol{\alpha}$ no input symbols were accepted.

It is easy to show that there is $\boldsymbol{N_1}$ such that for every word $\boldsymbol{x} \in L(\mathcal{M})$ there is a sequence $\boldsymbol{\Delta}$ that accepts $\boldsymbol{x}$ such that while processing $\boldsymbol{x}$ according to $\boldsymbol{\Delta}$ every configuration will have a length less than $\boldsymbol{N_1}$ ( indeed if configuration $\boldsymbol{\alpha}$ satisfies conditions I, II and III, $\boldsymbol{CB}$ is a part of $\boldsymbol{\alpha}$ and $\boldsymbol{D}$ is a point inside $\boldsymbol{CB}$ such that

1) $\boldsymbol{g}_1^{\boldsymbol{D}}(\boldsymbol{\alpha}) = \boldsymbol{g}_1^{\boldsymbol{B}}(\boldsymbol{\alpha})$
2) $\boldsymbol{g}_2^{\boldsymbol{D}}(\boldsymbol{\alpha}) = \boldsymbol{g}_2^{\boldsymbol{B}}(\boldsymbol{\alpha})$
3) $\boldsymbol{M}_{i,j}^{CD}(0) = \boldsymbol{M}_{i,j}^{CB}(0)$ $\boldsymbol{i}, \boldsymbol{j}$ = {0;1}

then it is possible to carry out processing of $\boldsymbol{x}$ for which instead of configuration $\boldsymbol{CB}$ configuration $\boldsymbol{CD}$ will appear ).

Lets divide the stack tape into consecutive zones $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \ldots$ each of size $\boldsymbol{N_1}$.

While processing $\boldsymbol{x}$ according to $\boldsymbol{\Delta}$ for every configuration created in a zone $\mathcal{K}_j$ at least one of the conditions I - III does not hold. Because at every moment of time no more that one of the zones cited above may be affected and head is located in one zone, then receiving of an input letter may affect one or more of the conditions I - III in no more than three zones. Therefore

$$s_{\mathcal{M}}(x,\Delta) \leq 3N_1 * |x| \qquad (1)$$

and thus

$$s_{\mathcal{M}}(x) \leq 3N_1 * |x|$$

In the following text while estimating the time function we will consider only resulting traces executed for which

1. every created configuration that satisfies conditions3 I,II,III has a length that doesn't exceed $N_1$ ( and therefore (1) holds ).
2. no empty cycles were allowed - if in the moment of time $t_1$ and $t_2$ : $t_1 > t_2$ $\mathcal{M}$ had the same configuration, then in the time frame $[t_1, t_2]$ an input symbols were provided.

Consider a word $x \in L(\mathcal{M})$ and such a resulting trace $\Delta$ of it execution.
Let $i$ to be a cell of the stack tape that is visited by the stack head while processing $x$ according to $\Delta$.
Consider zones $\mathcal{K}(j)=(i+(j-2)N_1+1;...;i+(j-1)N_1)$, $j=1,2,...$ . Configuration in the zone $\mathcal{K}(j)$:

- begin to take shape after the head gets into the cell $i+(j-2)N_1+1$ and right to the zone $\mathcal{K}(j-1)$ there are only empty cells.
- terminates when the head gets into the cell $i+(j-1)N_1$ and right to that cell there are only empty cells.

Note that during this process configuration in the zone $\mathcal{K}(j-2)$ is not affected ( otherwise the configuration $\mathcal{K}(j-1)$ would be wiped out ) and the head is not getting right from the zone $\mathcal{K}(j+1)$ ( otherwise the configuration would appear also in the zone $\mathcal{K}(j+1)$ ).
Because $|\mathcal{K}(j-1)\mathcal{K}(j)\mathcal{K}(j+1)|=3N_1$ and the number of configurations with the length not exceeding $3N_1$ is no more than $N_2 = \dfrac{S^{3N_1+1}-1}{S-1}$ , where $S$ is the number of symbols allowed by the SA $\mathcal{M}$ , then the number of configurations that appear while creating zone $\mathcal{K}(j)$ doesn't exceed $N_2 m(k_1(j)+1)$ where

$m$ - is number of states of the SA $\mathcal{M}$
$k_1(j)$ - number of input symbols received during creation of the zone $\mathcal{K}(j)$.

