ON THE TIME AND TAPE COMPLEXITY OF LANGUAGES I

H. B. Hunt III[†]

Cornell University
Ithaca, New York

ABSTRACT

We investigate the following:

(1) the relationship between the classes of languages accepted by deterministic and nondeterministic polynomial time bounded Turing machines;

(2) the time and tape complexity of many predicates on the regular sets;

(3) the relationship between the classes of languages accepted by deterministic or nondeterministic polynomial time bounded Turing machines and the class of languages accepted by polynomial tape bounded Turing machines; and

(4) the complexity of many predicates about stack automata.

We find several problems with nonpolynomial lower complexity bounds.

§1. INTRODUCTION

Before describing our results we need several definitions and facts.

Definition 1.1: PTIME is the class of all languages (over some countably infinite alphabet $\bar{\Sigma}$) recognized by some deterministic polynomial time bounded Turing machine.

Definition 1.2: NPTIME is the class of all languages (over some countably infinite alphabet $\bar{\Sigma}$) recognized by some nondeterministic polynomial time bounded Turing machine.

Definition 1.3: PTAPE is the class of all languages (over some countably infinite alphabet $\bar{\Sigma}$) recognized by some deterministic or nondeterministic polynomial tape bounded Turing machine.

In [17] Savitch showed that every L(n)-tape bounded nondeterministic Tm can be simultated by an $(L(n))^2$-tape bounded deterministic Tm, provided $L(n) \geq \log_2(n)$. In particular, this implies that the classes of languages accepted by deterministic and nondeterministic polynomial tape bounded Turing machines are the same.

Definition 1.4: Let $\Sigma = \{0,1\}$. Let Π be the class of functions from Σ^* into Σ^* computable by some polynomial time bounded Tm. Let L and M be languages. Then $L \underset{\text{PTIME}}{\leq} M$

[†]This research was supported by a National Science Foundation Graduate Fellowship in Computer Science.

(L is p-reducible to M) if there is a function $f \in \Pi$ such that $x \in L \iff f(x) \in M$. A language M is p-complete if M (or $\bar{M}$) $\in$NPTIME and every language in NPTIME is p-reducible to M.

In Section 2 we find several necessary and sufficient conditions for PTIME to equal NPTIME. We find p-complete proper subproblems of Karp's p-hard problems (See Section 2 or Karp [13]). We also find several p-complete problems of linear nondeterministic time complexity.

In Section 3 we study the complexity of predicates on the regular sets when the regular sets are enumerated by

(1) regular expressions, nondeterministic f.s.a., or type 3 grammars,

(2) regular expressions with $\cap$,

(3) regular expressions with "squaring", or

(4) extended regular expressions.

In Section 4 we study the complexity of problems about stack automata such as emptiness, membership, equivalence, etc. Section 5 is a short conclusion.

§2. PTIME

Definition 2.0: Ndtime(L(n)) is the class of all languages (over some countably infinite alphabet $\bar{\Sigma}$) recognized by some nondeterministic L(n) time bounded Tm, Dtime(L(n)) is defined analogously.

Before stating the main results we need two lemmas.

Lemma 2.1: For all $k \geq 1$, there is a 2 tape Tm M_k such that for all $w \in \Sigma^+$, $L(M_k) = \{x \mid x = w\,\text{¢}\,1^{|w|^{2^k}}\,\$$ with ¢, $\$ \notin \Sigma\}$. Moreover, M_k operates within linear time.

Lemma 2.2: For all $k \geq 1$, there is a single tape Tm M_k such that for all $w \in \Sigma^+$, $L(M_k) = \{x \mid x = w\,\text{¢}\,1^{|w|^{2^k}}\,\$$ with ¢, $\$ \notin \Sigma\}$. Moreover, M_k operates within time $O(|x|\log|x|)$.

Both proofs may be found in [11]. Both use counters.

