On One Query Self-Reducible Sets

AUTHOR(S):

OGIWARA, Mitsunori; LOZANO, Antoni

数理解析研究所講究録
第 754 巻 1991 年 45-56

On One Query Self-Reducible Sets

*Mitsunori OGIWARA**
Tokyo Institute of Technology

Antoni LOZANO†
Universitat Politècnica de Catalunya

Abstract

We study one word-decreasing self-reducible sets, which are introduced by Lozano and Torán [21]. These are usual self-reducible sets with the peculiarity that the self-reducibility machine makes at most one query to a word lexicographically smaller than the input. We can prove that for any class K chosen from {NP, PP, $C_=P$, MOD_2P, MOD_3P, $\cdots$} it holds that (1) if there is a sparse $\leq^P_{btt}$-hard set for K then K is in P, and (2) if there is a sparse $\leq^{SN}_{btt}$-hard set for K then K is in NP $\cap$ co-NP. The main result also shows that if there is a sparse $\leq^{SN}_{btt}$-hard set for PSPACE then PSPACE = NP. This generalizes the result from Ogiwara and Watanabe [24] to the mentioned complexity classes.

1 Introduction

One of the central roles in the study of structural complexity theory resides in finding structural differences or similarities among complexity classes. Since almost every complexity class is defined by using some resource bounded computational model, finding relationships among such classes sometimes requires us to specify different computational models, and therefore, it seems tremendously hard to find such relationships. Above all, as subclasses of PSPACE, there have been introduced many complexity classes [4,9,12,25,28,35]. For example, Gill, and independently Simon defined PP as the class of sets having probabilistic polynomial time acceptors with error probability $< 1/2$ [12,28]. Papadimitriou and Zachos defined $\oplus$P as the class of sets for which there is a polynomial time nondeterministic Turing machine such that a string is in the set if and only if the machine has an odd number of accepting computation paths for the string [25]. Based on this definition, Beigel, Gill, and Hertrampf, and independently, Cai and Hemachandra defined MOD_kP as the class of sets for which there exists a polynomial time nondeterministic Turing machine such that for every x, x is in the set if and only if the number of accepting computation paths of the machine on input x is not a multiple of k, where $k \geq 2$ [4,9]. Wagner introduced $C_=P$ (respectively, CP) as the class of sets for which there exist a polynomial time computable function and a polynomial time nondeterministic Turing machine such that a string x is in the set if and only if the number of accepting computation paths of the machine on the input x is equal to (respectively, larger than or equal to) the value of the function for x [35]. Also, he showed that CP is equal to PP. So, concerning the classes that are located between NP and PSPACE, the

*Department of Information Sciences, Ookayama, Tokyo 152, Japan. email: ogiwara@is.titech.ac.jp.
†Department of Software (L.S.I.). Pau Gargallo 5, 08028 Barcelona, Spain. email: lozano@lsi.upc.es.

most important unsolved questions are the following:

"Does the polynomial time hierarchy have infinite levels ?," and

"Is any class defined above included in the polynomial time hierarchy ?"

Some results that settle these questions partially have been obtained in [5,6,18,29,31,32,33]. Nevertheless, until now, neither of the above questions is solved.

On the other hand, it is widely known that we can classify sets into some categories by using different reducibilities to sets of small density [8,16,17]. Especially, a set having a census function bounded above by some polynomial is called sparse. Relative to this notion, the following questions have been considered by many researchers [7, 11,14,15,22,24,34,36,38,39].

"For a class K and for a reducibility $\leq_r$, is there any sparse set to which every set in K is $\leq_r$-reducible ?," or "Suppose that every set in K is $\leq_r$-reducible to some sparse set. Will then any unexpected inclusions follow ?"

As a matter of fact, after Berman and Hartmanis conjectured that all $\leq^P_m$-complete sets for NP are P-isomorphic, and thereby conjectured that there are neither sparse nor co-sparse $\leq^P_m$-hard sets for NP[7], reducibilities of NP sets to sparse sets have been considered for a long time [11,22,23,34,36,39]. And the question whether NP having sparse $\leq^P_{btt}$-hard sets implies P = NP or not had been left open for a long time until Ogiwara and Watanabe solved the question affirmatively [24].

