RELATIONSHIPS BETWEEN PUSHDOWN AUTOMATA AND TAPE-BOUNDED TURING MACHINES

BURKHARD MONIEN
INSTITUT FÜR INFORMATIK DER UNIVERSITÄT HAMBURG
HAMBURG, DEUTSCHLAND

Abstract: In this paper we will show that any language accepted by a $L(n)$ tape-bounded and $T(n)$ time-bounded nondeterministic aux PDA also is accepted by a $L(n)\cdot\log_2 T(n)$ tape-bounded deterministic Turing machine. We will give some applications of this result concerning the tape complexity of classes of formal languages.

1. THE THEOREMS

Little is known about the amount of tape a Turing machine needs to accept a given language. An outstanding result in this area was obtained by Hartmanis-Lewis-Stearns (1965) who showed that each context-free language is accepted by a $(\log_2 n)^2$ tape-bounded deterministic Turing machine. By our main theorem which is a generalization of the Hartmanis-Lewis-Stearns result we hope to gain further insight into these problems.

S.A.Cook (1971) defined the conception of an aux PDA and showed the equivalence between $L(n)$ tape-bounded aux PDA and $2^{d\cdot L(n)}$ time-bounded deterministic Turing machines.

A k-tape deterministic (nondeterministic) aux PDA $M = (S,T,\delta,F,\gamma_0,s_0)$ consists of a finite control (S set of states, $s_0 \in S$ start state, $F \subset S$ set of final states), an input tape with read-only head, k tapes with read-write heads and a pushdown tape (T set of tape symbols, $\gamma_0 \in T$ start symbol on the pushdown tape). The next move function δ is deterministic (nondeterministic).

A configuration of a 1-tape aux PDA M is a 6-tupel (s,i,w,j,v,u), where $s \in S$ is the state of M; $i \in \{1,..,l(w)\}$ and $j \in \mathbb{N}$ are the positions of the input head and the head of

the working tape ; $w,v,u \in T^*$ are the insriptions of the input tape, the working tape and the pushdown tape.
The next move function δ implies a mapping
$(s,i,w,j,v,ua) \longmapsto (s',i+\mu,w,j+\eta,v',u\xi)$, where $a \in T$,
$\xi \in \{\varepsilon, a \in T\}$ and $\mu,\eta \in \{-1,0,+1\}$.
$\longmapsto^*$ is the transitive closure of $\longmapsto$.

A word $w \in T^*$ is accepted by M , if and only if there exist $t \in F$, $v \in T^*$ and $i,j \in \mathbb{N}$ such that
$(s_0,1,w,1,\varepsilon,\gamma_0) \longmapsto^* (t,i,w,j,v,\varepsilon)$.
M is called a $L(n)$ tape-bounded and $T(n)$ time bounded aux PDA if and only if for each w, accepted by M, there exist a sequence of at most $T(l(w))$ moves leading to a final state and to an empty pushdown tape such that during this computation no more than $L(l(w))$ cells are used on the working tape.
All these definitions can easily be transfered to the case of the k-tape aux PDA.

Theorem 1 Suppose a set L is accepted by a $L(n)$ tape-bounded and $T(n)$ time-bounded nondeterministic aux PDA. Suppose $L(n)$ and $\log_2 T(n)$ are functions easily computable in the sense which is defined below and $L(n) \geq \log_2 n$ for all $n \in \mathbb{N}$. Then there exists a $L(n)\cdot\log_2 T(n)$ tape-bounded deterministic Turing machine that accepts L .

This theorem will be proved in the next paragraph.

Each $L(n)$ tape-bounded nondeterministic Turing machine is $2^{d\cdot L(n)}$ time-bounded. Therefore theorem 1 implies the theorem of W.J.Savitch (1970).
Theorem 2 Each set L accepted by a $L(n)$ tape-bounded nondeterministic Turing machine , where $L(n) \geq \log_2 n$ for all $n \in \mathbb{N}$, is accepted by a $L(n)^2$ tape-bounded deterministic Turing machine.