Theorem 2.3: The following are equivalent:

(1) PTIME = NPTIME;

(2) all linear time nondeterministic 2-tape Tm languages $\in$PTIME;

(3) all $n\log n$ time nondeterministic single tape Tm language $\in$PTIME;

(4) the language $\mathscr{L} = \{(\alpha,\beta) \mid \alpha$ and β are regular expressions over $\{0,1\}$ with no occurrence of * and $L(\alpha) = L(\beta)\} \in$ PTIME;

(5) the language $\mathscr{L} = \{(\alpha,\beta) \mid \alpha$ and β are nondeterministic f.s.a., $L(\alpha) = L(\beta)$, and $L(\alpha)$ and $L(\beta)$ are finite$\} \in$ PTIME; and

(6) there is a positive integer $k > 0$ such that for all functions $f(n) \geq n$, ndtime $(f(n)) \subseteq$ dtime $([f(n)]^k)$.

Proof of 2.3: The proof of the equivalence of (1) through (5) may be found in [11]. We sketch the proofs of the equivalence of (1) and (4), and of (1) and (6).

(1) => (4): Given a regular expression β and a string x, one can check if $x \in L(\beta)$ in det. polynomial time. β a "U" and "·" regular expression implies $\forall x \in L(\beta)$, $|x| \leq |\beta|$ (Here β is a string over the alphabet $(\{(,),\cdot, U,0,1\})$.

Hence given two "U" and "·" regular expressions α and β, to verify $L(\alpha) \neq L(\beta)$ we need only nondeterministically guess a string x of length $\leq \max(|\alpha|,|\beta|)$ s.t. $x \in L(\alpha) \cap \overline{L(\beta)}$ or $x \in \overline{L(\alpha)} \cap L(\beta)$.

(4) => (1): Cook [4] has shown that PTIME = NPTIME iff D_3- tautology is an element of PTIME. Let $f = \bigvee_{i=1}^{m} c_i$ be a D_3- Boolean form. Then each c_i is the product of at most three literals. For each c_i let $\beta_i = \beta_{i1}\,\beta_{i2}\,,\ldots,\,\beta_{in}$ where

$$\beta_{ij} = \begin{cases} (0+1) & \text{if } x_j \text{ and } \bar{x}_j \text{ are not literals in } c_i, \\ 0 & \text{if } \bar{x}_j \text{ is a literal in } c_i, \text{ and} \\ 1 & \text{if } x_j \text{ is a literal in } c_i. \end{cases}$$

Let $\beta = \bigcup_{i=1}^{m} \beta_i$. Then $y \in L(\beta)$ iff y satisfies some clause c_i. Thus, $L(\beta) = \{0,1\}^n$ iff f is a tautology.

(1) => (6): We note that Cook's proof in [4], that D_3-tautology is p-complete, actually says that there is a k s.t. if a nondeterministic Tm T takes time $\leq t(n)$, then there is a Boolean form A(n) s.t.

(i) $|A(n)| \leq 0([t(n)]^k)$ and

(ii) A(n) is a tautology iff T accepts its input.

Our algorithm proceeds as follows:

For i = n step 1 do

(i) write out A(i)

(ii) if A(i) is a tautology then halt;

The time required $\leq 0\left(\sum_{j=n}^{t(n)} \{\text{time to execute (i) and (ii) for jth iterations}\}\right)$.

But by assumption (ii) requires deterministic time $O(|A(i)|^{\ell})$.

$\therefore$ The time required $\leq 0\,[t(n) \cdot t(n)^{k\cdot\ell}]$.

Corollary 2.4: There are p-complete problems of nondeterministic time complexity $0(n \log n)$ on some single tape Tm and of linear nondeterministic time complexity on some 2 tape Tm.

Proof: This follows since there are padded universal polynomial Tms of this complexity. For a complete proof see [11].

Theorem 2.5: (Book and Greibach [2]). $\mathscr{L}$ is recognized by a linear time nondeterministic multi-tape Tm iff $\mathscr{L}$ is the ϵ-free homomorphic image of the intersection of three context-free languages.

As an immediate consequence of Corollary 2.4 and Theorem 2.5 we get the following:

Corollary 2.6: There is an ϵ-free homomorphism h_0 and three context-free languages L_0, L_1, L_2, s.t. $h_0(L_0 \cap L_1 \cap L_2)$ is a p-complete problem of linear nondeterministic time complexity; and

Corollary 2.7: PTIME = NPTIME iff PTIME is closed under ϵ-free homomorphism.

This suggests that ϵ-free homomorphism plays an important role in the interrelationship between PTIME and NPTIME. Informally, as the reader will see below closure under ϵ-free homomorphism corresponds to non-determinism.

