数理解析研究所講究録
第695巻 1989年 178-187

POLYNOMIAL-TIME ACCESSIBILITY TO SYMMETRIC SOLUTIONS
(extended abstract)

Tomoyuki Yamakami (山上智幸)
Department of Mathematics, Rikkyo University, Nishi-ikebukuro, Tokyo, 171, Japan

ABSTRACT. Any NP∩coNP-set A is known to be characterized by a P-predicate structure, *i.e.*, $\mathrm{A}=\{x \mid (\exists y, |y|=p(|x|))\mathrm{Q}(x,y)\}=\{x \mid (\forall y, |y|=p(|x|))\mathrm{R}(x,y)\}$ for some P-predicates Q,R and polynomial p. Strings satisfying $|y|=p(|x|)\wedge(\mathrm{Q}(x,y)\vee\mathrm{R}(x,y))$ are called symmetric solutions for A on an input x. This paper studies the computational complexity of the accessibility to symmetric solutions for any NP∩coNP-sets.

§1. ***INTRODUCTION.*** One of central open problems in Complexity Theory is to determine whether P contains NP∩coNP or not. NP∩coNP behaves itself like a mixture of P and NP. It is closed under the polynomial-time (*abbr*. p-time) many-one reducibility while it is still unknown to possess a complete language. We notice that P≠NP is immediately concluded unless P=NP∩coNP.

Let us recall that every set A in NP∩coNP has the structure in the form

$$\mathrm{A}=\{x \mid (\exists y, |y|=p(|x|))\mathrm{Q}(x,y)\}=\{x \mid (\forall y, |y|=p(|x|))\mathrm{R}(x,y)\}$$

with two defining P-predicates Q,R and a defining polynomial p. We just call it a P-predicate structure of A. To investigate the complexity of A, we now turn our attention to its P-predicate structures. Let us say y a symmetric solution for A on an input x if $|y|=p(|x|)\wedge(\mathrm{Q}(x,y)\vee\neg\mathrm{R}(x,y))$ is satisfied. The symmetric solutions for A directly represent the complexity of P-redicate structures of A. For example, if we can access such a solution easily then A is easily recognizable. Hence it is less complex. So we may observe how intricate we access symmetric solutions for A.

Let us call f an access function for a set A∈NP∩coNP if f computes a symmetric solution for A on each input. This paper aims at investigating the computational complexity of each access function for NP∩coNP-sets. To see the complexity of computing access functions, here is used four types of classes consisting of restricted p-time computable functions with oracles. More precisely, we define and use the following classes of functions on Σ^* (say $\{0,1\}^*$), denoted by $\mathrm{PF_r(B)}$, which are computed by deterministic algorithms with an oracle set B in several restricted manners, where $\mathrm{r}\in\{\mathrm{T,tt,T}[O(logn)],\mathrm{btt}\}$. For any oracle machine M, Query(M,B,x) represents the set of all query words in the computation of $\mathrm{M^B}(x)$. Let Set($g(x)$) be a set $\{y_1,\cdots,y_k\}$ whenever $g(x)=\langle y_1,\cdots,y_k\rangle$ for a function g, where $\langle\ \rangle$ stands for a standard pairing operater. Write either #B or |B| to denote the cardinality of a set B. Let us define:

(1) A function f is in $\mathrm{PF_T(B)}$ if some p-time deterministic oracle transducer M computes f with B as an oracle. Let $\mathrm{PF_T}$ mean $\mathrm{PF_T}(\emptyset)$.

(2) A function f is in $\mathrm{PF_{tt}(B)}$ if $f\in\mathrm{PF_T(B)}$ via a transducer M which all query words appearring in the computation of $\mathrm{M^B}(x)$ is listable by some function $g\in\mathrm{PF_T}$, *i.e.*, $\forall x$(Query(M,B,x)=Set($g(x)$)). Call g a query list.

(3) A function f is in $\mathrm{PF}_{\mathrm{T}[O(logn)]}\mathrm{(B)}$ if $f\in\mathrm{PF_T(B)}$ via a transducer M, where #Query(T,B,x)$\leq c\,log|x|+d$ holds for all x by some absolute constants $c,d\geq0$.

(4) A function f is in $\mathrm{PF_{btt}(B)}$ if $f\in\mathrm{PF_{tt}(B)}$ via a transducer M, where for all x #Query(M,B,x) is bounded by some absolute constant.

In the continued section we introduce the concept of p-time accessibilities. A p-time r-accessible class with an oracle B, $\mathrm{PA_r(B)}$, is the collection of languages A in NP∩coNP such that an access function for A belongs to $\mathrm{PF_r(B)}$, where $\mathrm{r}\in\{\mathrm{T,tt,T}[O(logn)],\mathrm{btt}\}$. Our paper contains a classification of NP∩coNP-sets by the complexity to compute access functions for them. One of these classes $\mathrm{PA_T(B)}$ is shown to coincide with the polynomially helped class $\mathrm{P_{help}(B)}$ discussed by Schöning[16] and Ko [14].