Similarly, the number of configurations that appear while erasing zone $\mathcal{K}(j)$ doesn't exceed $N_2 m(k_2(j)+1)$ where $k_2(j)$ is the number of input symbols received during erasing of the zone $\mathcal{K}(j)$. Therefore with each configuration in zone $\mathcal{K}(j)$ we can associate no more than $N_2 m(k(j)+1)$ configurations that appear while creating/erasing $\mathcal{K}(j)$, were $k(j)=k_1(j)+k_2(j)$ - is the number of input symbols received during creation/erasing of the zone $\mathcal{K}(j)$.

Assume that at some moment of processing $x$ according to $\Delta$ the full configuration $\alpha$ satisfies the following condition:

$$i+(j-2)N_1+1 \le |\alpha| \le i+(j-1)N_1$$

Then at that moment the configuration being processed ( created or erased ) is located in zone $\mathcal{K}(j-1)$ or zone $\mathcal{K}(j)$ or in zone $\mathcal{K}(j+1)$. Because while processing $x$ according to $\Delta$ there are no empty cycles,then the head may appear in the cell $i$ no more that $m(k_\alpha+1)$ cycles, where $k_\alpha$ is the number of input symbols received while configuration $\alpha$ did not change. Therefore during processing of a configuration in zone $\mathcal{K}(j)$ and during it erasing the head may be in the cell $i$ no more that

$$\sum_\alpha m(k_\alpha+1)=mk(j)+N_2m^2(k(j)+2)=k(j)(N_2m^2+m)+2N_2m^2 =t(i,j)$$

But for every $j$: $|\mathcal{K}(j)|=N_1$ and therefore inside zones $\mathcal{K}(0),\mathcal{K}(1)...$may appear no more than $3|x|$ configurations. Therefore the total time that head may be in the cell $i$ does not exceed

$$\begin{aligned} t(i)&=\sum_j t(i,j)=(N_2m^2+m)\sum_j k(j)+ 2N_2m^2\sum_j 1 \\ &\le(N_2m^2+m)|x|+2N_2m^2 3|x| \\ &=N_3|x| \end{aligned}$$

Therefore, $t_{\mathcal{M}}(x,\Delta)\le \sum_i t(i)\le N_3|x|s_{\mathcal{M}}(x,\Delta)\le const|x|^2$

That concludes the proof of the theorem 1.

# $3

Assume $\Sigma=\{0,1,\varepsilon\}$, word $x\in\Sigma^*$ and $a\in\Sigma$. Define $a(x)$ to be equal to the number of letters $a$ inside the word $x$. Furthermore define

$$T_n=\{x\in\Sigma^* \mid \varepsilon(x)=n \}$$
$$T(0)=\bigcup_{i=0}^{\infty} T_{2i}$$
$$T(1)=\bigcup_{i=0}^{\infty} T_{2i+1}$$

Define the language $L=\{x=x_1\varepsilon...x_{2k}\varepsilon\}$ such that:

1) $x_i\in\{0,1\}\{0,1\}^*$
2) $0(x_{2i})=0(x_1x_3...x_{2i-1})$; $1(x_{2i})=0$, $i=1,...,k-1$

3) $1(x_{2k})=1(x_1x_3...x_{2i-1})$; $1(x_{2k})=0$

It is easy to see that language $L$ is accepted even by a deterministic SA.

Theorem 2:

For every SA $\mathcal{M}$ such that $L$=L$(\mathcal{M})$ the following holds:

$\exists c$ for almost all $n$: $s_{\mathcal{M}}(n)\geq c*n$, $t_{\mathcal{M}}(n)\geq c*n^2$.