Theorem 2.7: Let $\mathscr{L} = \{f\ c\ x \mid f$ is a D_3-Boolian form expressed as a string over some finite alphabet Σ, $c, 0, 1 \notin \Sigma$. $x = x_1 \cdot x_2 \cdot \ldots \cdot x_n \in \{0,1\}^n$ for some $n > 0$. f is a function of the n variables $t_1,\ldots,t_n$ and $f(x_1,\ldots,x_n)$ is false$\}$. Let the ϵ-free homomorphism h_0 be defined as follows:

i) $h_0: [\Sigma \cup \{c,0,1\}]^* \to [\Sigma \cup \{c,0,1\}]^*$,

ii) $\forall\ \sigma \in \Sigma \cup \{c\}$, $h_0(\sigma) = \sigma$,

iii) $h_0(0) = 0$, and

iv) $h_0(1) = 0$.

Then $\mathscr{L} \in$ PTIME and $h_0(\mathscr{L})$ is p-complete.

Proof: It is obvious that $\mathscr{L} \in$ PTIME. We prove $f\ c\ 0^n \in h_0(\mathscr{L})$ iff f is not a tautology. $f\ c\ 0^n \in h_0(\mathscr{L}) \Rightarrow \exists$ an $x = x_1 \ldots x_n \in \{0,1\}^n$ such that $f\ c\ x \in \mathscr{L}$. But $f\ c\ x \in \mathscr{L} \Rightarrow f(x_1,\ldots,x_n)$ is false. f is not a tautology $\Rightarrow$ that there is an assignment of values $(x_1,\ldots,x_n)$ to the n variables of f s.t. $f(x_1,\ldots,x_n)$ is false. $\therefore f\ c\ x_1 \ldots y_n \in \mathscr{L}$ and $f\ c\ 0^n \in h_0(\mathscr{L})$.

In [13] Karp shows that NPTIME is p-reducible to the problems (1) Equivalence of Regular Expressions, (2) Equivalence of Nondeterministic Finite State Automata, and (3) Context-Sensitive Recognition. He was unable to show any of these problems to be p-complete. We have given proper subproblems of 1 and 2 which are p-complete. Since all O(n log n) or linear time bounded languages are context-sensitive, we have found proper p-complete subproblems of Karp's three hard problems.

Finally in [9] we find context-sensitive language $\{L_i\}$ s.t. $L_i \in$ det CSL iff det CSL = CSL. These languages play a very similar role to the p-complete languages of theorem 2.3 and corollary 2.4.

Note: In [1] Book independently proved the equivalence of (1) and (2) of Theorem 2.3 and Corollary 2.7.

§3: REGULAR SETS AND THE CLASS PTAPE

Definition 3.1: A language $\mathscr{L}$ is tape-hard if PTAPE $\underset{\text{PTIME}}{\leq} \mathscr{L}$.

Definition 3.2: A language $\mathscr{L}$ is tape-complete if $\mathscr{L}$ is tape-hard and $\mathscr{L} \in$ PTAPE.

Definition 3.3: $c\backslash\mathscr{L} = \{x \mid c\ x \in \mathscr{L}\}$. $\mathscr{L}/c = \{x \mid x\ c \in \mathscr{L}\}$.

Definition 3.4: Let $\mathscr{P}$ be a predicate on $2^{\{0,1\}^*}$. $\mathscr{P}_L = \bigcup_{x \in \{0,1\}^*} \{x\backslash\mathscr{L} \mid \mathscr{P}(\mathscr{L})$ is true$\}$.

$\mathscr{P}_R = \bigcup_{x \in \{0,1\}^*} \{\mathscr{L}/x \mid \mathscr{P}(\mathscr{L})$ is true$\}$.

Definition 3.5: (1) A regular expression with intersection (r.e.i.) is defined recursively as follows:

(a) λ, ϕ, $\{0\}$, $\{1\}$ are r.e.i.'s,

(b) If A,B are r.e.i.'s, then (A) $\cup$ (B), (A) $\cap$ (B), (A)*, and (A) (B) are r.e.i.'s, and

(c) A is an r.e.i. if and only if it satisfies (a) or (b).