In order to settle this question, they introduced the notion of 'left-sets', which is a certain type of self-reducible structure, and showed that every 'left-set' in NP which is $\leq^P_{btt}$-reducible to sparse sets is already in P. The proof technique takes advantage of the structure of left-sets and of the structure resulting from the reduction to a sparse set, to show that any set with these two characteristics can be decided deterministically in polynomial time. Furthermore, Ogiwara extended the notion of left-sets and showed that the existence of $\leq^P_{btt}$ (respectively, $\leq^{SN}_{btt}$) hard sets for PP implies PP = P (respectively, PP = NP) [23]. So, in order to extend the result to more classes, it can seem useful to consider sets with a more rich internal structure than that of left-sets. We consider, for this goal, one word-decreasing self-reducible sets, which have the desired properties: its time complexity decreases if they are reducible to sparse sets, and we can find self-reducible sets of this type which are complete for the classes NP, PP, $\mathrm{MOD}_k\mathrm{P}$, $\mathrm{C}_=\mathrm{P}$, and PSPACE.

As the self-reducible structure, we will use one word-decreasing self-reducible sets introduced by Lozano and Torán [21]. The notion of one word-decreasing self-reducible sets is a variation of Balcázar's wdq self-reducible sets [2] and a generalization of left-sets [23,24]. A set A is one word-decreasing self-reducible if there exists a polynomial time deterministic oracle Turing machine which accepts A with oracle A itself in such a way that for every input x, the machine queries at most one string to its oracle and the query string is lexicographically smaller than x. We define strict one word-decreasing self-reducible sets as a restriction of the last ones. We show that

1. PP and $\mathrm{C}_=\mathrm{P}$ have a $\leq^P_m$-complete set which is one word-decreasing self-reducible, and
2. NP, $\mathrm{MOD}_2\mathrm{P}$, $\mathrm{MOD}_3\mathrm{P}$, $\cdots$, have a $\leq^P_m$-complete set which is strictly one word-decreasing self-reducible.

As for the reducibility, we will consider polynomial time bounded truth-table reducibility and strong nondeterministic polynomial time bounded

truth-table reducibility[1,19,20]. We will show the folloing theorems:

Theorem 3.5 *If a set A is 1-wd self-reducible and $\leq^{SN}_{btt}$ reducible to some sparse set, then A is in* NP$\cap$ *co-NP.*

Theorem 3.7 *If a set A is strictly 1-wd self-reducible and $\leq^{P}_{btt}$ reducible to some sparse set, then A is in* P.

Then, by combining these results, we will prove that for any class K chosen from {NP, PP, $C_=$P, MOD_2P, MOD_3P, $\cdots$}, it holds that

1. if there is a sparse $\leq^{P}_{btt}$-hard set for K then K is in P, and
2. if there is a sparse $\leq^{SN}_{btt}$-hard set for K then K is in NP $\cap$ co-NP.

Also, as PSPACE has 1-wd self-reducible complete sets, Theorem 3.5 implies that

3. if there is a sparse $\leq^{SN}_{btt}$-hard set for PSPACE then PSPACE = NP.

2 Preliminaries

We fix now some of the notation that will be used throughout this paper. We use the alphabet $\Sigma = \{0,1\}$, and define the basic notation about sets and words as it is done in [3]. By 'polynomials' we mean monotone nondecreasing polynomials. We will denote the cardinality of a set A with $\| A$. We assume the canonical lexicographic order on Σ^*. A string x is less than y (write $x < y$) if either (1) $|x| < |y|$ or (2) $|x| = |y|$ and there exist strings $u, v, w \in \Sigma^*$ such that $x = u0v$ and $y = u1w$. For a string $x \in \Sigma^* \setminus \{\lambda\}$, pred($x$) denotes the predecessor of x; that is, pred(x) is max$\{y : y < x\}$. Also, for a string $x \in \Sigma^*$, suc(x) denotes the successor of x; that is, suc(x) is min$\{y : y > x\}$. For a string x, ord(x) denotes $\| \{y : |y| = |x|$ and $y < x\}$. $\langle \cdot, \cdot \rangle$ denotes an easily-computable encoding of two strings into one string. We assume that for every x, x', y and y' in Σ^* with $|x| = |x'|$ and $|y| = |y'|$, (i) $|\langle x, y\rangle| = |\langle x', y'\rangle|$, (ii) if $y' =$ pred(y), then pred($\langle x, y\rangle$) $= \langle x, y'\rangle$, and (iii) if $x' < x$, then $\langle x, y\rangle > \langle x', y'\rangle$. For simplicity, for $k \geq 2$ and k strings $y_1, y_2, \cdots, y_k$, $\langle y_1, y_2, \cdots, y_k\rangle$ denotes $\langle\langle \ldots \langle y_1, y_2\rangle, \cdots\rangle, y_k\rangle$. Furthermore, N denotes the set of natural numbers.

