相対化された指数時間計算量クラスについて

和泉, 正明 / IZUMI, Masa-aki

(出版者 / Publisher)
法政大学工学部
(雑誌名 / Journal or Publication Title)
法政大学工学部研究集報 / 法政大学工学部研究集報
(巻 / Volume)
20
(開始ページ / Start Page)
49
(終了ページ / End Page)
59
(発行年 / Year)
1984-03
(URL)
https://doi.org/10.15002/00004078

ON SOME RELATIVIZED COMPLEXITY CLASSES DEFINED BY EXPONENTIAL-TIME-BOUNDED ORACLE TURING MACHINES

Masa-aki IZUMI

Abstract

Let for all $k \geq 1$ $\mathcal{NE}_k$ denote the class of languages acceptable by 2^{lin}-time-bounded Turing machines making at most n^k nondeterministic moves on inputs of length n for each n. For any $k \geq 1$, DEXT$\subseteq \mathcal{NE}_k \subseteq$NEXT. We shall construct oracles A, B and D_k ($k \geq 2$) such that

(1) $\mathrm{DEXT}^A = \mathcal{NE}_1^A = \mathcal{NE}_2^A = \ldots = \mathcal{NE}_k^A = \ldots = \mathcal{NE}^A = \mathrm{NEXT}^A$;

(2) $\mathrm{DEXT}^B = \mathcal{NE}_1^B \subsetneq \mathcal{NE}_2^B \subsetneq \ldots \subsetneq \mathcal{NE}_k^B \subsetneq \ldots \subsetneq \mathcal{NE}^B$;

(3) $\mathrm{DEXT}^{D_k} = \mathcal{NE}_1^{D_k} \subsetneq \mathcal{NE}_2^{D_k} \subsetneq \ldots \subsetneq \mathcal{NE}_k^{D_k} = \mathcal{NE}_{k+1}^{D_k} = \ldots \mathcal{NE}^{D_k} = \mathrm{NEXT}^{D_k}$.

Results regarding closure of the classes $\mathcal{NE}_k$ under complementation are also given. Furthermore, we investigate relativized versions of the open question of whether NEPT equals co-NEPT, where co-NEPT$=\{L \mid \bar{L} \in$NEPT$\}$.

1. Introduction

Let DEXT (resp. NEXT) be the class of languages accepted by deterministic (resp. nondeterministic) 2^{lin}-time-bounded Turing machines, where 2^{lin} means $\lambda n[2^{cn}]$ for some c. Clearly, DEXT$\subseteq$NEXT. We do not know whether NEXT properly contains DEXT. Now, let for all $k \geq 1$ $\mathcal{NE}_k$ denote the class of languages acceptable by 2^{lin}-time-bounded Turing machines making at most n^k nondeterministic moves on inputs of length n for each n. Then the following hierarchy holds:

$$\mathrm{DEXT} = \mathcal{NE}_1 \subseteq \mathcal{NE}_2 \subseteq \ldots \subseteq \mathcal{NE}_k \subseteq \ldots \subseteq \mathcal{NE} \subseteq \mathrm{NEXT}$$

Refining the original question DEXT$=$? NEXT, we can ask whether $\mathcal{NE}_k =$? $\mathcal{NE}_{k+1}$ for any $k \geq 1$. The latter question for any k is open, too. To support this we can relativize the question analogously to the work of Baker, Gill and Solovay[1], and show that for every $k \geq 1$, the corresponding relativized questions have affermative answers for some oracles but negative answers for other oracles. Thus, for every $k \geq 1$, we feel that this is further evidence of the difficulty of the $\mathcal{NE}_k =$? $\mathcal{NE}_{k+1}$ question. We also construct oracles for each k such that the relativized version of the above hierarchy is distinct up to the k-th level but collapses from the k-th level onwards. The closure property of these relativized classes under complementation is also investigated.

Let NEPT be the class of languages accepted by nondeterministic 2^p-time-bounded

Department of Instrument and Control Engineering
College of Engineering, Hosei University

Turing machines, where p is a polynomial. It is not known whether NEPT equals co-NEPT, where co-NEPT$=\{L|\overline{L}\in$NEPT$\}$. We investigate relativized versions of this question. In particular, we shall construct oracles G and H such thai (i) DEPT$^G\neq$ NEPT^G but NEPT^G is closed under complementation; (ii) DEPT$^H\subsetneq$NEPT$^H\cap$co-NEPT^H $\subsetneq$NEPT^H.

We use in the present paper some methods developed in Baker-Gill-Solovay[1], Book[3] and Kintala-Fischer[5].

2. Preliminaries

Let ω be the set of all natural numbers and let Σ be an alphabet. We assume $\Sigma=\{0, 1\}$. Σ^* denotes the set of all finite strings consisting of 0's and 1's. A subset L of Σ^* is called language and we denote the complement of L by $\overline{L}$: $\overline{L}=\Sigma^*-L$. For $x\in\Sigma^*$, $|x|$ denotes the length of x. Our model for computation is the oracle Turing machine. An *oracle Turing machine* (abbreviated by OTM) is a multi-tape Turing machine with a *query tape* and with three special internal states called the *query state* $q_?$, the *yes state* q_Y and *no state* q_N. When an OTM M is associated with an *oracle* $X\subseteq\Sigma^*$, we denote it by M^X and call an OTM with oracle X. When an OTM M^X enters the state $q_?$, the machine asks the oracle X whether the string written on its query tape belongs to X. If the string is in X, then M^X enters q_Y, otherwise M^X enters q_N.