(2) A regular expression with squaring (r.e.s.) is defined analogously with (b) replaced by

(b') If A,B are r.e.s., then so are (A) $\cup$ (B), (A)* (A)$\cdot$(B), and $(A)^2$

(3) An extended regular expression (e.r.e) is defined similarly with (b) replaced by

(b") If A,B are e.r.e's, then so are (A) $\cup$ (B), (A) $\cap$ (B), (A)*, (A)$\cdot$(B), and $\neg$(A).

Theorem 3.6: Let $\mathscr{P}$ be any predicate on the regular sets over {0,1} s.t.

(a) $\mathscr{P}(\{0,1\}^*)$ is true and

(b) $\mathscr{P}_L$ or $\mathscr{P}_R \subsetneq$ regular sets over {0,1}*.

Then the following hold:

(1). $\mathscr{L} = \{\alpha \mid \alpha$ is a regular expression over {0,1} and $\mathscr{P}(L(\alpha))$ is true$\}$ is tape-hard;

(2) $\mathscr{L} = \{\alpha \mid \alpha$ is a r.e.i. over {0,1} and $\mathscr{P}(L(\alpha))$ is true$\}$ is not polynomial in tape;

(3) $\mathscr{L} = \{\alpha \mid \alpha$ is a r.e.s. over {0,1} and $\mathscr{P}(L(\alpha))$ is true$\}$ requires exponential tape; and

(4) $\mathscr{L} = \{\alpha \mid \alpha$ is an e.r.e. over {0,1} and $\mathscr{P}(L(\alpha))$ is true$\}$ is not elementary recursive.

Proof: A proof of (1), (3), and (4) can be found in [11]. A proof of (2) can be found in [12]. We sketch the proof of 1. We assume the reader is familiar with the notation and results of [15].

Proof of 1: Let M be a deterministic Tm with states S, tape alphabet T, and designated halting state $q_f \in S$. Let $\{\#,\$\} \cup T \cup (S \times T)$, and assume $\{0,1\} \in T$. Let L be a regular set over $\{0,1\}^*$ s.t. $\bar{L} \notin \mathscr{P}_L$. We define a regular expression β_y such that

$$L(\beta_y) = \begin{cases} \Sigma^* \text{ if M does not accept } y = y_1,\ldots,y_n \text{ in space - n} \\ \Sigma^* - Z_y \cdot \$ \cdot L^1, \text{ otherwise.} \end{cases}$$

β_y is characterized as follows:

1) words that do not begin with $\#(q_0,y_1) \ldots y_n \#$, or

2) words that do not contain q_f to the left or a \$ - sign, or

3) words that are not of the form $\#\cdot(\Sigma-\$)\cdot*\#\cdot\$\cdot L$, or

4) words that violate the next-move requirement to the left of the \$-sign.

1) and 2) are the same as in [11] or [15]. (3) follows since $\neg(\#\cdot(\Sigma-\$)^*\cdot\#\cdot\$\cdot L)$ is regular and fixed for all inputs to M. 4) is written as follows:

$$\bigcup_{\sigma_1,\sigma_2,\sigma_3 \in \{\#\} \cup T \cup (S \times T)} [(\Sigma-\$)^*\cdot\sigma_1\cdot\sigma_2\cdot\sigma_3\cdot(\Sigma-\$)^{|y|-1}\cdot((\Sigma-\$)-f_m(\sigma_1,\sigma_2,\sigma_3))\cdot(\Sigma-\$)^*\cdot\$\cdot\{0,1\}^*.$$

[1] Here Z_y is the computation of M on y as defined in [11] or [15].

Inspection of the lengths of the regular expressions needed to describe (1), (2), (3) and (4) shows that β_y is linear in $|y|$. Finally if $L(\beta_y) = \Sigma^*$ then $(L(\beta_y))$ is true. Suppose $L(\beta y) = \Sigma^*\text{-}Z_y \cdot \$ \cdot L$ and suppose $\mathscr{P}(L(\beta_y))$ is true, then $(Z_y \cdot \$ \setminus \Sigma^*\text{-}Z_y \cdot \$ \cdot L) = \bar{L} \in \mathscr{P}_L$ (=><=).

We list several predicates on the regular sets that satisfy the conditions of Theorem 3.7. For underlined predicates, L = {α| α is a regular expression over {0,1} and P(L(α)) is true} can be recognized in polynomial tape.