Our computational model is the polynomial time Turing machine. We can assume w.l.o.g. that our nondeterministic Turing machines have polynomial clocks, that they have exactly $2^{p(|x|)}$ computation paths on input x, and any of these paths can be uniquely encoded by a word in $\Sigma^{=p(|x|)}$. For each $x \in \Sigma^*$, $\mathrm{acc}_N(x)$ and $\mathrm{rej}_N(x)$ denotes the set of all strings in $\Sigma^{=p(|x|)}$ representing accepting computation paths of N on x and rejecting computation paths of N on x, respectively.

A set S is sparse if there exists a polynomial p such that for every natural number n, $\| S^{\leq n} \leq p(n)$.

For a polynomial time deterministic Turing transducer M and for a string $x \in \Sigma^*$, $M(x)$ denotes the output of M on x. For a polynomial time nondeterministic Turing transducer N and for a string $x \in \Sigma^*$, $N(x)$ denotes the set of nonempty strings which some computation path of N outputs on input x.

Definition 2.1 [21] A set A is one word-decreasing self-reducible (1-wd self-reducible, for short) if there exists a polynomial time Turing transducer M such that for every $x \in \Sigma^*$, the following conditions are satisfied.

(1) $M(x)$ is one of the following form: either *true*, *false*, (id, y), or $(\neg, y)$, where id (respectively, $\neg$) is the identity function (respectively, negation) of one argument and $y \in \Sigma^*$ and $y < x$, and

(2) if $M(x) \in \{true, false\}$, then $x \in A$ iff $M(x) = true$ and if $M(x) = (\alpha, y)$, then $x \in A$ iff $\alpha(\chi_A(y)) = true$.

We define a restriction of 1-wd self-reducible sets. Consider the self-reducibility chain from an input x, consisting of x, the string printed by the self-reducibility machine on input x, and so on, until we arrive to a string that the machine decides directly (it writes *true* or *false*). In a strictly 1-wd self-reducible set, every string in a self-reducibility chain has two components: the first one is the predecessor (in lexicographical order) of the first component of the previous string in the chain, and the second one is an extra information with only a polynomial number of values.

Definition 2.2 A set A is strictly one word-decreasing self-reducible (strictly 1-wd self-reducible, for short) if there exists a polynomial time Turing transducer M and a polynomial p such that for every x, M witnesses that A is 1-wd self-reducible and for every $x \in \Sigma^*$,

(3) if $M(x)$ is of the form (α, y), then there exist z, z', w, and w' in Σ^* such that

(i) $x = \langle z, z' \rangle$ and $y = \langle w, w' \rangle$,

(ii) $|z| = |w|$ and $|z'| = |w'|$, and

(iii) $w = \mathrm{pred}(z)$ and $\mathrm{ord}(z'), \mathrm{ord}(w') < p(|x|)$.

For a natural number $k > 0$, *a k-truth-table* is a mapping from $\{true, false\}^k$ to $\{true, false\}$. For a k-truth-table α and k strings $y_1, \cdots, y_k$, $(\alpha, y_1, \cdots, y_k)$ is called *a k-tt condition*. For a k-tt condition $(\alpha, y_1, \cdots, y_k)$ and a set B, $(\alpha, y_1, \cdots, y_k)$ is *satisfied by* B if $\alpha(\chi_B(y_1), \cdots, \chi_B(y_k)) = true$. For a convention, a 0-truth-table is a constant boolean function of 0-argument; that is, a 0-truth-table is either *true* or *false*. Furthermore, a 0-tt condition is either $(true)$ or $(false)$, and a 0-tt condition σ is satisfied by a set B if $\sigma = (true)$.