Because of S.A.Cook's results each $T(n)$ time-bounded deterministic Turing machine may be simulated by a $\log_2 T(n)$ tape-bounded aux PDA. But a $L(n)$ tape-bounded aux PDA in general needs $2^{2^{d\cdot L(n)}}$ moves to accept a word w , $l(w) = n$, and that is why no relationships between time and tape complexity classes of Turing machines are obtained by theorem 1 .

Because of the Greibach normal form theorem each context-free language is accepted by a pushdown automaton which moves its head in each step one cell to the right. This automaton is a $L(n) = \log_2 n$ tape-bounded and $T(n) = n$ time-bounded nondeterministic aux PDA. Thus we proved :

Theorem 3 (Hartmanis-Lewis-Stearns (1965))
Each context-free language is accepted by a $(\log_2 n)^2$ tape-bounded deterministic Turing machine.

By means of the results of Harrison-Ibarra (1968) and the Greibach normal form theorem it is easy to prove that each nondeterministic k-head 1-way pushdown automaton is equivalent to a nondeterministic k-head 1-way pushdown automaton which moves in each step at least one of its heads one cell to the right. Theorem 1 therefore implies the following :

Theorem 4 Each language which is acceptable by a nondeterministic k-head 1-way pushdown automaton is accepted by a $(\log_2 n)^2$ tape-bounded deterministic Turing machine.

O.H.Ibarra (1970) defined n - simple matrix languages. To each n there a exists a k_n such that any n - simple matrix language with a right endmarker is acceptable by a nondeterministic k_n-head 1-way pushdown automaton. If $L(n) \geq \log_2 n$, then the tape complexity $L(n)$ of a language is independent of the use of endmarkers. We get from theorem 4 :

Theorem 5 Each n - simple matrix language is accepted by a $(\log_2 n)^2$ tape-bounded deterministic Turing machine.

Because of S.A.Cook's results each language is acceptable by an aux PDA using only a small amount of tape. In general however this aux PDA needs an enormous number of moves to accept a word. Theorem 1 shows that in studying classes of formal languages it is interesting to investigate whether these languages are acceptable by aux PDA's using only a small amount of tape and of time.

2. PROOF OF THEOREM 1

We use quite the same method as Hartmanis-Lewis-Stearns (1965) and W.J.Savitch (1970) .

The $L(n)$ tape-bounded and $T(n)$ time-bounded k-tape nondeterministic aux PDA which accepts L may be simulated by a $L(n)$ tape-bounded and $T(n)^2$ time-bounded 1-tape nondeterministic aux PDA $M = (S,T,\delta,F,\gamma_0,s_0)$.

For each $w = a_1 \,..\, a_n \in T^*$, $l(w) = n$, we define a set $X = X_w =$ $= S\times\{1,..,n\}\times T\times\{1,..,L(n)\}\times T^{L(n)}\times S\times\{1,..,n\}\times\{1,..,L(n)\}\times T^{L(n)}$ and a partial mapping $V = V_w : X\times X \longrightarrow 2^X$.

Suppose $x = (s,i,\gamma,j,v,s',i',j',v') \in X$

$y = (s_1,i_1,\gamma_1,j_1,v_1,s_1',i_1',j_1',v_1') \in X$

Then $z = (s_2,i_2,\gamma_2,j_2,v_2,s_2',i_2',j_2',v_2') \in V(x,y)$ if and only if one of the following conditions is true.