Section 3 studies some fundamental properties of the class $\mathrm{PF_r(B)}$ for $\mathrm{r}\in\{\mathrm{T,tt,T}[O(logn)],\mathrm{btt}\}$. $\mathrm{PF_r(B)}$ can be considered as a natural extension of the language class $\mathrm{P_r(B)}$ [5,23]. One of main results in this section is the existence of a recursive oracle B such that $\mathrm{P_T(B)=P_{tt}(B)}$ and $\mathrm{PF_T(B)\neq PF_{tt}(B)}$.

In Section 4 we discuss several p-time accessible classes falling on P. It is shown that the restrained classes $\mathrm{PA_{btt}(B)}$ and $\mathrm{PA}_{\mathrm{T}[O(logn)]}\mathrm{(B)}$ collapse to P. Hence the p-time accessibility needs superlogarithmically-many oracle queries only if it is more complex. Moreover we see that $\mathrm{PA_{tt}(PSEL)}$ falls on P contrary to the fact $\mathrm{PA_T(PSEL)=PA_T(P/poly)}$, where PSEL is the class of p-selective sets.

Relations among $\mathrm{PA_r(B)}$, $\mathrm{r}\in\{\mathrm{T,tt}\}$, and other known classes are principally discussed in Section 5. A typical result we obtain here is that FewP∩coFewP is contained in both $\mathrm{PA_T(FewP)}$ and $\mathrm{PA_{tt}(NP)}$.

Let us further call B a r-self-accessible set if B is in $\mathrm{PA_r(B)}$, where $\mathrm{r}\in\{\mathrm{T,tt}\}$. In Section 6 we discuss these sets and strict-d-self-reducible sets, a variation of d-self-reducible sets.

Here is assumed reader's familiarity with [3,11,17,21]. It is remarked that our approach in this paper is closely related to the robust algorithms [14,16], the polynomial terseness [2] and the promise problems [20].

§2. ***POLYNOMIAL-TIME ACCESSIBILITY.*** To capture a P-predicate structure of a set A in NP∩coNP, we need to investigate the structural complexity of computing its access function f following p-time algorithms by several restricted ways using oracles. The problem we now meet is how to evaluate its complexity. Recall that we have already mentioned four types of function classes, *i.e.*, $\mathrm{PF}_r(\mathrm{B})$ for $r\in\{\mathrm{T},\mathrm{tt},\mathrm{T}[O(logn)],\mathrm{btt}\}$. So we merely examine whether $f\in\mathrm{PF}_r(\mathrm{B})$ holds or not for each f, however, from more general standpoint we here introduce and discuss the p-time r-accessible classes $\mathrm{PA}_r(\mathrm{B})$ of every language whose access function belongs to $\mathrm{PF}_r(\mathrm{B})$.

We start by defining $\mathrm{PA}_r(\mathrm{B})$ precisely, where $r\in\{\mathrm{T},\mathrm{tt},\mathrm{T}[O(logn)],\mathrm{btt}\}$.

DEFINITION 2.1. Assume $r\in\{\mathrm{T},\mathrm{tt},\mathrm{T}[O(logn)],\mathrm{btt}\}$ and $\mathrm{B}\subseteq\Sigma^*$.

(1) A language A belongs to the p-time r-accessible class with the oracle B, $PA_r(B)$, if and only if there exist two defining P-predicates Q, R, a defining polynomial p and a function $f\in\mathrm{PF}_r(\mathrm{B})$ such that

(i) $\mathrm{A}=\{x\mid(\exists y,|y|=p(|x|))\mathrm{Q}(x,y)\}=\{x\mid(\forall y,|y|=p(|x|))\mathrm{R}(x,y)\}$, and

(ii) $\forall x[|f(x)|=p(|x|)\wedge(\mathrm{Q}(x,f(x))\vee\neg\mathrm{R}(x,f(x)))]$.

(2) For a complexity class $\mathscr{C}$, $PA_r(\mathscr{C})$ is the union of $\mathrm{PA}_r(\mathrm{B})$ for every oracle set $\mathrm{B}\in\mathscr{C}$.

Notice that the number of symmetric solutions on each input is clearly more than zero. Thus, for each access function f,

$$\mathrm{A}=\{x\mid\mathrm{Q}(x,f(x))\}=\{x\mid\mathrm{R}(x,f(x)\}$$

is always satisfied.

Let us see some basic natures of $\mathrm{PA}_r(\mathrm{B})$ in the following two propositions. Their proofs are obtainable from definition without difficulty.

PROPOSITION 2.2. *Let r be in* $\{T,tt,T[O(logn)],btt\}$.