If the next-state-operation of M is single-valued is said to be *deterministic*, otherwise M to be *nondeterministic*. Now suppose that M^X runs on an input x and M^X halts after some running time. If the final state of M is a special state called an accepting state we say M^X *accepts* x. Otherwise we say M^X *rejects* x. Let L be a language. L is *accepted* (or *recognized*) by an OTM M with oracle X (denoted by $L=R(M^X)$) if the following condition holds:

$$\forall x\in\Sigma^*[x\in L \text{ iff } M^X \text{ accepts } x].$$

Let f be a function from ω into ω (f: $\omega\to\omega$). An OTM M is *f-time-bounded* if every computation of M on any input x halts within $f(|x|)$ steps, whatever oracle X is used. DTIME$^X(f)$ (resp. NTIME$^X(f)$) is the class of languages accepted by deterministic (resp. nondeterministic) f-time-bounded OTM's with oracle X. We consider the following classes of languages:

$$\mathcal{P}^X=\bigcup_{i=1}^{\infty}\text{DTIME}^X(\lambda n[n^i]),$$

$$\text{DEXT}^X=\bigcup_{c=1}^{\infty}\{\text{DTIME}^X(\lambda n[2^{cn}])|c>0\} \text{ and}$$

$$\text{DEPT}^X=\bigcup\{\text{DTIME}^X(\lambda n[2^{p(n)}])|p \text{ is a polynomial}\}.$$

$\mathcal{NP}^X$, NEXT^X and NEPT^X are nondeterministic counterparts of $\mathcal{P}^X$, DEXT^X and DEPT^X, respectively.

A c-ary nondeterministic move of a machine is a move in which the number of choices for the next step of the machine is c. Such a nondeterministic move is sometimes said to have a *fan-out* of c. By a nondeterministic move, we mean a "strict" nondeterministic move where there are at least two choices for the next step of the machine. Any OTM M can be so designated that all the nondeterministic moves made by M have the same fan-out c for some constant c, which depends on M.

Definition 2. 1. For any $k\geq 1$ and for any oracle X, let $\mathcal{NE}_k^X=\{L|L\subseteq\{0,1\}^*$ and there is a constant c such that L is accepted by a 2^{lin}-time-bounded OTM with oracle X making at most n^k c-ary nondeterministic moves on inputs of length n for each $n\}$. We also define $\mathcal{NE}^X=\bigcup_{k=1}^{\infty}\mathcal{NE}_k^X$.

For any oracle X,

$$\text{DEXT}^X=\mathcal{NE}_1^X\subseteq\mathcal{NE}_2^X\subseteq\ldots\subseteq\mathcal{NE}_k^X\subseteq\ldots\subseteq\mathcal{NE}^X\subseteq\text{NEXT}^X.$$

Let DE_i (resp. NE_i) be the i-th deterministic (resp. nondeterministic) exponential-time-bounded OTM with its strict time-bound g_i, where $g_i(n)=2^{a_in}$, $0<a_0<a_1<\ldots$ For each i, we can effectively determine a c_i such that c_i is the fan-out of the nondeterministic moves made by NE_i. For any $k\geq 1$, let $NE_{i,k}$ denote the OTM which results by attaching a $\lambda n[n^k]$-time clock to NE_i. This clock stops NE_i if $\lambda n[n^k]$ nondeterministic moves are exceeded. Let DEP_i (resp. NEP_i) be the i-th deterministic (resp. nondeterministic) exponential-polynomial-time-bounded OTM with its strict time-bound $h_i(n)=2^{p_i(n)}$, where we may assumed $p_i(n)=C_in^{C_i}$, $0<C_0<C_1<\ldots$

For every $k\geq 1$ and all oracles X, a language L_0 is *$\mathcal{NE}_k^X$-complete* if (i) $L_0\in\mathcal{NE}_k^X$ and (ii) for $L\in\mathcal{NE}_k^X$ there is function $\varphi:\Sigma^*\to\Sigma^*$ such that φ is computable by a deterministic exponential-time-bounded transducer [then we say "φ is exponential-time-*computable*"] and such that for every x $x\in L$ iff $\varphi(x)\in L_0$. We denote (ii) by $L\leq_m^E L_0$.

3. $\mathcal{NE}_k=?\ \mathcal{NE}_{k+1}$ question

It is known that there exists an oracle set A such that $\text{DEXT}^A=\text{NEXT}^A$. It is obvious that for any such A, $\mathcal{NE}_k^A=\mathcal{NE}_{k+1}^A$ for every $k\geq 1$.

Definition 3. 1. For every $k\geq 1$ and alll oracles X, let

$$L_k(X)=\{x|\exists y[|y|=|w|^k \text{ and } w1^{2|w|}y\in X]\}.$$

Clearly, $L_k(X)\in\mathcal{NE}_k^X$. Note that $|w1^{2|w|}y|$ is always even.