(i) $\gamma_2 = \gamma_1$, $(s_2,i_2,w,j_2,v_2,\gamma_2) \longmapsto (s,i,w,j,v,\gamma_2\gamma)$

$s' = s_1$,$i' = i_1$,$j' = j_1$,$v' = v_1$,$s_1' = s_2'$,$i_1' = i_2'$,$j_1'=j_2'$,$v_1'=v_2'$

(ii) $\gamma_2 = \gamma$, $(s_2,i_2,w,j_2,v_2,\gamma_2) \longmapsto (s_1,i_1,w,j_1,v_1,\gamma_2\gamma_1)$

$s = s_1'$,$i = i_1'$,$j = j_1'$,$v = v_1'$,$s' = s_2'$,$i' = i_2'$,$j' = j_2'$,$v' = v_2'$

$z \in V(x,y)$ is equivalent to Cook's notation : x and y generate z

Partial mappings $t = t_w : X \longrightarrow \mathbb{N}$, $d = d_w : X\times X \longrightarrow \mathbb{N}\cup\{0\}$ are defined for each $w \in T^*$ by :

(1) If $x = (s,i,\gamma,j,v,s',i',j',v')$ and $(s,i,w,j,v,\gamma) \longmapsto (s',i',w,j',v',\varepsilon)$ then $t(x) = 1$, otherwise $t(x) = \min\{t(z) + t(y) \mid y,z \in X$, $x \in V(y,z)$, $t(x)$ and $t(y)$ are defined $\}$

(2) If $x = y$ then $d(x,y) = 0$, otherwise $d(x,y) = \min\{d(z,y) + t(u) \mid u,z \in X$, $x \in V(u,z)$, $d(z,y)$ and $t(u)$ are defined $\}$

$t(x)$ is defined if and only if $x = (s,i,\gamma,j,v,s',i',j',v')$ and $(s,i,w,j,v,\gamma) \longmapsto^* (s',i',w,j',v',\varepsilon)$. $t(x)$ is a lower bound for the minimal number of steps $\longmapsto^*$ is composed of.

If $y = (s,i,\gamma,j,v,s',i',j',v')$ and $x = (s_1,i_1,\gamma_1,j_1,v_1,s_1',i_1',j_1',v_1')$ then $d(x,y)$ is defined iff there exists a string $\varphi \in T^*$ such that $(s_1,i_1,w,j_1,v_1,\gamma_1) \longmapsto^* (s,i,w,j,v,\varphi\gamma) \longmapsto^*$

$\longmapsto^* (s',i',w,j',v',\varphi) \longmapsto^* (s_1',i_1',w,j_1',v_1',\varepsilon)$

In this case $d(x,y)$ is a lower bound for the minimal number of steps the first and third $\longmapsto^*$ are composed of.

We have assumed that $L(n) \geq \log_2 n$ for all $n \in \mathbb{N}$ .If n is great enough then no more than $5\cdot L(n)$ cells are necessary in order to

store an element $x \in X$.
We number the elements of X , starting with 1 , and write $x \hat{=} m$ if m is the number assigned to x . Then there exists a $d \in \mathbb{N}$ such that $m \leq N = 2^{d \cdot L(n)}$ if $m \hat{=} x$ for some $x \in X$.

We define a deterministic Turing machine M_1 that accepts L and uses for each $w \in T^*$, $l(w) = n$, on each tape at most $(1 + d \cdot L(n)) \cdot 3 \cdot \log_2 T(n)$ cells.
Tape 1 of M_1 is divided into $r = 3 \cdot \log_2 T(n)$ segments of length $(1 + d \cdot L(n))$. In each segment is stored a pair (x,B) where $x \in \{1,\dots,N\}$ is the number of an element of X and $B \in \{0,1\}$.
Tape 1 always stores a sequence $(x_1,B_1)(x_2,B_2) \dots (x_r,B_r)$.

We demand that $L(n)$ and $\log_2 T(n)$ are functions easily computable in the following sense : If M_1 has w , $l(w) = n$, on its input tape , it is able to compute $L(n)$ and $\log_2 T(n)$ using no more than $L(n) \cdot \log_2 T(n)$ cells on the tapes.