(1) $A\in PA_r(C)$ *and* $B\in PA_r(C)$ *imply* $A\oplus B\in PA_r(C)$.

(2) $P_r(B)\subseteq P_r(C)$ *implies* $PA_r(B)\subseteq PA_r(C)$.

(3) $PA_r(B)\subseteq P_r(B)\cap(NP\cap coNP)$.

(4) $PA_{btt}(B)\subseteq PA_{T[O(logn)]}(B)\subseteq PA_{tt}(B)\subseteq PA_T(B)$.

PROPOSITION 2.3. *Let* $r\in\{T,tt,btt\}$.

(1) $A\in P_r(B)$ *and* $B\in PA_r(C)$ *imply* $A\in PA_r(C)$.

(2) $A\in PA_r(B)$ *and* $B\in P_r(C)$ *imply* $A\in PA_r(C)$.

(3) $A\in PA_r(B)$ *and* $B\in PA_r(C)$ *imply* $A\in PA_r(C)$.

It is noted that $\mathrm{PA}_r(\emptyset)=\mathrm{PA}_r(\mathrm{P})=\mathrm{P}$ holds for every $r\in\{\mathrm{T},\mathrm{tt},\mathrm{T}[O(logn)],\mathrm{btt}\}$, and also $\mathrm{PA}_r(2^{\Sigma^*})\subseteq\mathrm{NP}\cap\mathrm{coNP}$ is shown, where 2^{Σ^*} denotes the power set of Σ^*.

Let Q,R be any P-predicates and p be any polynomial defining A, *i.e.*,

$$A=\{x|(\exists y,|y|=p(|x|))Q(x,y)\}=\{x|(\forall y,|y|=p(|x|))R(x,y)\}.$$

Then we define two prefix sets Pref(Q,R,p) and MinPref(Q,R,p) by:

$$\mathrm{Pref}(Q,R,p)=\{\langle x,u\rangle|\exists v[|uv|=p(|x|)\wedge(Q(x,uv)\vee\neg R(x,uv))]\},$$

$$\mathrm{MinPref}(Q,R,p)=\{\langle x,0^i\rangle|(y)_i=1,\text{ where } y \text{ is lexicographically the minimal string s.t. } |y|=p(|x|)\wedge(Q(x,y)\vee\neg R(x,y))\},$$

where $(y)_i$ expresses the i-th bit of the string y. Pref(Q,R,p) must be d-self-reducible and it satisfies MinPref(Q,R,p)$\in P_T$(Pref(Q,R,p)).

To show a given set A is in either $PA_T(B)$ or $PA_{tt}(B)$, the next lemma is so useful.

LEMMA 2.4. *Suppose Q,R are any P-predicates and p is any polynomial defining a set A. Then the followings hold.*

(1) $A\in PA_T(Pref(Q,R,p))$.

(2) $A\in PA_{tt}(MinPref(Q,R,p))$.

PROPOSITION 2.5. $PA_T(NP)=PA_{tt}(\Delta_2^p)=NP\cap coNP$.

It is still unknown for us whether $PA_{tt}(NP)=NP\cap coNP$ is infered or not. However, the consequence of Proposition 2.5 can be generalized to the following theorem.

THEOREM 2.6. $PA_T(B)=PA_{tt}(P_T(B))$. *In paticular* $PA_T(\mathscr{C})=PA_{tt}(\mathscr{C})$ *if a complexity class* $\mathscr{C}$ *satisfies either* $P_T(\mathscr{C})=P_{tt}(\mathscr{C})$ *or* $P_T(\mathscr{C})=\mathscr{C}$.

This result is widely applicable. For example, we obtain $PA_T(\mathscr{C})=PA_{tt}(\mathscr{C})$ for the following classes $\mathscr{C}$: P,NP∩coNP,BPP,ZPP,P/poly,(NP∩coNP)/poly.

Schöning[16] introduced the concept of polynomially helping the robust algorithms. In the last place of this section, we see the relationship of two concepts, the p-time T-accessibility and the polynomially helping the robust algorithms. This was first announced (in a different form) only for self 1-helpers by Balcázar[4].

Any deterministic Turing machine M is said to be robust if $L(M,\emptyset)=L(M,A)$ holds for every oracle A. Let us assume M is a robust machine. An oracle B helps M (or $L(M,\emptyset)$) if M^B always halts in p-time. $P_{help}(B)$ denotes the class of languages helped by B.

THEOREM 2.7. $PA_T(B)=P_{help}(B)$.

§3. ***$PF_r(B)$-FUNCTIONS.*** This section discusses basic properties of the function classes $PF_r(B)$, $r\in\{T,tt,T[O(logn)],btt\}$. Let us first see $PF_r(B)$ to be a natural extention of ordinary language classes $P_r(B)$ [5,23]. Let χ_A denote the characteristic function for A.