1. " = {0,1}*" ;
2. " = $\{0,1\}^+$" ;
3. "cofinite" ;
4. "coinfinite";
5. "star event" $(L(\alpha) = L(\alpha)^*)$;
6. "γ-event" $(L(\alpha) = \gamma(L(\alpha))$ def. $\{x|\exists y \in L(\alpha)$ and $|x|=|y|\})$;
7. "reversal event" $(L(\alpha) = L(\alpha)^{rev.})$[1];
8. for all k > 0, "k-definite (k-reverse-definite, or k-generalized definite) event"
9. "definite (reverse-definite, or generalized definite) event";
10. for all k > 0, "strictly k-testable event", "strictly locally testable event", or "locally testable event";
11. "star-free (noncounting, loop-free, or group free, etc) event";
12. "comet (reverse comet, or generalized comet) event";
13. "input-output behavior of a finite state (finite state linear) sequential machine";

and 14. "accepted by some strongly connected deterministic finite state automaton".

For the definitions of 8, 9, 10, and 11 see [4]. The definition of 12 can be found in [3] and the definition of 13 can be found in [7].

Theorem 3.7: Let $\mathscr{P}$ be any predicate on the regular sets over {0,1} s.t.

(a) $\mathscr{P}(\phi)$ is true and

(b) $\mathscr{P}_L$ or $\mathscr{P}_R \subsetneq$ regular sets over {0,1}*.

Then (1) $\mathscr{L}$= {α| α is a 2-way deterministic f.s.a. over {0,1} and $\mathscr{P}(L(\alpha))$ is true} is tape-hard and

(2) $\mathscr{L}$= {α| α is an e.r.e. over {0,1} and $\mathscr{P}(L(\alpha)|$ is true} is not elementary recursive.

Proof: (1) follows since a 2-way deterministic f.s.a. M can check a string to see if it is the "valid computation of Turing machine M_i on input x" in such a way that the construction of M given M_i and x is polynomial in $|x|$.

---

[1] "reversal event" results from the proof of the theorem.

(2) follows since the predicate "$=\phi$" is not elementary recursive for e.r.e.'s. This result was announced at the 1972 IEEE SWAT conference by A. Meyer and L. Stockmeyer.

Perhaps, the reader has noticed that Theorems 3.6 and 3.7 are similar in form and spirit to the theorems appearing in "A Note on Undecidable Properties of Formal Languages" by S. Greibach. In [11] we present strict extensions of Greibach's two theorems. The reader should also note that Theorem 3.6 provides strong intuition as to why Karp [13] was unable to show that the three problems mentioned in Section 2 are p-complete. Such a proof would prove NPTIME = PTAPE. For a more detailed discussion see [11]. Finally the reader should note that the results of Meyer and Stockmeyer in [15] and the present author's results in [11] and this paper suggest why little or no success has been achieved in finding easy "canonical" forms for the regular sets or at least for nontrivial proper subclasses of the regular sets.

We list several other problems which are tape-hard.

Theorem 3.8: The following are tape-hard:

(1) all predicates mentioned after theorem 4.6 when applied to 2-way deterministic or nondeterministic f.s.a.;

*(2) finiteness or infiniteness for 2-way deterministic finite state automata;

(3) equivalence of type 3 grammars; equivalence of (p,q)- linear grammars; and equivalence of skewlinear tuple grammars (for definitions see (6);

(4) structural equicalence for context-free grammars;

(5) equivalence of generalized nondeterministic f.s.a.;

*(6) context-sensitive recognition; and

(7) $\mathscr{L} = \{(P,\alpha) \mid P$ is a deterministic pushdown automaton, α is a regular expression and $L(P) = L(\alpha)$ $(L(P) \subseteq L(\alpha))\}$.

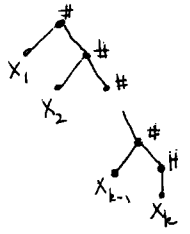
(Note: * denotes a tape complete problem.)

Proof: (1) The 2-way deterministic device could accept all but correct computations.

(2) Finiteness satisfies the conditions of Theorem 4.7.

(3) Obvious.

(4) For type 3 grammars, structural equivalence is equivalent to equivalence. All deviation trees for type 3 grammars of a word $x_1,\ldots,x_k$ look alike if we disregard nonterminals.



(5) Obvious.