M_1 starts the following algorithm :

A. M_1 writes $\underbrace{(1,0)(1,0) \dots (1,0)}_{r}$ on tape 1 .

B. If $B_\mu = 1$ for all $\mu = 1,\dots,r$, goto D.
Otherwise let ν be the smallest index such that $B_\nu = 0$.
If $x_\nu \hat{=} (s,i,\gamma,j,v,s',i',j',v')$ and if $(s,i,w,j,v,\gamma) \longmapsto (s',i',w,j',v',\varepsilon)$ then goto C.
If there are $\chi,\eta \in \{1,\dots,r\}$ such that $B_\chi = B_\eta = 1$ and $x_\nu \in V(x_\chi,x_\eta)$ then goto C.
Otherwise goto D.

C. ν as in B.
If $x_\nu \hat{=} (s_0,1,\gamma_0,1,\varepsilon,t,i,j,v)$ with $t \in F$ then stop the algorithm and accept w.
Otherwise : $B_\nu = 1$; $x_\mu = 1$, $B_\mu = 0$ for all $\mu = 1,\dots,\nu-1$
Goto B.

D. ν as in B.
If $x_\mu = N$ for all $\mu = 1,\dots,r$ then stop the algorithm and reject w .
Otherwise let χ be the smallest index such that $x_\chi \neq N$.
Set $x_\chi = x_\chi + 1$ and $x_\mu = 1$, $B_\mu = 0$ for all $\mu = 1,\dots,\chi-1$.
Goto B.

The following lemma shows how M_1 works.

Lemma 1 Suppose $i \in \mathbb{N}, w \in T^*, X = X_w, V = V_w, t = t_w, d = d_w$.

(1) Suppose $x,y,z \in X$ and $t(x) \leq 2^{i-1}$, $t(y) \leq 2^i - 1$, $t(z) \leq 2^i - 1$.
If the inscription of tape 1 has the form
$\underbrace{(1,0)\ \dots\ (1,0)}_{3i}(x_{3i+1},B_{3i+1})\ \dots\ (x_r,B_r)$
then tape 1 stores after a finite number of moves of M_1 :
$\underbrace{(1,0)\ \dots\ (1,0)}_{3i-3}(x,1)(y,1)(z,1)(x_{3i+1},B_{3i+1})\ \dots\ (x_r,B_r)$

(2) Suppose tape 1 stores
$\underbrace{(1,0)\dots(1,0)}_{3i}(x_{3i+1},B_{3i+1})\dots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\dots(x_r,B_r)$
Suppose $B_\nu = 1$ for all $\nu = 3i+1, \dots, q$ and there exists $\mu \in \{3i+1, \dots, r\}$ such that $B_\mu = 1$ and $d(x,x_\mu) \leq 2^i - 1$.
Then tape 1 stores after a finite number of moves of M_1 :
$\underbrace{(1,0)\ \dots\ (1,0)}_{q}(x,1)(x_{q+2},B_{q+2})\ \dots\ (x_r,B_r)$

Proof: In our proof we only consider what is written on tape 1 . The other tapes are used to do auxiliary computations.
If starting with the insription $(x_1,B_1)(x_2,B_2)\ \dots\ (x_r,B_r)$ M_1 reaches the insription $(x_1',B_1')(x_2',B_2')\ \dots\ (x_r',B_r')$ in a finite number of moves , then we write
$(x_1,B_1)\ \dots\ (x_r,B_r) \Longmapsto (x_1',B_1')\ \dots\ (x_r',B_r')$.

(1) and (2) are proved simultaneously by induction .

i = 1

$2^{i-1} = 2^i - 1 = 1$. Therefore (1) is obvious. To prove (2) we first mention that because of $d(x,x_\mu) = 1$ there is a $y \in X$ such that $t(y) = 1$ and $x \in V(x_\mu,y)$. M_1 works as follows :

$(1,0)(1,0)(1,0)(x_4,B_4)\dots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\dots(x_r,B_r)$
$\Longmapsto (1,0)(1,0)(y,1)(x_4,B_4)\dots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\dots(x_r,B_r)$
$\Longmapsto \underbrace{(1,0)\ \dots\ (1,0)}_{q}(x,1)(x_{q+2},B_{q+2})\ \dots\ (x_r,B_r)$

Now let $i \in \mathbb{N}$ be a number such that (1) and (2) are true for all natural numbers less or equal than i . We want to prove (1) and (2) for the number i+1 .