PROPOSITION 3.1. *Assume* $r\in\{T,tt,T[O(logn)],btt\}$ *and* $A,B\subseteq\Sigma^*$.

(1) $A\in P_r(B) \iff \chi_A\in PF_r(B)$.

(2) $PF_{btt}(B)\subseteq PF_{T[O(logn)]}(B)\subseteq PF_{tt}(B)\subseteq PF_T(B)$.

PROPOSITION 3.2. *Let r be in* $\{T,tt,T[O(logn)],btt\}$.

(1) $PF_r(B)\subseteq PF_r(C) \iff P_r(B)\subseteq P_r(C)$.

(2) $PF_{btt}(B) \subseteq PF_{tt}(C) \iff P_{btt}(B) \subseteq P_{tt}(C)$.

(3) $PF_{tt}(B) \subseteq PF_{T}(C) \iff P_{tt}(B) \subseteq P_{T}(C)$.

(4) $PF_{tt}(B) \subseteq PF_{btt}(C) \implies P_{tt}(B) \subseteq P_{btt}(C)$.

(5) $PF_{T}(B) \subseteq PF_{tt}(C) \implies P_{T}(B) \subseteq P_{tt}(C)$.

Corresponding to (4) and (5), the inverse implications do not always hold because the following oracles exist. Let **N** be the set of nonnegative integers.

***THEOREM* 3.3.** *There exist two recursive oracles A and B such that*

(1) $P_{btt}(A)=P_{tt}(A)$ and $PF_{btt}(A) \neq PF_{tt}(A)$, and

(2) $P_{tt}(B)=P_{T}(B)$ and $PF_{tt}(B) \neq PF_{T}(B)$.

More strongly we can claim that REC-TALLY, the set of recursive tally sets, separates PF_{tt}(REC-TALLY) from $\mathrm{PF}_{T[O(logn)]}(2^{\Sigma^*})$. This shows a limitation of the power of functions obtained by logarithmically-many oracle queries. Note that PF_{tt}(REC-TALLY)=PF_{T}(REC-TALLY).

***THEOREM* 3.4.** *There exists a set $S \in REC\text{-}TALLY$ such that $PF_{T[O(logn)]}(2^{\Sigma^*}) \neq PF_{tt}(S)$.*

§4. ***COLLAPSING CLASSES TO P.*** In Proposition 2.5, it was shown that PA_{T}(NP) and $\mathrm{PA}_{tt}(\Delta_2^{\mathrm{P}})$ exactly correspond to NP∩coNP. In the opposite direction, this section studies several p-time accessible classes which collapse to P.

We start with the case of p-time btt- and T[$O(logn)$]-accessible classes. Observations in Section 3 showed that $\mathrm{PF}_{T[O(logn)]}$(B)-functions are no more powerful than PF_{tt}(REC-TALLY)-functions. Therefore, if an access function for a given set A can be chosen among PF_{btt}(B) or $\mathrm{PF}_{T[O(logn)]}$(B), then A is interestingly shown to be computed just by a P-algorithm without any help of oracles. This intuitively indicates that the p-time accessibility needs superlogarithmically-many oracle queries only if it is more complex than P itself. Now let us prove $\mathrm{PA}_{T[O(logn)]}(2^{\Sigma^*})=\mathrm{PA}_{btt}(2^{\Sigma^*})$=P.

***THEOREM* 4.1.** *$PA_{T[O(logn)]}(2^{\Sigma^*})=PA_{btt}(2^{\Sigma^*})=P$.*

Recently Ko[14] showed Strong-P/poly is no-helper, in other words, PA_{T}(Strong-P/poly)=P. Assume $\mathscr{C}$ is either P, NP∩coNP or NP. His argument can be generalized for Strong-$\mathscr{C}$/poly. We define a language B to be in Strong-$\mathscr{C}$/poly if there exist a set B∈$\mathscr{C}$ and integers $c,d \geq 0$ such that, for every $n>0$, some w satisfies both relations $|w| \leq clogn+d$ and $(\forall x, |x| \leq n)[x \in A \iff \langle x,w \rangle \in B]$.

***PROPOSITION* 4.2.** *$PA_{T}(Strong\text{-}\mathscr{C}/poly)=PA_{T}(\mathscr{C})$, where $\mathscr{C}$ is either P, NP∩coNP or NP.*

Turn to the p-time tt-accessible classes which can collapse to P. We next see two collapsing results concerning the concepts of the polynomial terseness [2] and the p-selectivity [13]. Let us begin with (f,g)-pterse sets, an extension of pterse sets.

DEFINITION 4.3. Let f,g be functions on **N**.