Theorem 3. 2. There is an oracle B such that $\mathcal{NE}_{k-1}^B\subsetneq\mathcal{NE}_k^B$ for every $k\geq 2$, i.e.,

$$\text{DEXT}^B=\mathcal{NE}_1^B\subsetneq\mathcal{NE}_2^B\subsetneq\ldots\subsetneq\mathcal{NE}_k^B\subsetneq\ldots\subsetneq\mathcal{NE}^B.$$

Proof. Let $e=\langle i,k\rangle$ be a recursive bijection from $\omega\times(\omega-\{0\})$ into ω ($e=\langle i,k\rangle$: $\omega\times(\omega-\{0\})\xrightarrow[\text{onto}]{1-1}\omega$). Let $B(s)$ be the set of all strings put in B before stage s and let $B(0)=\phi$. During construction of B, each index $e=\langle i,k\rangle$ is cancelled at some stage n_e when we ensure that $NE_{i,k-1}^B$ does not accept $L_k(B)$. Each index will be eventually cancelled. Let $n_{-1}=0$ and start at stage 0.

Stage s. Let $e=\langle i, k\rangle$ be the first uncancelled index. If any of the following two conditions is not satisfied, then skip this stage and go to next stage. Otherwise, cancel e at this stage and choose $n_e=s$ as follows:
suppose $e-1=\langle j, m\rangle$ for some $j\geq 0$ and $m\geq 1$.

(i) $2^s > g_j(n_{e-1})$;

(ii) $c_i^{s^{k-1}}.g_i(s) < 2^{s^k}$.

Run the machine $M=NE_{i,k-1}^{B(s)}$ on $z=0^s$. If M rejects z, then add to B some string of the form $z1^{2^{|z|}}v$ such that $|v|=|z|^k$ and $z1^{2^{|z|}}v$ is not queried during any possible computation of M on z. Such a string exists because of condition (i) and (ii). If M accepts z let $B(s+1)=B(s)$.

Let $B=\bigcup_{s=0}^{\infty} B(s)$.

By construction, $NE_{i,k-1}^{B(s)}$ rejects $z=0^s$ iff $NE_{i,k-1}^{B}$ rejects $z=0^s$ iff $\exists v[|v|=|z|^k$ and $z1^{2^{|z|}}v\in B]$ iff $z\in L_k(B)$. So, $NE_{i,k-1}^{B}$ does not accept $L_k(B)$, and hence $L_k(B)\notin \mathcal{NE}_{k-1}^{B}$. □

Theorem 3. 3. For every $k\geq 2$, there is an oracle D such that

$$\mathrm{DEXT}^D=\mathcal{NE}_1^D \subsetneq \mathcal{NE}_2^D \subsetneq \dots \subsetneq \mathcal{NE}_k^D = \mathcal{NE}_{k+1}^D = \dots = \mathcal{NE}^D = \mathrm{NEXT}^D.$$

Proof. We will say that a string $y=0^d1x10^t$ is *admissible* if $|x|\geq d$ and $|y|=a_d|x|\geq |x|^{(k+1)/k}$. Let $D(s)$ be the set of all strings put in D before stage s. During construction of D, some strings will be reserved for $\overline{D}$. An index $e=\langle i, m\rangle$ (for some canonical enumeration of the pairs of the form $\langle i, m\rangle$ such that $i\geq 0$ and $2\leq m\leq k$) will be cancelled at some stage n_e when we ensure that $NE_{i,m-1}^{D}$ does not accept $L_m(D)$. Set $D(0)=\phi$, $n_{-1}=0$, and start at stage 0.

Stage s: Execute the following two routines.

Routine A. For every string y of length s, if $y=0^d1x10^t$ is admissible, run $NE_d^{D(s)}$ on input x. In any such computation only strings of length $<2^s$ are queried.

For each such y and associated d and x, if any computation of $NE_d^{D(s)}$ accepts x, then place some string of the form $y1^{2^{|y|}}w$ into D where

(i) $w\in\{0, 1\}^*$; $|w|=|x|^k+\varepsilon$ where $\varepsilon=0$ or 1 so that $NE_d^{D(s)}$ accepts x,

(ii) $y1^{2^{|y|}}w$ has not been reserved for $\overline{D}$ in an earlier stage.

(We will ensure in condition (iv) of the following routine, which is the only routine reserving for $\overline{D}$, that such w is available, if needed.)

Routine B. Let $e=\langle i, m\rangle$ be the first uncancelled index. If any of the following four conditions is not satisfied, then skip this routine and go to stage $s+1$. Otherwise, cancel e at this routine and choose $n_e=s$ as follows:
suppose $e-1=\langle j, h\rangle$ for some $j\geq 0$ and $2\leq h\leq k$.