(1) Suppose $y \in X$ and $t(y) \leq 2^{i+1} - 1$.
Then there exist $y_0,\dots,y_m,z_0,\dots,z_m \in X$ such that $z_0 = y_0$,

$y_k \in V(y_{k-1},z_k)$ for $k = 1,..,m$, $y_m = y$ and
$t(y_0) \le 2^i - 1$, $t(y_1) \ge 2^i$, $t(z_1) \le 2^i - 1$.
Suppose $x_{3i+3} \in X$, $B_{3i+3} \in \{0,1\}$ are arbitrary elements.
Because of our assumption,(1), M_1 works in the following way :

$$\underbrace{(1,0)\ \dots\ (1,0)}_{3i+2}(x_{3i+3},B_{3i+3})(x_{3i+4},B_{3i+4})\ \dots\ (x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)\ \dots\ (1,0)}_{3i}(y_1,0)(y,0)(x_{3i+3},B_{3i+3})\ \dots\ (x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)\dots(1,0)}_{3i-2}(y_0,1)(z_1,1)(y_1,0)(y,0)(x_{3i+3},B_{3i+3})\dots(x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)\ \dots\ (1,0)}_{3i}(y_1,1)(y,0)(x_{3i+3}B_{3i+3})\dots\ (x_r,B_r)$$

Because $d(y,y_1) = t(y) - t(y_1) \le 2^{i+1} - 1 - 2^i = 2^i - 1$
our assumption,(2),implies that M_1 changes the above inscription
of tape 1 and gets

$$(1,0)\ \dots\ (1,0)(y,1)(x_{3i+3},B_{3i+3})\ \dots\ (x_r,B_r)$$

It is easy to see that the following is true also.

$$\underbrace{(1,0)\ \dots\ (1,0)}_{3i+3}(x_{3i+4},B_{3i+4})\ \dots\ (x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)\ \dots\ (1,0)}_{3i}(x,0)(y,1)(z,1)(x_{3i+4},B_{3i+4})\ \dots\ (x_r,B_r)$$

where $t(x) \le 2^i$, $t(y) \le 2^{i+1} - 1$, $t(z) \le 2^{i+1} - 1$.
There exist $u,v \in X$ such that $x \in V(u,v)$ and $t(u) \le 2^i - 1$,
$t(v) \le 2^i - 1$. Because of our assumption,(1),M_1 does

$$\underbrace{(1,0)\ \dots\ (1,0)}_{3i}(x,0)(y,1)(z,1)(x_{3i+4},B_{3i+4})\ \dots\ (x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)..(1,0)}_{3i-2}(u,1)(v,1)(x,0)(y,1)(z,1)(x_{3i+4},B_{3i+4})..(x_r,B_r)$$

$$\Longmapsto \underbrace{(1,0)\ \dots\ (1,0)}_{3i}(x,1)(y,1)(z,1)(x_{3i+4},B_{3i+4})\ \dots\ (x_r,B_r)$$

(2) Suppose $x,x_\mu \in X$ and $d(x,x_\mu) \le 2^{i+1} - 1$.
Then there exist $y_1,..,y_m,z_0,..,z_m \in X$ such that $z_0 = x_\mu$, $z_m = y$ and
$z_k \in V(z_{k-1},y_k)$ for $k = 1,..,m$,
$\sum_{k=1}^{m} t(y_k) = d(x,x_\mu) \le 2^{i+1} - 1$.
If $d(x,x_\mu) \le 2^i - 1$ then because of our assumption there is
nothing to prove.Therefore we assume that $d(x,x_\mu) \ge 2^i$
We have to consider two cases.
a.) $t(y_k) \le 2^i - 1$ for all $k = 1,..,m$.
A number $p \in \{1,..,m\}$ is defined by

$\sum_{k=1}^{p} t(y_k) \le 2^i - 1$ and $\sum_{k=1}^{p+1} t(y_k) \ge 2^i$.