(1) A set B is *(f,g)-pterse* if and only if, for any p-time deterministic oracle Turing machine M, it holds that:

$$(*)\ \forall n\forall\{x_1,\cdots,x_{f(n)}\}[f(n)\geq g(n)\wedge\#\mathrm{Query}(\mathrm{M},\mathrm{B},\langle x_1,\cdots,x_{f(n)}\rangle)<g(n)] \Longrightarrow$$
$$\exists k\exists\{y_1,\cdots,y_{f(k)}\}[\mathrm{M}^{\mathrm{B}}(\langle y_1,\cdots,y_{f(k)}\rangle)\neq\langle\chi_{\mathrm{B}}(y_1),\cdots,\chi_{\mathrm{B}}(y_{f(k)})\rangle].$$

(2) A set B is *(poly,logpoly)-pterse* if B is $(p,\log q)$-pterse for any polynomials p and q.

A remark we raise now is only that the k-penseness is equivalent to the (k,k)-pterseness. In the following theorem we use the counting argument directly.

THEOREM 4.4. *$PA_{tt}(B)=P$ unless B is (poly,logpoly)-pterse.*

Selman[18] showed $\mathrm{P_T(PSEL)=P_T(TALLY)}$, while $\mathrm{P_{tt}(PSEL)\neq P_T(PSEL)}$ is recently proved by Watanabe[24]. This difference is clear in the case $\mathrm{PA_r(PSEL)}$. From Selman's result,

$$\mathrm{PA_T(PSEL)=PA_T(TALLY)=PA_T(P/poly)}$$

holds, however, $\mathrm{PA_{tt}(PSEL)}$ collapses to P. Before showing the collapse of $\mathrm{PA_{tt}(PSEL)}$, we introduce wider class of bounded-p-selective sets containing PSEL.

Recall that preorder $\preccurlyeq$ on Σ^* is partially p-time computable if there exists a $\mathrm{PF_T}$-function f such that

(i) $x\preccurlyeq y$ and $y\not\preccurlyeq x$ imply $f(x,y)=f(y,x)=x$,

(ii) $x\preccurlyeq y$ and $y\preccurlyeq x$ imply $f(x,y)=f(y,x)\in\{x,y\}$, and

(iii) $x\not\preccurlyeq y$ and $y\not\preccurlyeq x$ imply $f(x,y)\notin\{x,y\}$.

We simply write $x\approx y$ if $x\preccurlyeq y$ and $y\preccurlyeq x$, and denote by $\preccurlyeq'$ the induced order on $\Sigma^*/\approx$ [13].

DEFINITION 4.5.

(1) A preorder $\preccurlyeq$ is *b-linear* if there is an integer $m\geq 0$ such that, for any $n>0$, some disjoint partition $\{\mathrm{B}_1,\cdots,\mathrm{B}_m\}$ of $\Sigma^{\leq n}$ has the following properties:

(i) $x\in\mathrm{B}_i$ and $y\in\mathrm{B}_i$ imply either $x\preccurlyeq y$ or $y\preccurlyeq x$, and

(ii) $x\in\mathrm{B}_i$, $y\in\mathrm{B}_k$ and $i\neq k$ imply both $x\not\preccurlyeq y$ and $y\not\preccurlyeq x$.

(2) A set A is *bounded-p-selective* if there exist a partially p-time computable preorder $\preccurlyeq$ and an integer $k>0$ such that

(i) $\preccurlyeq'$ is b-linear on $\Sigma^*/\approx$, and

(ii) $\mathrm{A}=\cup\{\mathrm{B}_i\,|\,\mathrm{B}_i\in\mathrm{Seg}(\alpha_i),\alpha_i$ is a $\preccurlyeq'$-chain$,1\leq i\leq k\}$,

where $\mathrm{Seg}(\alpha)$ is the set of all initial segments on α. *bPSEL* denotes the family of all bounded-p-selective sets.

We notice that bPSEL is contained between PSEL and the class wPSEL of weakly p-selective sets [13]. Hence it is infered that, for $\mathrm{r}\in\{\mathrm{T,tt}\}$,

$$\mathrm{PA_r(PSEL)\subseteq PA_r(bPSEL)\subseteq PA_r(wPSEL)}.$$

Let us claim the collapse of $\mathrm{PA_{tt}(bPSEL)}$ to P.

THEOREM 4.6. *$PA_{tt}(bPSEL)=PA_{tt}(PSEL)=P$.*

§5. ***RELATIONS AMONG CLASSES IN NP$\cap$coNP.*** We have just seen that p-time T[$O(logn)$]- and btt-accessible classes fall into P in the previous section. Therefore we devote our attention to the elucidation of structural complexities of the remaining accessible classes among well-known complexity subclasses of NP$\cap$coNP.