(i) $2^s > g_j(n_{e-1})$;

(ii) no string of length $\geq 2^s$ is reserved for $\overline{D}$;

(iii) $c_i^{s^{m-1}} \cdot g_i(s) < 2s^m$;

(iv) $\forall \varepsilon, y, d, x, t[\varepsilon=0$ or $y=0^d 1 x 10^t$ and $|y| \geq s$ and y is admissible $\longrightarrow$ $\exists w\{|w|=|x|^k+\varepsilon$ and $(NE_{i,m-1}^{D(s)}$ does not query $y1^{2|y|}w$ in the computation on $0^s)\}]$.

Run the machine $M=NE_{i,m-1}^{D'}$, where $D'=D(s) \cup$ {the odd length strings you just added by routine A in this stage}, on input $z=0^s$. We reserve for $\overline{D}$ all strings of length $\geq 2^s$ queried during any possible computation of M on z. If M rejects z, then add to D some string of the form $z1^{2|z|}v$ such that $|v|=s^m$ and $z1^{2|z|}v$ is not queried during any possible computation of M on z. If M accepts z let $D(s+1)=D'$.

Let $D=\bigcup_{s=0}^{\infty} D(s)$.

Claim 1. At routine B, condition (iv) holds.

[Proof. For large enough s and for any admissible $y=0^d 1 x 10^t$ such that

$$s \leq |y| = a_d|x| \leq |x|^{(k+1)/k}$$

$NE_{i,m-1}^{D(s)}$ will be reserved at most

$$c_i^{s^{m-1}} \cdot g_i(s) = 2^{(\log c_i)s^{m-1}+a_i s}$$

$$2^{(\log c_i)|x|^{(k+1)(m-1)/k}+a_i|x|^{(k+1)/k}}$$

$$2^{|x|^k} \text{ for large enough } x$$

since $(k+1)(m-1)/k<k$ if $2 \leq m \leq k$ and $k \geq 2$.]

Claim 2. Each index e can eventually be cancelled.

[Proof is clear.]

Claim 3. When routine A of stage s is executed, such a string $y1^{2|y|}w$ exists and $y1^{2|y|}w$ is not queried at any earlier stage.

[Proof. Let s' be the last stage that routine B was executed before stage s, and let $e'=\langle i', m'\rangle$ be the first uncancelled index at stage s'. So, by Claim 1,

(1) $\forall \varepsilon, y', d', x', t'[\varepsilon=0$ or 1 and $y'=0^{d'}1x'10^{t'}$ and $|y'| \geq s'$ and y' is admissible $\longrightarrow \exists w\{|w|=|x'|^k+\varepsilon$ and $(NE_{i',m'-1}^{D(s')}$ does not query $y'^{2|y'|}w$ in the computation on $0^{s'})\}]$.

Let $y=0^d 1 x 10^t$ be any admissible string taken at stage s. Since $s'<s$, by (1) there is a string w such that $|w|=|x|^k+\varepsilon$, $y1^{2|y|}w$ is even and $NE_{i',m'-1}^{D(s')}$ does not query $y^{2|y|}w$ in the computation on $0^{s'}$. Since only strings of length $<2^{s'}$ are queried at routine B of stage $<s'$ (because of condition (i) of routine B), $y1^{2|y|}w$ is not in $\overline{D}$ at routine B of earlier stage $<s$. Moreover, only strings of length $<2^s$ are queried at routine A of stages $\leq s$. So, $y1^{2|y|}w$ is not queried at any earlier stage.]

Claim 4. When routine B of stage s is executed, such a string $z1^{2|z|}v$ exists and $z1^{2|z|}v$ is not queried at any earlier stage $<s$ and routine A of stage s.

[Proof. The number of strings queried in all the computation of $NE_{i,m-1}^{D'}$ on $z=0^s$ is less than 2^{s^m} (by (iii)). So there is a string $z1^{2^{|z|}}v$ of length 2^s+s^m+s that is not queried in the above computation. This $z1^{2^{|z|}}v$ is not queried at any earlier stage $<s$ and routine A of stage s. For, $z1^{2^{|z|}}v$ is not in $\overline{D}$ so far because of condition (ii) of routine B. So, $z1^{2^{|z|}}v$ is not queried at routine B of any stage $<s$. Since $|z1^{2^{|z|}}v|>2^s$, $z1^{2^{|z|}}v$ is not queried at routine A of any stage $\leq s$.]

Claim 5. $\mathcal{NE}^D_{m-1} \subsetneq \mathcal{NE}^D_m$ for every m such that $2\leq m\leq k$.

[Proof. Let m $(2\leq m\leq k)$ be arbitrary. Then take i (≥ 0) arbitrary. Now, let $e=\langle i, m\rangle$. By Claim 2, n_e is determined. Let $n_e=s$. By Claim 4, $NE^D_{i,m-1}$ rejects $z=0^s$ iff $NE^{D'}_{i,m-1}$ rejects $z=0^s$ iff $\exists v[|v|=|z|^m$ and $z1^{2^{|z|}}v\in D]$ iff $z\in L_m(D)$. So, $NE^D_{i,m-1}$ does not accept $L_m(D)$, and hence $L_m(D)\notin\mathcal{NE}^D_{m-1}$.]

Claim 6. $\mathcal{NE}^D_k=\mathrm{NEXT}^D$.