Proof:

Consider an arbitrary word $x=x_1\varepsilon...x_{2k}\in L$. Assume that while processing $x$ by the sequence $\Delta$ that accepts $x$, configuration that appears at the right of the point $A$ is erased while processing input sequence $\varepsilon x_{2k-2}x_{2k-1}\varepsilon x_{2k}\varepsilon$. Write the word $x$ as

$x = y_1y_2y_3$

with $y_2$ being the part of the word $x$ during processing of which configuration at the right of the point $A$ is created and erased; note that $y_3=\varepsilon x_{2k-2}x_{2k-1}\varepsilon x_{2k}\varepsilon$.

*a)*

Assume the points $B$ and $C$ located at right from the point $A$ and the following conditions are satisfied:

1) $g_1^{OB} = g_1^{OC}$
2) $g_2^{OB} = g_2^{OC}$
3) $M_{i,j}^{AB}(T(l)) = M_{i,j}^{AC}(T(l))$ $i,j,l$ = {0;1}

Let show that while creating and erasing configuration $BC$ while the head is located inside zone $OA$ and in zone $BC$, no input symbols are received.
Indeed, let for example while creation of the configuration in zone $BC$ when the head was located in zone $OA$ and in zone $BC$ received not an empty word $z$. Then for every $n\geq 0$ it is easy to build the resulting processing of a word $x_n=y_1y_2(n)y_3$ for which the configuration $OA\ AB\ (BC)^n$ appears and during creation of every configuration $BC$ the head being in the zone $OA$ and in the zone where the new configuration is created, works the same way as during creation $BC$. Therefore removing from $y_2(n)$ some letters will produce word $z_n$. But for every $n$ two last even blocks of $x_n$ equal $x_{2k-2}$ and $x_{2k}$. For $n$ being sufficiently big:

$$|z_n|>(|x_{2k-2}|+|x_{2k}|)^2$$

which is impossible.

*b)*

From *a)* we conclude that there is a sequence $\Delta'$ that accepts the word $x'=y_1y'_2y_3$ such that while processing $x'$ in accordance to $\Delta'$ right from the point $A$ there are not going to be points $B$ and $C$ that satisfy conditions 1), 2), 3) and while the head is located in $OA$ the processing symbols will be the same as processing $x$

in accordance to **Δ**. Therefore there exist $N_1$ such that while processing $x'$ in accordance to **Δ'** the configuration that is affected to the right of ***OA*** will not be larger than $N_1$.

Note that if before processing $x_{2k-2}$ at the time $t_1$ and $t_2$ : $t_1 > t_2$ SA $\mathcal{M}$, whie processing the word $x'$, has the same configurations then in the time interval $[t_1, t_2]$ no input symbols were provided: indeed if in the time interval $[t_1, t_2]$ not an empty word $z$ was provided then for every $n \geq 0$ it is easy to construct the accepting sequence of processing the word $yz_n \varepsilon x_{2k-2} x_{2k-1} \varepsilon x_{2k} \varepsilon$ which is, as shown above, is impossible.

Therefore there exist $N_2$ such that the total number of symbols processed when the head was at the right of ***A*** does not exceed $N_2$.
It is easy to see that the total number of symbols from the not-even blocks of the words $x$ and $x'$ that reside in $y_2$ and $y'_2$ is the same.

Note that if while processing $x$ in accordance to **Δ** while the head was in the zone ***OA*** the input word $z \in \{0,1\}^*$ was part of an not-even block of the word $x$ then while processing $x'$ in accordance to **Δ'** the word $z$ will be provided while the head is inside zine ***OA*** being part some not-even block of a word $x'$. Therefore, the total number of symbols from the not-even blocks of the word $x$ that were processed while head being at right from ***OA*** is equal to the total number of that symbols of the word $x'$ and therefore does not exceed $N_2$.