We here denote $\Theta^{\mathrm{p}}_{k+1}$ by $\mathrm{P}_{\mathrm{T}[O(logn)]}(\Sigma^{\mathrm{p}}_{k})$ [23] for $k \geq 0$. In [14], Hemachandra proved $\Theta^{\mathrm{p}}_{2}=\mathrm{P}_{tt}(\mathrm{NP})$. On the other hand Cai *et al.*[7] showed $\mathrm{BH}=\mathrm{P}_{\mathrm{btt}}(\mathrm{NP})$ for the Boolean hierarchy BH. Thus the inclusion relation $\mathrm{NP}\subseteq\mathrm{BH}\subseteq\Theta^{\mathrm{p}}_{2}\subseteq\Delta^{\mathrm{p}}_{2}$ holds evidently. We remark that it is still open whether any one of these inclusions becomes proper or not. Nevertheless, the following result is simply obtainable.

PROPOSITION 5.1. $PA_T(NP\cap coNP)\subseteq PA_{tt}(NP)=PA_{tt}(BH)=PA_{tt}(\Theta^{p}_{2})$.

Let SPARSE be the class of sparse sets and PCLOSE be that of p-close sets. By the recent work of Book and Ko[5], it holds that

$$\mathrm{P}_{\mathrm{T}}(\mathrm{P/poly})=\mathrm{P}_{r}(\mathrm{SPARSE})=\mathrm{P}_{r}(\mathrm{TALLY})=\mathrm{P}_{r}(\mathrm{PCLOSE}),$$

where $r\in\{\mathrm{T},\mathrm{tt}\}$. Then we immediately get:

PROPOSITION 5.2. $PA_T(P/poly)=PA_r(SPARSE)=PA_r(TALLY)=\mathrm{PA}_r(PCLOSE)$ *for* $r\in\{T,tt\}$.

Now denote by $\mathrm{EL}^{\mathrm{p}}_{k}$ the k-th level of the extended low hierarchy [17]. $\mathrm{PA}_{tt}(\mathrm{EL}^{\mathrm{p}}_{3})$ contains all these classes.

PROPOSITION 5.3. $PA_T(NP\cap coNP)\cup PA_T(P/poly)\subseteq PA_T((NP\cap coNP)/poly)\subseteq PA_{tt}(EL^{p}_{3})$.

Let us recall that REC-TALLY is the collection of recursive tally sets. Similar to Book and Ko's result, $\mathrm{P}_{\mathrm{T}}(\mathrm{APT})$ can be shown to coincide with the p-time Turing reduced class from REC-TALLY. It should be noted that APT is the family of languages A so that some deterministic Turing machine recognizing A halts in p-time on every input in Σ^{*}-S, where S is a sparse set. Hence every APT-set is recursive. It is known that every set in APT is not many-one complete for NP unless P=NP [17].

LEMMA 5.4. $P_T(APT)=P_{tt}(APT)=P_r(REC\text{-}TALLY)=P_r(REC\text{-}SPARSE)$ *for* $r\in\{T,tt\}$.

PROPOSITION 5.5. $PA_T(APT)=PA_{tt}(APT)=PA_r(REC\text{-}TALLY)=PA_r(REC\text{-}SPARSE)$, *where* $r\in\{T,tt\}$.

It reminds us that ZPP is a natural probabilistic class contained in BPP$\cap$(NP$\cap$coNP) [8]. It is known that every ZPP-set A is chracterized, for any polynomial q, by some defining P-predicates Q,R and defining polynomial p which satisfy $\#\{y\,|\,|y|=p(|x|),\neg\mathrm{Q}(x,y)\wedge\mathrm{R}(x,y)\}<2^{p(|x|)-q(|x|)}$ for all input x (see e.g.[17]). Let us see that ZPP is located between two p-time accessible classes $\mathrm{PA}_{\mathrm{T}}(\mathrm{BPP})$ and $\mathrm{PA}_{\mathrm{T}}(\mathrm{APT})$.

THEOREM 5.6. $PA_T(BPP)\subseteq ZPP\subseteq PA_T(APT)$.

Another well-known class belonging to NP$\cap$coNP is the class UP$\cap$coUP. Due to Ko's insight [14], we obtain:

PROPOSITION 5.7. $PA_T(UP\cap coUP)=PA_{tt}(UP\cap coUP)=UP\cap coUP$.

For the case $\mathrm{FewP}\cap\mathrm{coFewP}$, we can use Hemachandra's technique of hiding informations to a NP-set [12] to see $\mathrm{FewP}\cap\mathrm{coFewP}\subseteq\mathrm{PA_T}(\mathrm{NP})$. Recall that every FewP-set has the property which the number of its solutions on each input is bounded by some absolute polynomial [1].

THEOREM 5.8. $\mathit{FewP}\cap\mathit{coFewP}\subseteq \mathit{PA_T}(\mathit{FewP})\cap \mathit{PA_{tt}}(\mathit{NP})$.