[Proof. Let $L\in\mathrm{NEXT}^D$ be arbitrary. So, there is an index d such that $L=R(NE^D_d)$. We define a nondeterministic $2^{l|n}$-time-bounded OTM M with oracle D making at most $|x|^k$ nondeterministic moves as follows:

Given an input x, first M constructs a string y such that

$$y=0^d\,1x\,10^t \text{ and } |y|=a_d|x|\leq|x|^{(k+1)/k} \text{ and } |x|\geq d.$$

Since there are only finitely many x's for which there is no such y, we make a finite table so that M accepts x iff $x\in L$ for such x's. Then M guesses a string w such that $|w|=|x|^k+\varepsilon$ ($\varepsilon=0$ or 1) and $y^{2^{|y|}}w$ is odd. M accepts x if $y^{2^{|y|}}w\in D$. Otherwise M rejects x. Let $|y|=s$. By Claim 3, M accepts x iff $y1^{2^{|y|}}w\in D$ iff $NE^{D(s)}_d$ accepts x iff NE^D_d accepts x iff $x\in L$. Guessing such a string w can be executed in $|x|^k$ nondeterministic moves, M can decide whether it accepts x within $2^{c|x|}$ steps for some c. Thus $L=R(M)$. Consequently, $L\in\mathcal{NE}^D_k$ and hence $\mathrm{NEXT}^D\subseteq\mathcal{NE}^D_k$.]

This completes the proof of Theorem 3. 3. □

4. Closure property under complementation

We do not know whether $\mathcal{NE}_k$ is closed under complementation for any $k\geq 2$. However in the relativized case we can exhibit oracle sets for each side of question.

Theorem 4. 1. For each $k\geq 2$, there is an oracle E such that $\mathcal{NE}^E_k$ is not closed under complementation.

Proof. We construct an E in stages. We denote by $E(s)$ the finite set of strings placed into E prior to stage s. Let $E(s)=\phi$, $n_{-1}=0$ and start at stage 0.

Stage s. Choose a sufficiently large n_s so that

(i) $n_s > n_{s-1}$;

(ii) $2^{n_s} > g_{s-1}(n_{s-1})$;

(iii) $g_s(n_s) < 2^{n_s^k}$.

Run the machine $M=NE_{s,k}^{E(s)}$ on $x=0^{n_s}$. If M accepts x, then choose any accepting computation and place into E some string of the form $x1^{2^{|x|}}y$ such that $|y|=|x|^k$ and $x1^{2^{|x|}}y$ is not queried in the computation. Such a string $x1^{2^{|x|}}y$ exists and $x1^{2^{|x|}}y$ is not queried at any earlier stage (by (i), (ii) and (iii)). If M rejects x let $E(s+1)=E(s)$. If at least one of conditions (i)−(iii) does not hold, then let $E(s+1)=E(s)$.

Let $E=\bigcup_{s=0}^{\infty} E(s)$.

By construction, $NE_{s,k}^{E}$ accepts x iff $NE_{s,k}^{E(s)}$ accepts x iff $\exists y[|y|=|x|^k$ and $x1^{2^{|x|}}y\in E]$ iff $x\in L_k(E)$ iff $x\notin\overline{L_k(E)}$. Consequently $NE_{s,k}^{E}$ does not accept $\overline{L_k(E)}$. Therefore $\overline{L_k(E)}\notin\mathcal{NE}_k^E$. □

Lemma 4. 2 (Constant speed-up). For any time-constructible $g(n)$ and any oracle X, $\mathcal{NE}_{g(n)}^X=\mathcal{NE}_{g(n)/2}^X$, where $\mathcal{NE}_{g(n)}$ denote the class of languages acceptable by 2^{lin}-time-bounded Turing machines making at most $g(n)$ nondeterministic moves on inputs of length n for each n.

Proof. See Kintala-Fischer[5]. □

Lemma 4. 3. For every $k\geq 1$, $\mathcal{NE}_k=\mathcal{NE}_{k+1}$ implies $\mathcal{NE}_{k+1}=\mathcal{NE}_{k+2}$.

Proof. $k(\geq 1)$ is fixed. Clearly, $\mathcal{NE}_{k+1}\subseteq\mathcal{NE}_{k+2}$.
For any $L_1\in\mathcal{NE}_{k+2}$, there is an index i such that $L_1=R(NE_{i,k+2})$. Let $M_1=NE_{i,k+2}$, and let $L_2=\{w10^m|w\in L_1$ and $|w10^m|=|w|\lceil|w|^{1/(k+1)}\rceil\}$. From M_1 one can construct a nondeterministic 2^{lin}-time-bounded Turing machine M_2 making at most n^{k+1} nondeterministic moves on inputs of length n for each n. Thus, $L_2=R(M_2)\in\mathcal{NE}_{k+1}$. Now, if $\mathcal{NE}_k=\mathcal{NE}_{k+1}$, then there is an index j such that $L_2=R(NE_{j,k})$. Let $M_3=NE_{j,k}$. But from M_3 one can construct a nondeterministic Turing machine M_4 which accepts L_1 and on input w uses the same number of nondeterministic moves as M_3 uses on input $w10^m$ where $|w10^m|=|w|\lceil|w^{1/(k+1)}\rceil$, on input w, M_4 uses $|w10^m|^k(\leq|w|^k(|w|^{1/(k+1)}+1)^k<Cw^{k+1}$ for some C) nondeterministic moves within $2^{c|w|}$ steps for some c. Thus, by Lemma 4. 2, $L_1=R(M_4)\in\mathcal{NE}_{k+1}$. But L_1 was chosen as an arbitrary language in $\mathcal{NE}_{k+2}$. Hence, $\mathcal{NE}_{k+2}=\mathcal{NE}_{k+1}$. □