*c)*

From *b)* we conclude that while processing $x$ in accordance to **Δ** every word that is received when head was right of ***A*** contains no more that $N_2$ symbols $\varepsilon$. Therefore it is possible to create a sequence **Δ''** that accepts a certain word $x''=y_1 y''_2 y_3$ such that while processing $x''$ in accordance to **Δ''**

(a) right from the point ***A*** there are not going to be points ***B*** and ***C*** that satisfy conditions 1), 2), 3) and

(b) 4) $M_{i,j}^{AB}(T(l)) = M_{i,j}^{AC}(T(l))$ $i,j=\{0;1\}$ $l=0,...,N_2$

(c) while head being in ***OA*** the same symbols received as while processing $x$ in accordance to **Δ**.

(d) if while processing $x$ in accordance to **Δ** at the time $t$ of the visit of the head in the zone right of ***A*** the wort $z_t$ was processed, then while processing $x''$ in accordance to **Δ''** at the time of $t$s visit of the head inside the zone right of ***OA*** the processed word $z''_t$ will be such that $\varepsilon(z_t)=\varepsilon(z''_t)$.

The same as in *b)* for a word $x''$ we determine that there exist $N_3$ such that the total number of symbols of the word $x''$ received while the head was right of ***OA*** doesn't exceed $N_3$ and the total number of symbols of the words $y''_2$ and $y_2$ that are found in the not-even blocks of the words $x''$ and $x$ are the same.

Consider $x''=x''_1\varepsilon...\varepsilon x_{2k-3}\varepsilon x_{2k-2}\varepsilon x_{2k-1}\varepsilon x_{2k}$.

Denote $r_i = O(x_{2i-1})$, $r''_i = O(x''_{2i-1})$.

Then $r_i = r''_i + \gamma_i$. But $\sum_{i=1}^{k-1} r_i = O(x_{2k-2}) = \sum_{i=1}^{k-1} r''_i = \sum_{i=1}^{k-1} (r''_i + \gamma_i) = \sum_{i=1}^{k-1} r''_i + \sum_{i=1}^{k-1} \gamma_i$ .

Therefore $\sum_{i=1}^{k-1} \gamma_i = O$.

Assume that while processing $x$ in accordance to $\Delta$ while receiving block $x_{2i}$ of the word $x$ when the head was located at the right $OA$, not empty words $u_1,...,u_m$ were received and while processing $x''$ in accordance to $\Delta''$ while receiving block $x''_{2i}$ of the word $x''$ when the head was located at the right $OA$, not empty words $z_1,...,z_n$ were received. Because while receiving $x_{2i}$ and $x''_{2i}$ when the head was in $OA$ the same symbols are received, then $\sum_{j=1}^{m} u_j = \sum_{j=1}^{n} z_j + \gamma_i$. Therefore

$$\sum_i \sum_j u_j = \sum_i \sum_j z_j + \sum_i \gamma_i = \sum_i \sum_j z_j$$

and thus while processing $x$ in accordance to $\Delta$, the total number of symbols of the word $x$ that are received when the head was located at the right of $OA$ is equal to the total number of symbols of the word $x''$ that, while processing $x''$ in accordance to $\Delta''$, are received when the head was located at the right of $OA$ and therefore does not exceed $N_3$.

Assume that while processing $x$ in accordance to $\Delta$ for the points $B$ and $C$ located right from the point $A$, conditions 1), 2), 3) and
4) $L_{i,j}^{AB}(T) = L_{i,j}^{AC}(T)$ $i,j=\{0;1\}$ where $T=\{x \mid x \in \Sigma^* \&\& |x| \leq N_3\}$
are satisfied. Then from *a)* and *b)* we conclude that there is an execution sequence $\Delta_1$ processing the word $x$ such that does not create configuration $BC$ and while the head is located in $OA$ the processing of $x$ according to $\Delta$ and $\Delta_1$ is identical. Therefore exists $N_4$ and an executions sequence $\Delta_2$ processing the word $x$ such that while processing $x$ according to $\Delta_2$ at right of $OA$ the length of the configuration that is changed does not exceed $N_4$. It is clear that for all $x$ and $\Delta$