We do not have any proof to show that $\mathrm{FewP}\cap\mathrm{coFewP}\subseteq\mathrm{PA_T}(\mathrm{FewP}\cap\mathrm{coFewP})$ and $\mathrm{PA_T}(\mathrm{FewP}\cap\mathrm{coFewP})\subseteq\mathrm{FewP}\cap\mathrm{coFewP}$ hold contrary to Proposition 5.7.

As for the class $\oplus\mathrm{P}$ [15], the situation is exceedingly different. Note that any $\oplus$P-set consists of strings of which the number of solutions is odd. It follows from definition that $\oplus\mathrm{P}$ is closed under complement. Moreover $\oplus\mathrm{P}$ contains FewP [11] and clearly UP. Notice that any relation between NP and $\oplus\mathrm{P}$ is not known.

PROPOSITION 5.9. $\mathit{PA_T}(\oplus P)=\mathit{PA_{tt}}(\oplus P)\subseteq\oplus P\cap(\mathit{NP}\cap \mathit{coNP})$.

§6. ***SELF-ACCESSIBILITY***. Let us first define r-self-accessible sets for $r\in\{\mathrm{T},\mathrm{tt}\}$.

DEFINITION 6.1. Let $r\in\{\mathrm{T},\mathrm{tt}\}$. We say a set A to be *r-self-accessible* if $\mathrm{A}\in\mathrm{PA_r}(\mathrm{A})$ is satisfied. $\mathit{SACCESS_r}$ is the collection of all r-self-accessible sets.

Clearly all sets in P are tt-self-accessible and

$$\mathrm{SACCESS_{tt}}\subseteq\mathrm{SACCESS_T}\subseteq\mathrm{PA_T}(\mathrm{NP}\cap\mathrm{coNP})$$

holds. By Ko[14], every set in $\mathrm{DSRED}\cap\mathrm{coDSRED}$ is T-self-accessible, where DSRED denotes the family of d-self-reducible sets. In the followings, we treat some natural subsets of $\mathrm{NP}\cap\mathrm{coNP}$ and show that they have the property of the self-accessibility.

Here we recall the d-self-reducible sets [13,19]. A set A is d-self-reducible if and only if there is a p-time computable partial ordering $<_\mathrm{A}$ on Σ^*, a polynomial p and a function $f\in\mathrm{PF_T}$ such that:

(1) any $<_\mathrm{A}$-decreasing chain begining with x has length $\leq p(|x|)$,

(2) $\forall x\forall y(y\in\mathrm{Set}(f(x)) \implies y<_\mathrm{A}x)$,

(3) $\forall x[\mathrm{Set}(f(x))=\emptyset \implies (x\in\mathrm{A} \iff f(x)=1)]$, and

(4) $\forall x[\mathrm{Set}(f(x))\neq\emptyset \implies (x\in\mathrm{A} \iff \exists y\in\mathrm{Set}(f(x))(y\in\mathrm{A}))]$.

All d-self-reducible sets are natural self-reducible sets which are many-one complete for NP. On the analogy of the T-self-accessibility of $\mathrm{DSRED}\cap\mathrm{coDSRED}$, we here define the restricted version of the d-self-reducible sets.

DEFINITION 6.2. A set A is *strict-d-self-reducible* if $\mathrm{A}\in\mathrm{DSRED}$ via a $\mathrm{PF_T}$-function f and there exists a function $g\in\mathrm{PF_T}$ satisfying $\forall x(\mathrm{Q}(f,x)=\mathrm{Set}(g(x)))$, where $\mathrm{Q}(f,x)=\cup_k\mathrm{Q}_k(f,x)$, $\mathrm{Q}_0(f,x)=\{x\}$ and $\mathrm{Q}_{k+1}(f,x)=\cup\{\mathrm{Set}(f(y))\,|\,y\in\mathrm{Q}_k(f,x)\}$. Denote by *sDSRED* the class of all strict-d-self-reducible sets.

By this definition all strict-d-self-reducible sets are d-self-reducible. Moreover, since SAT is obviously strict-d-self-reducible, so

$$P_m(\mathrm{sDSRED})=P_m(\mathrm{DSRED})=P_m(\mathrm{NP})$$

is infered although whether sDSRED=DSRED is not known.

The next lemma is due to Selman′s argument [19].

LEMMA 6.3.

(1) *If* $A\in DSRED\cap coDSRED$ *then there exist P-predicates* Q,R *and a polynomial* p *defining* A *such that* $P_T(A)=P_T(Pref(Q,R,p))$.

(2) *If* $A\in sDSRED\cap cosDSRED$ *then there exist P-predicates* Q,R *and a polynomial* p *defining* A *such that* $P_{tt}(A)=P_{tt}(MinPref(Q,R,p))$.