Theorem 4. 4. For every $k\geq 2$, there is an oracle F such that

(1) $\mathcal{NE}_k^F$ is closed under complementation and

(2) $\text{DEXT}^F\neq\mathcal{NE}_k^F$.

Proof. Take the oracle F such that $\mathcal{P}^F\neq\mathcal{NP}^F$ but $\mathcal{NP}^F$ is closed under complementation. (See Theorem 5 in Baker-Gill-Solovay[1].)
We show that $\mathcal{NE}_k^F$ is closed under complementation. Consider

$A_k(F)=\{0^i1x10^n|$ some computation of $NE_{i,k}^F$ accepts x in no more than n steps$\}$.

Then $A_k(F)\in\mathcal{NP}^F$ and hence $\overline{A_k(F)}\in\mathcal{NP}^F$.
So, let $A_k(F)=R(NP_i^F)$ for some i, where

NP_i

is the i-th nondeterministic polynomial-time-bounded OTM with strict time-bound p_i. Here $p_i(n)=C_i n^{C_i}, C_0<C_1<\ldots$ It is obvious that above NP_i making at most $|x|^k$ nondeterministic moves on inputs of the form $0^i 1x10^n$. Let $L\in\mathcal{NE}_k^F$ be arbitrary. Since $A_k(F)$ is $\mathcal{NE}_k^F$ -complete, there is an exponential-time-computable function $\varphi:\Sigma^*\to\Sigma^*$ such that $x\in\overline{L}$ iff $\varphi(x)\in\overline{A_k(F)}$. Define an OTM M with oracle D as follows: Given x M constructs $\varphi(x)$. This is done in exponetial time. Ask if $\varphi(x)$ is in $\overline{A_k(F)}$. Then M simulates NP_i^F on the string $\varphi(x)$ and if NP_i^F accepts $\varphi(x)$ M accepts x. This is (nondeterministically) done within $p_i(|\varphi(x)|)$ steps $\leq 2^{a|x|}$ steps for some constant a. And NP_i^F making at most $|x|^k$ nondeterministic moves on inputs of length $\varphi(x)$. So, the entire computation of M on x is done in exponential time with at most $|x|^k$ nondeterministic moves. Hence $L\in\mathcal{NE}_k^F$.

Now we claim $\mathrm{DEXT}^F\neq\mathcal{NE}_k^F$. For, suppose $\mathrm{DEXT}^F=\mathrm{NEXT}^F$. Then by the relativized from of a theorem of Book's[2], Tally $(\mathcal{P}^F)=$Tally $(\mathcal{NP}^F)$, where Tally $(\mathcal{C})$ is the subclass of $\mathcal{C}$ consisting of only tally languages in $\mathcal{C}$. In their proof of Theorem 5 in Baker-Gill-Solovay[1], we can replace their $L(F)=\{x|\exists y\in F[|y|=|x|[\}$ by Hopcroft-Ullman's $L'(F)=\{0^n|\exists y\in F[|y|=n]\}$ (see [4; p. 363]) so that $L'(F)\in\mathcal{NP}^F-\mathcal{P}^F$. This implies Tally$(\mathcal{P}^F)\neq$Tally $(\mathcal{NP}^F)$, a contradiction. Consequently, we must have DEXT^F $=\mathrm{NEXT}^F$. Therefore, by Lemma 4. 2. $\mathrm{DEXT}^F\neq\mathcal{NE}_k^F$ for every $k\geq 2$. □

Corollary 4. 5, For esch $k\geq 2$, there is an orscle F such that

(1) $\mathcal{NP}^F$ is closed under complementation and

(2) $\mathrm{DEXT}^F\neq\mathcal{NE}_k^F$.

5. NEPT= ? co-NEPT question

In this last section, we investigate the NEPT= ? co-NEPT question. Now, we have two theorems about relativized versions of this question.

Definition 5. 1. For any oracle X, let

$L_{exp}(X)=\{0^n|\exists y[|y|=2^n \text{ and } y\in X]\}$.

Clearly, $L_{exp}(X)\in\mathrm{NEXT}^X$. So, $L_{exp}(X)\in\mathrm{NEPT}^X$.

Definition 5. 2. For any oracle X, let

$K(X)=\{0^i 1x10^n|$ some computation of NEP_i^X accepts x within 2^n steps$\}$.

Clearly, $K(X)\in\mathrm{NEXT}^X$. So, $K(X)\in\mathrm{NEPT}^X$.

Theorem 5. 3. There is an oracle G such that

(1) NEPT^G is closed under complementation and

(2) $\mathrm{DEPT}^G\neq\mathrm{NEPT}^G$.