$$t_{\mathcal{M}}(x,\Delta_2) \leq const\, t_{\mathcal{M}}(x,\Delta)$$

Based on this we can build SA $\mathcal{M}'$ such that
$L(\mathcal{M}')=L(\mathcal{M})$ and $\mathcal{M}'$ accepts the words at the empty tape for every word $x=x_1\varepsilon...x_{2k} \in L$ and execution sequence $\Delta$ of $\mathcal{M}$ that accepts $x$ there exists execution sequence $\Delta'$ of $\mathcal{M}'$ that accepts $x$ such that while processing $x$ according to $\Delta'$ of $\mathcal{M}'$ before accepting $x_{2k-2}$ nothing is erased and

$$t_{\mathcal{M}}(x,\Delta') \leq const\, t_{\mathcal{M}}(x,\Delta)$$

Now we will evaluate the processing time of SA $\mathcal{M}'$.
For every configuration $\alpha$ that contains the bottom of the stack $O$

and created while processing a word $x \in L$ by the execution sequence $\Delta$ of $\mathscr{M}$ that accepts $x$, define a set

$$f(\alpha,x,\Delta)=(\ g_1(\alpha),\ r,\ g_2(\alpha),\ M_{1,1}^{\alpha}(T(0))\ ,\ M_{1,1}^{\alpha}(T(1))\ )$$

where

$g_1(\alpha)$ - is the state that the head ( processing $x$ according to $\Delta$ ) first leaves $\alpha$

$r$ - parity of number of symbols $\varepsilon$ than were processed so far

$g_2(\alpha)$ - is the state that the head ( processing $x$ according to $\Delta$ ) the last time enters $\alpha$ ( it is clear that at this moment right of $\alpha$ there are only empty cells )

Consider the set $T_f$ of the words $x=x_1\varepsilon...x_{2k} \in L$ such that while processing $x$ according to some execution sequence $\Delta(X)$ that accepts $x$

- the configuration $\alpha(x)$ that contains the bottom of the stack $0$ is created
- while processing some block $x_{2i}$, $i<k$ of the word $x$, $\alpha(x)$ is not visited by the head of the stack; denote $N(x,f)$ to be $|x_{2i}|$ and denote $f$ to be $f(\alpha,x,\Delta)$

Set $N_f = \min_{x \in Tf} N(x,f) = N(x_f,f)$ ,$N = \max_{f} N(x_f,f)$ .

Consider a word $x=x_1\varepsilon...x_{2k} \in L$ such that 0(x1)>N and an arbitrary execution sequence $\Delta$ that accepts $x$. Assume that while processing $x$ according to $\Delta$ the full configuration after processing $x_1$ is $\alpha$.
Let show that while processing $x$ according to $\Delta$ while reading every even block $x_{2i}$, $i<k$ of the word $x$ the head visits configuration $\alpha$.
Indeed, if that is not so then $x \in T_{f0}$ where

$$f_0=f(\alpha,x,\Delta)=(\ g_1(\alpha),\ r,\ g_2(\alpha),\ M_{1,1}^{\alpha}(T(0))\ ,\ M_{1,1}^{\alpha}(T(1))\ )$$

Lets organize SA $\mathscr{M}'$ in the following manner. <u>At first</u> we will process the initial parts of the word $x$ according to $\Delta$ till the head leaves $\alpha$ in the state $g_1(\alpha)$. Assume that that happened after processing the word $y_1$ ( note that $y_1$ can be written as $x_1\varepsilon z$, where $z \in \sum^*$ ). <u>After that</u> let process the sub word of $x_{f0}$ that is processed while processing $x_{f0}$ according to $\Delta(x_{f0})$ after the head left $\alpha(x_{f0})$ in the state $g_1(\alpha)$ till the first entrance of the head inside $\alpha(x_{f0})$ working according to $\Delta(x_{f0})$. Assume that after that while processing $x_{f0}$ according to $\Delta(x_{f0})$ the head leaves $\alpha(x_{f0})$ in the state $g_2$ and while all that the word $v$ is received. Because $M_{1,1}^{\alpha(xf0)}(T(i)) = M_{1,1}^{\alpha}(T(i))$ $i$=0,1, then in that case this can be achieved while providing the word $v'$ such that

$$\varepsilon(v')=\varepsilon(v)\ mod(2).$$

Continue this process; note that when at the right from $\alpha$ the even block of $x_{f0}$

will be provided ( that has a length $N_{f0}$ ) then this block will be even in the word $x'$ that was obtained as the result of the described process. But the first block of the word $x'$ is equal to $x_1$ and $\theta(x_1)>N\geq N_{f0}$.