Section 2 introduced two types of prefix sets $\mathrm{Pref}(Q,R,p)$ and $\mathrm{MinPref}(Q,R,p)$ reduced from P-predicate structures of basic sets. These prefix sets directly reflect symmetric solutions for basic sets. So we now take two special classes consisting of these prefix sets which are computable in p-time using basic sets as oracles.

DEFINITION 6.4. *PREF* (*MINPREF resp.*) contains all languages A such that $\mathrm{Pref}(Q,R,p)\in P_T(A)$ ($\mathrm{MinPref}(Q,R,p)\in P_{tt}(A)$ *resp.*) holds for some P-predicates Q,R and some polynomial p defining A.

At the last of this section we prove the desired relations to be hold, *i.e.*, the sets defined above are all self-accessible.

THEOREM 6.5.

(1) $DSRED\cap coDSRED\subseteq PREF\subseteq SACCESS_T$.

(2) $sDSRED\cap cosDSRED\subseteq MINPREF\subseteq SACCESS_{tt}$.

REFERENCES

[1] E.W.Allender and R.S.Rubinstein, P-printable sets, *SIAM J. Comput.* 17 (1988) 1193-1202.

[2] A.Amir and W.I.Gasarch, Polynomial terse sets, *Inform. Comput.* 77 (1988) 37-56.

[3] J.L.Balcázar, J.Díaz and J.Gabarró , *Structural Complexity I* (Springer-Verlag, 1988).

[4] J.L.Balcázar, Self-reducibility, *Lecture Notes in Computer Science* 247 (1987) 136-147.

[5] R.V.Book and K.Ko, On sets truth-table reducible to sparse sets, *SIAM J. Comput.* 17 (1988) 903-919.

[6] R.V.Book, Sparse sets, tally sets, and polynomial reducibilities, *Lecture Notes in Computer Science* 324 (1988) 1-13.

[7] J.Cai, T.Gundermann, J.Hartmanis, L.A.Hemachandra, V.Sewelson, K. Wagner and G.Wechsung, The Boolean hierarchy I: structural properties, *SIAM J. Comput.* 17 (1988) 1232-1252.

[8] J.Gill, Computational complexity of probabilistic Turing machines, *SIAM J. Comput.* 6 (1977) 675-695.

[9] J.Hartmanis and L.Hemachandra, On sparse oracles separating feasible complexity classes, *Lecture Notes in Computer Science* 210 (1986) 321-333.

[10] J.H.Hartmanis and L.A.Hemachandra, Complexity classes without machines: on complete languages for UP, *Theoret. Comput. Sci.* 58 (1988) 129-142.

[11] L.A.Hemachandra, Structure of complexity classes: separations, collapses, and completeness, *Lecture Notes in Computer Science* 324 (1988) 59-72.

[12] L.A.Hemachandra, The strong exponential hierarchy collapses, *Proc. ACM Symposium of Theory of Computing*, 1987, 110-122.

[13] K.Ko, On self-reducibility and weak p-selectivity, *J. Comput. System Sci.* 26 (1983) 209-221.

[14] K.Ko, On helping by robust oracle machines, *Proc. Conf. Structure in Complexity Theory*, 1987, 182-190.

[15] C.H.Papadimitrious and S.K.Zachos, Two remarks on the power of counting, *Lecture Notes in Computer Science* 145 (1982) 269-275.

[16] U.Schöning, Robust algorithms: a different approach to oracles, *Theoret. Comput. Sci.* 40 (1985) 57-66.

[17] U.Schöning, Complexity and Structure, *Lecture Notes in Computer Science* 211 (1986).

[18] A.L.Selman, P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP, *Math. Systems Theory* 13 (1979) 55-65.

[19] A.L.Selman, Natural self-reducible sets, *SIAM J. Comput.* 17 (1988) 989-996.

[20] A.L.Selman, Promise problems complete for complexity classes, *Inform. Comput.* 78 (1988) 87-98.

[21] L.Stockmeyer, Classifying the computational complexity of problems, *J. Symb. Logic* 52 (1987) 1-43.

[22] L.G.Valiant and V.V.Vazirani, NP is as easy as detecting unique solutions, *Theoret. Comput. Sci.* 47 (1986) 85-93.

[23] K.W.Wagner, On restricting the access to an NP-oracle, *Lecture Notes in Computer Science* 317 (1988) 682-696.

[24] O.Watanabe, $\leq^{p}_{T}\mathrm{PSELECT} \neq \leq^{p}_{tt}\mathrm{PSELECT}$, manuscript, 1988.

[25] T.Yamakami, Computational complexity of languages counting random oracles, to appear in *Lecture Notes in Mathematics*.

[26] S.Zachos, Probabilistic quantifiers and games, *J. Comput. System Sci.* 36 (1988) 433-451.