Proof. It is easily that NEPT^G is closed under comlementation if and only if $\overline{K(G)}\in\mathrm{NEPT}^G$. We shall construct an oracle G such that (i) $L_{exp}(G)\in\mathrm{NEPT}^G-\mathrm{DEPT}^G$

and (ii) $u \in \overline{K(G)}$ iff u is a prefix of some string v in G such that $|v|=|u|^2$. Then $\mathrm{DEPT}^G \neq \mathrm{NEPT}^G$ from (i); and $\overline{K(G)} \in \mathrm{NEPT}^G$ and so NEPT^G is closed under complementation.

At stage s in the construction, we decide the membership in G of all strings of length 2^s. In the course of the construction, some strings will be reserved for $\overline{G}$, that is, designated as nonmembers of G. An index e will be cancelled at some stage when we ensure that DEP_e^G does not accept $L_{\mathrm{epx}}(G)$. $G(s)$ denotes those strings placed into G prior to stage s. Let $G(0)=\phi$ and start at stage 0.

Stage $s=2i$. For every string z of length 2^s not reserved for $\overline{G}$ at an earlier stage, determine the prefix u of z of length 2^i. If $u=0^j 1 x 1 0^l$, then place z into G iff $NEP_j^{G(s)}$ does not accept x in fewer than 2^l steps.

Stage $s=2i+1$. Let e be the first uncancelled index. If any string of length $\geq 2^s$ has been reserved for $\overline{G}$, or $h_e(s) \geq 2^{2^i}$, then add no elements to G at this stage. Otherwise, run $DEP_e^{G(s)}$ on input 0^s and reserve for $\overline{G}$ all strings of length $\geq 2^s$ queried during this computation. If $DEP_e^{G(s)}$ rejects 0^s, then add to G the least string of length 2^s not queried. Finally cancel index e.

Every index is eventuarry cancelled, and when index e is cancelled at some stage, we have guaranteed that DEP_e^G does not accept $L_{\mathrm{exp}}(G)$. Therefore $L_{\mathrm{exp}}(G) \in \mathrm{NEPT}^G - \mathrm{DEPT}^G$.

At any odd stage $2i+1$, at most $h_e(s) < 2^{2^i}$ strings are reserved for $\overline{G}$, and so fewer $2^{2^0}+2^{2^1}+\ldots+2^{2^{i-1}} < 2^{2^i}$ strings of length 2^{2i} can be reserved for $\overline{G}$ at odd stages before stage $2i$. Therefore every string u of length 2^i is the prefix of at least one string v of length 2^{2i} which is never reserved for $\overline{G}$. By construction, $u \in \overline{K(G)}$ iff u is the prefix of a string of length $|u|^2$ in G and so $\overline{K(G)} \in \mathrm{NEPT}^G$. □

Theorem 5. 4. There is an oracle H such that $\mathrm{DEPT}^H \subsetneq \mathrm{NEPT}^H \cap \text{co-}\mathrm{NEPT}^H \subsetneq \mathrm{NEPT}^H$.

Proof. We may assume that for the $\{C_i | i \in \omega\}$.

(1) $C_0 > 0$ and $2^{(2C_{i-1}-1)^2} > h_i(2C_i)$ for all i

holds. Let for any X

$$L_{\mathrm{even}}(X) = \{x \mid |x| \text{ is even and } \exists y[y \in X \text{ and } |y| = 2^{|x|^2}]\}.$$

$$L_{\mathrm{odd}}(X) = \{x \mid |x| \text{ is odd and } \exists y[y \in X \text{ and } |0y| = 2^{|x|^2}]\}.$$

We shall construct H such that

(2) $L_{\mathrm{even}}(H) \in \mathrm{NEPT}^H - \text{co-}\mathrm{NEPT}^H$,

(3) $L_{\mathrm{odd}}(H) \in \mathrm{NEPT}^H \cap \text{co-}\mathrm{NEPT}^H - \mathrm{DEPT}^H$,

and such that

(4) $\forall n[n \text{ is odd} \longrightarrow \exists y(0y \in H \text{ and } |0y| = 2^{n^2}) \text{ iff } \neg\exists y(1y \in H \text{ and } |1y| = 2^{n^2})]$

holds. (4) implies $\overline{L_{\mathrm{odd}}(H)} \in \mathrm{NEPT}^H$, and hence $L_{\mathrm{odd}}(H) \in \mathrm{NEPT}^H \cap \text{co-}\mathrm{NEPT}^H$. Let $H(s)$ be the set of all strings put in H before stage s and let $H(0)=\phi$.

Stage $s=3i$. We add all strings $1^{\alpha}(\alpha=2^{n^2})$ with n satisfying the following condition (5) to $H(s)$ to make $H(s+1)$:

(5) n is odd, $2^{n^2} < h_i(2C_i)$, $\forall z(|z|=2^{n^2} \longrightarrow z \notin H(s))$ and $n \neq 2C_i-1$.