Consider the word $x=x_1\varepsilon...\varepsilon x_{2k}\in L$ such that $x_1=0^{N+1}$ and $x_{2i+1}=1$, $i=1,...,k-2$
Let $\Delta$ to be an arbitrary execution sequence that accepts $x$. Assume that after processing $x_1$ ( while processing $x$ according to $\Delta$ ) a configuration $\alpha$ is created. The when processing every block $x_{2i}$, $i<k$ of the word $x$ the head visits $\alpha$ and therefore visits the cell $|\alpha|$. But while processing $x$ according to $\Delta$ $\mathcal{M}'$ before receiving $x_{2k-2}$ doesn't erase anything. Therefore, in order to avoid identical descriptions while processing $2m$ blocks of, the full configuration must increase ( $m$ being the number of states of SA $\mathcal{M}'$). Therefore after processing $x_1\varepsilon...x_{2i-1}$ the full configuration has a length not less that $\frac{i-1}{m}$ + $|\alpha|$ and thus during processing of the word $x_{2i-1-2m}\varepsilon...\varepsilon x_{2i-1}$ the head must visit the cell $\frac{i-1}{m}$ + $|\alpha|$.
But while processing $x_{2i}$ the head visits the cell $|\alpha|$ and therefore

$$t_{\mathcal{M}'}(x,\Delta)\geq \sum_{i=1}^{\frac{k-1}{m}} \frac{i-1}{m} \geq \frac{k^2}{8*m^3}$$

Also it is clear that $s_{\mathcal{M}'}(x,\Delta) \geq \frac{k}{m}$ .
Note that $|x|=N+1+(k-1)(N+1)+3k=k(N+4) \approx K$
Therefore $\exists c'$ such that $t_{\mathcal{M}'}(x,\Delta) \geq c'|x|^2$ and $s_{\mathcal{M}'}(x,\Delta) \geq c'|x|$

Let $n$ to be a sufficiently large number. Then exists $j$, $0 \leq j \leq N+4$ and a word $x=x_1\varepsilon...\varepsilon x_{2k}\in L$ such that

1) $x_1=0^{N+1}$
2) $x_{2i+1}=1$, $i=1,...,k-2$
3) $x2k-1=10j$
4) $|x|=n$

Then just like in b) we can establish that there is $c(j)$ such that for every execution sequence $\Delta$ that accepts $x$ the following holds:

$$t_{\mathcal{M}'}(x,\Delta)\geq c(j)|x|^2 \text{ and } s_{\mathcal{M}'}(x,\Delta) \geq c(j)|x|$$

Therefore, choosing $c = \min_j c(j)$, for almost all $n$

$$t_{\mathcal{M}}(n)\geq cn^2 \text{ and } s_{\mathcal{M}'}(n) \geq cn$$

That concludes the proof of the theorem 2.

Note that the results of the theorems 1 and 2 that were obtained for non deterministic SA can easily be extended to obtain the similar results for deterministic SA.

## Acknowledgments

Would like to thank M.K. Valiev and V.A. Nepomniashy for interesting discussions during the work on this article.

## References

1. S. Ginsburg, S. Greiback, M. Harrison. “Stack automata and compiling”, J.ACM 14, 182-201(1967).
2. S. Ginsburg, S. Greiback, M. Harrison. “One-way stack automata”, J.ACM 14, 389-418(1967).
3. S. Greiback. “Checking automata and one-way stack languages”, J. Comp and System Sciences v.3 n.2, 196-217(1969).
4. J. Hopcroft, J. Ullman. “Sets accepted by one-way stack automata are context-sensitive”, Information and Control v.13 n.2, 114-133217(1968).