(6) For a detailed proof see [11].

(7) Let $L(P) = \{0,1\}^*$.

Note: The proof that "={0,1}*" is "hard" for regular expressions is due to A. Meyer and L. Stockmeyer. The proof that "={0,1}*" is not polynomial in tape for regular expressions with intersection is due to the author.

§4. STACK AUTOMATA

We reduce the membership problem for various kinds of 2-way stack automata to several problems for the corresponding 1-way stack automata. Since the complexity for the 2-way devices is known, this allows us to derive exponential lower time bounds and nonlinear lower tape bounds for the predicates for the 1-way devices.

Theorem 4.1: (Cook [5] and Hopcroft and Ullman [10]):

(1) A language L is accepted by a 2-way SA iff it is accepted by some 2^{cn^2} deterministic time bounded Tm;

(2) A language L is accepted by a 2-way det SA iff it is accepted by some $2^{cn\ell ogn}$ deterministic time bounded Tm;

and (3) A language L is accepted by a 2-way NESA iff it is accepted by some n^2 nondeterministic tape bounded Tm.

Before stating our results we need the following well-known result due to Hennie and Stearns:

Lemma 4.2: If $T_1(n)$ is a real-time countable function, then there is a set A of strings which is accepted by some deterministic off-line Turing machine within time $T_1(n)$, but by no such machine within time $T_2(n)$ for any function T_2 satisfying

$$\lim_{n\to\infty} \inf \frac{T_2(n)\ \ell og[T_2(n)]}{T_1(n)} = 0 .$$

But for all $\epsilon > 0$, $\lim_{n\to\infty} \frac{2^{n^{2-\epsilon}} \cdot n^{2-\epsilon}}{2^{n^2}} = \lim_{n\to\infty} \frac{n^{2-\epsilon}}{2^{n^2(1-\frac{1}{n\epsilon})}} = 0.$

Similarly, $\lim_{n\to\infty} \frac{2^{(n\ell ogn)^{1-\epsilon}} \cdot [n\ell ogn]^{1-\epsilon}}{2^{n\ell ogn}} = \lim_{n\to\infty} \frac{[n\ell ogn]^{1-\epsilon}}{2^{n\ell ogn[1-\frac{1}{[n\ell ogn]\epsilon}]}}$.

Theorem 4.3: Emptiness, finiteness, generalized membership, and equivalence require at least[1]

(1) $2^{n^{2-\epsilon}}$ det. time for 1-way SA;

(2) $2^{[n\ell ogn]^{1-\epsilon}}$ det. time for 1-way det SA; and

(3) $n^{2-\epsilon}$ nondet. tape for 1-way NESA

for all $\epsilon > 0$.

---

[1] For generalized membership, the input is an ordered pair of an automaton and an input. Equivalence for 1-way SA's is known to be undecidable. However, it is unknown whether equivalence is decidable for 1-way det SA's.

Proof: We sketch the proof for emptiness in (1). From Lemma 4.2 we know that there are 2-way SA's whose membership problem requires det time at least $2^{n^{2-\epsilon}}$.

Let S_{i_0} be such a 2-way device. For each input $x = x_1,\ldots,x_n$ to S_{i_0} we build a 1-way device $S_{i_0,x}$ s.t. .

(a) we embed the input tape configurations of S_{i_0} into the finite control of $S_{i_0,x}$,

(b) $S_{i_0,x}$ uses its stack to simulate the stack of S_{i_0},

(c) $S_{i_0,x}$ first simulates S_{i_0} on x if S_{i_0} accepts x then $S_{i_0,x}$ accepts all of its input except ϵ. Otherwise, $S_{i_0,x}$ accepts the empty set, and

(d) $|S_{i_0,x}| \leq$ k nℓogn, where k depends only on S_{i_0} .