Stage $s=3i+1$. Let $z_i=0^{2C_i-1}$ and run $DEP_i^{H(s)}$ on z_i. If it rejects z_i we take a string of the form $0y$ of length $2^{(2C_i-1)^2}$ not queried in the comsutation, and let $H(s+1)=H(s)\cup\{0y\}$. Clearly such a string $0y$ exists. If $DEP_i^{H(s)}$ accepts z_i we take a string of the form $1y$ of length $2^{(2C_i-1)^2}$ not queried in the computation, and let $H(s+1)=H(s)\cup\{1y\}$. Such a $1y$ exists, too.

Stage $s=3i+2$. Let $x_i=0^{2C_i}$ and run $NEP_i^{H(s)}$ on x_i. Suppose x_i is accepted. Then we choose an accepting computation, and take a string w of length $2^{(2C_i)^2}$ not queried in the computation and queried in the computation of $DEP_i^{H(s-1)}$ on z_i. Such a w exists. For, the number of all strings queried in both computation is at most

$$h_i(2C_i-1)+h_i(2C_i)=2^{p_i(2C_i-1)}+2^{p_i(2C_i)} < 2^{\beta}, \text{ where } \beta=2^{(2C_i)^2}.$$

Claim 1. (4) holds.

[Proof is obvious by the construction.]

Claim 2. For every i, DEP_i^H rejects z_i iff $DEP_i^{H(s)}$ rejects z_i for $s=3i+1$.

[Proof. Consider both computations on z_i of DEP_i^H and $DEP_i^{H(s)}$. Length of queried strings are $<h_i(2C_i-1)$, and lengths of strings put in H at later stages $3j+1$ or $3j+2>3i+1$ are $\geq 2^{(2C_{i+1}-1)^2}> h_i(2C_i)$ by (1). Lengths of strings put in H at later stage $3j>3i+1$ also are $>h_i(2C_i)$. Further if a string of length $2^{(2C_i-1)^2}$ or $2^{(2C_i)^2}$ is in H, then it was not queried in the computation of $DEP_i^{H(s)}$ on z_i. So, the computation of $DEP_i^{H(s)}$ on z_i coincides with that of DEP_i^H.]

Claim 3. For every i, NEP_i^H accepts x_i iff $NEP_i^{H(s)}$ accepts x_i for $s=3i+2$.

[Proof. Consider any computation of $NEP_i^{H(s)}$ on x_i. By similar argument as above, we see that lengths of strings put in H at later stages$>3i+2$ are larger than $h_i(2C_i)$. And only strings of length $<h_i(2C_i)$ are contained in $H(s)$. If $NEP_i^{H(s)}$ accepts x_i, then a string w of length $2^{(2C_i)^2}$ is in H and w is no queried in a chosen accepting computation. So the computation coincides with a computation of NEP_i^H on x_i. Hence NEP_i^H accepts x_i. If $NEP_i^{H(s)}$ rejects x_i, then no string of length $2^{(2C_i)^2}$ is in H. Since $H(s)$, of course, contains no string of length $2^{(2C_i)^2}$, any computation of $NEP_i^{H(s)}$ on x_i is a computation of NEP_i^H on x_i, too. Hence NEP_i^H rejects x_i.]

By Claim 1, $L_{\mathrm{odd}}(H)\in \mathrm{NEPT}^H \cap \text{co-}\mathrm{NEPT}^H$, as noted above. By Claim 2, for every i:

DEP_i^H rejects z_i iff $DEP_i^{H(3i+1)}$ rejects z_i iff $\exists y[0y\in H$ and $|0y|=2^{|z_i|^2}]$ iff $z_i\in L_{\mathrm{odd}}(H)$.

So, $L_{\mathrm{odd}}(H)\notin \mathrm{DEPT}^H$ and hence $L_{\mathrm{odd}}(H)\in \mathrm{NEPT}^H\cap\text{co-}\mathrm{NEPT}^H-\mathrm{DEPT}^H$.

By Claim 3, for every i:

NEP_i^H accepts x_i iff $NEP_i^{H(3i+2)}$ accepts x_i iff $\exists y[y \in H$ and $|y| = 2^{|x_i|^2}]$ iff $x_i \in L_{even}(H)$ iff $x_i \notin \overline{L_{even}(H)}$. So, $\overline{L_{even}(H)} \notin \mathrm{NEPT}^H$ and hence $L_{even}(H) \in \mathrm{NEPT}^H - \text{co-}\mathrm{NEPT}^H$. □

Acknowledgment

I would like to thank Professor Hisao Tanaka for his valuable guidance.

References

1) T. Baker, J. Gill and R. Solovay: Relativizations of the $P=?$ NP question, SIAM J. Comput., 4 (1975), pp. 431-442.
2) R. Book: Tally languages and complexity classes, Information and Control, 26 (1974), pp. 186-193.
3) R. Book: Comparing complexity classes, J. Comput. Syst. Sci., 9 (1974), pp. 213-229.
4) J. Hopcroft and J. Ullman: Introduction to automata theory, languages and computation, Addison-Wesly, Mass., (1979).
5) C. Kintala and P. Fischer: Refining nondeterminism in relativized polynomial-time bounded computations, SIAM J. Comput., 9 (1980), pp. 46-53.