Because of

$$\sum_{k=p+2}^{m} t(y_k) = \sum_{k=1}^{m} t(y_k) - \sum_{k=1}^{p+1} t(y_k) \le 2^{i+1} - 1 - 2^i = 2^i - 1$$

the following relations are valid :

$d(x,z_{p+1}) \le 2^i - 1$, $d(z_{p+1},z_p) = t(y_{p+1}) \le 2^i - 1$ and $d(z_p,x_\mu) \le 2^i - 1$.

Our assumption,(2),therefore implies :

$$\underbrace{(1,0)\ldots(1,0)}_{3i+3}(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow \underbrace{(1,0)\ldots(1,0)}_{3i+2}(z_p,1)(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow \underbrace{(1,0)\ldots(1,0)}_{3i+1}(z_{p+1},1)(z_p,1)(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow (1,0)\ \ldots\ (1,0)(x,1)(x_{q+2},B_{q+2})\ \ldots\ (x_r,B_r)$$

b.) There is a $p \in \{1,..,m\}$ such that $t(y_p) \ge 2^i$.

$d(x,x_\mu) = d(x,z_p) + d(z_p,z_{p-1}) + d(z_{p-1},x_\mu)$

This implies :

$$d(x,z_p) + d(z_{p-1},x\ ) = d(x,x_\mu) - d(z_p,z_{p-1}) \le 2^{i+1} - 1 - 2^i = 2^i - 1$$

Our assumption,(2),and what we have proved in (1) therefore shows :

$$\underbrace{(1,0)\ldots(1,0)}_{3i+3}(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow \underbrace{(1,0)\ldots(1,0)}_{3i+2}(y_p,1)(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow \underbrace{(1,0)\ldots(1,0)}_{3i+1}(z_{p-1},1)(y_p,1)(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow \underbrace{(1,0)\ldots(1,0)}_{3i}(z_p,1)(z_{p-1},1)(y_p,1)(x_{3i+4},B_{3i+4})\ldots(x_q,B_q)(x,0)(x_{q+2},B_{q+2})\ldots(x_r,B_r)$$

$$\models\!\!\Rightarrow (1,0)\ \ldots\ (1,0)(x,1)(x_{q+2},B_{q+2})\ \ldots\ (x_r,B_r)$$

q.e.d.

It is obvious that M_1 is a $(1 + d\cdot L(n))\cdot 3\cdot \log_2 T(n)$ tape-bounded deterministic Turing machine. All we have to show is that M_1 accepts L .

If $w \notin L$ then our algorithm never leads to a pair $(x,1)$ where $x = (s_0,1,\gamma_0,1,\varepsilon,t,i,j,v)$ with $t \in F$ and therefore M_1 rejects w.

If $w \in L$ then there exists a $x = (s_0,1,\gamma_0,1,\varepsilon,t,i,j,v)$ with $t \in F$ such that $t(x) \leq T(l(w))$ and during the corresponding computation no more than $L(l(w))$ cells are used on the working tape.

Because of Lemma 1 M_1 needs at most $3 \cdot \log_2 t(x) \leq 3 \cdot \log_2 T(l(w))$ segments to compute $(x,1)$. Therefore M_1 accepts w.

As the tape complexity of a language is independent of a constant factor theorem 1 is proved.

q.e.d.

References :

S.A.Cook (1971) Characterizations of Pushdown Machines in Terms of Time-Bounded Computers , JACM 18 , 4

Harrison-Ibarra (1968) Multi-Tape and Multi-Head Pushdown Automata , Inf.Contr. 13 , 433

Hartmanis-Lewis-Stearns (1965) Memory bounds for the Recognition of Context-Free and Context-Sensitive Languages , IEEE Conf.Rec. Switch.Th.,191

O.H.Ibarra (1970) Simple Matrix Languages , Inf.Contr. 17 , 359

W.J.Savitch (1970) Relationships Between Nondeterministic and Deterministic Tape Complexities , JCSS 4 , 177