The move-rules of S_{i_0} of form $(p,\alpha,\beta,q,\gamma,\delta)$ are replaced by $((p_1 n_1),-,\beta(q,n_2),0,\delta)$, where n_1 and n_2 are binary integers $\leq |x|=n$, which denote the position of the input tape ptr. of S_{i_0} . The number of such move rules in $\leq 0(n)$. The tape needed for each is $\leq 0(\ell ogn)$. Clearly for each move-rule for S_{i_0} we can check which move rules for $S_{i_0,x}$ are allowable. The time is bounded by P(n) for some polynomial. Let T(n) be the amount of time needed to decide emptiness for 1-way SA. Then

$$\ell\text{im inf}_{n\to\infty} \frac{[P(n) + T(n\ell ogn)]\ell og[P(n) + T(n\ell ogn)]}{2^{n^2}} > 0$$

Let $T'(x) = 2^{x^{2-\epsilon'}}$. Then

$$\ell\text{im inf}_{n\to\infty} \frac{2^{(n\ell ogn)^{2-\epsilon'}} \cdot (2-\epsilon') \cdot \ell og(n\ell ogn)}{2^{n^2}} \to 0. \quad \text{Hence}$$

$$\ell\text{im inf}_{n\to\infty} \frac{T'(x)}{T(x)} \to 0. \quad \text{Thus asymptotically } T(x) > 2^{x^{2-\epsilon'}} .$$

We note that such results hold for the nested stack automata and should hold for the indexed context-free grammars. We conjecture such results also hold for context-free grammars on trees. (See [16]).

§5. CONCLUSION

We feel the most important results in this paper are Theorems 3.6 and 3.7. In them we have found a natural "complexity core" that makes recursive problems hard to do. We know of several other such cores and we feel that they exist in many structures.

ACKNOWLEDGEMENTS

I wish to thank my thesis advisor John Hopcroft for his encouragement, time, and help. I also wish to thank Professors W. Ogden and W. C. Rounds of CWRU and Mr. Sartaj Sahni of Cornell University for several useful discussions on the results of this paper.

REFERENCES

[1] Book, R. V., "On Languages Accepted in Polynomial Time", Harvard University Technical Report, 5-72.

[2] Book, R. V. and Greibach, S., "Quasi-realtime Languages", Mathematical Systems Theory 4 (1970), 97-111.

[3] Brzozowski, J. A. and Cohen, R., "On Decomposition of Regular Events", JACM, Vol. 16, No. 1.

[4] Cook, S. A., "The Complexity of Theorem-proving Procedures", Proceedings Third ACM Symposium On Theory of Computing (1971), 151-158.

[5] Cook, S. A., "Characterizations of Pushdown Machines in Terms of Time-Bounded Computers", JACM, Vol. 18, No. 1, 4-18.

[6] Costich, O. L., "An Application of Tree Automata to Linear and Skewlinear Tuple Languages", University of Iowa Technical Report, No. 51, 2-72.

[7] Gray, J. and Harrison, M. A., "The Theory of Sequential Relations", Information and Control 9, 435-468.

[8] Greibach, S., "A Note on Undecidable Properties of Formal Languages", Mathematical Systems Theory 2:1 (1968), 1-6.

[9] Hartmanis, J. and Hunt, H. B. III, "The LBA Problem and its Importance in the Theory of Computing", to be presented at the AMS Conference on Computational Complexity, April 1973.

[10] Hopcroft, J. E. and Ullman, J. D., Formal Languages and Their Relation to Automata , Addison-Wesley Publishing Co., Reading, Mass., 1969.

[11] Hunt, H. B. III, "On the Time and Tape Complexity of Languages, I", Cornell University, Department of Computer Science, Technical Report No. 73-156.

[12] Hunt, H. B. III, "The Equivalence Problem For Regular Expressions with Intersection is not Polynomial in Tape" (to appear as a Cornell University Department of Computer Science Technical Report.)

[13] Karp, R. M., "Reducibility Among Combinatorial Problems", Technical Report No. 3, April 1972, Department of Computer Science, University of California, Berkeley.

[14] McNaughton, R. and Papert, S., "Context-Free Automata", MIT Press, Cambridge, Massachusetts, 1971.

[15] Meyer, A. R. and Stockmeyer, L. J., "The Equivalence Problem for Regular Expressions With Squaring Requires Exponential Space", 13th Annual Symposium on the Theory of Computing (1970), 125-129.

[16] Rounds, W. C., "Tree-Oriental Proofs of Some Theorems on Context-Free and Indexed Languages", Proceedings Second ACM Symposium on the Theory of Computing (1970), 109-116.

[17] Savitch, W., "Relationship Between Nondeterministic and Deterministic Tape Complexities", J. of Comput. and System Sci. 4 (1970), 177-192.