Definition 2.3 For a natural number $k \geq 0$, a set A is polynomial time k-truth-table reducible to a set B (polynomial time k-tt reducible, for short and write $A \leq^{\mathrm{P}}_{k-\mathrm{tt}} B$) if there exists a polynomial time deterministic Turing transducer M such that for every $x \in \Sigma^*$, the following conditions are satisfied:

(1) $M(x)$ is a k-tt condition and

(2) $x \in A$ if and only if $M(x)$ is satisfied by B.

Moreover, a set A is polynomial time bounded truth-table reducible to a set B (polynomial time btt reducible, for short and write $A \leq^{\mathrm{P}}_{\mathrm{btt}} B$) if there exists some $k \in \mathbf{N}$ such that $A \leq^{\mathrm{P}}_{k-\mathrm{tt}} B$.

Definition 2.4 For a natural number $k \geq 0$, a set A is strongly nondeterministic polynomial time k-truth-table reducible to a set B (SN k-tt reducible, for short and write $A \leq^{\mathrm{SN}}_{k-\mathrm{tt}} B$) if there exists a polynomial time nondeterministic Turing transducer N such that for every $x \in \Sigma^*$, the following conditions are satisfied:

(1) $N(x)$ is not the empty set,

(2) for every string $z \in N(x)$,

(a) z is a k-tt condition and

(b) $x \in A$ if and only if z is satisfied by B.

Moreover, a set A is strongly nondeterministic polynomial time bounded truth-table reducible to a set B (SN btt reducible, for short and write $A \leq^{\mathrm{SN}}_{\mathrm{btt}} B$) if there exists some $k \in \mathbf{N}$ such that $A \leq^{\mathrm{SN}}_{k-\mathrm{tt}} B$.

Now we define the counting classes in a general setting, using the notation in [13], and the main complexity classes that we will use, in terms of the functions $\#acc_N(\cdot)$ and $\#rej_N(\cdot)$.

Definition 2.5 For a polynomial time decidable two-place predicate[1] Q on $\mathbf{N} \times \mathbf{N}$, a set L is in $\{Q\}\mathrm{P}$ if there exists a polynomial time nondeterministic Turing machine N such that for every $x \in \Sigma^*$,

$$x \in L \iff Q(\#acc_N(x), \#rej_N(x)).$$

Let $k \geq 2$ and let Q_{plus}, Q_{maj}, Q_{half}, and $Q^{(k)}_{\not\equiv 0}$ be two-place predicates such that for every $a, b \in \mathbf{N}$, $Q_{\mathrm{plus}}(a,b) = [a > 0]$, $Q_{\mathrm{maj}}(a,b) = [a \geq b]$, $Q_{\mathrm{half}}(a,b) = [a = b]$, and $Q^{(k)}_{\not\equiv 0}(a,b) = [a \not\equiv 0 \pmod{k}]$, respectively. Then, all of these predicates are polynomial time decidable and $\mathrm{NP} = \{Q_{\mathrm{plus}}\}\mathrm{P}$, $\mathrm{PP} = \{Q_{\mathrm{maj}}\}\mathrm{P}$, $\mathrm{C_{=}P} = \{Q_{\mathrm{half}}\}\mathrm{P}$, and $\mathrm{MOD}_k\mathrm{P} = \{Q^{(k)}_{\not\equiv 0}\}\mathrm{P}$ for every $k \geq 2$. Especially, $\mathrm{MOD}_2\mathrm{P}$ is denoted by $\oplus\mathrm{P}$.

The following relationships are well-known.

Proposition 2.6

1. [12,35] $\mathrm{NP} \cup \text{co-NP} \subseteq \mathrm{PP}$.
2. [26,33] $\text{co-NP} \subseteq \mathrm{C_{=}P} \subseteq \mathrm{PP}$.
3. [30,37] For any $k \geq 1$, if $\Sigma^{\mathrm{P}}_k = \Pi^{\mathrm{P}}_k$, then $\mathrm{PH} = \Sigma^{\mathrm{P}}_k = \Pi^{\mathrm{P}}_k$.
4. [33] $\mathrm{PP} \subseteq \mathrm{NP}^{\mathrm{C_{=}P}}$.

Furthermore, each counting class has 1 word-decreasing self-reducible $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete sets.

Lemma 2.7 Let Q be a polynomial time decidable two-place predicate on $\mathbf{N} \times \mathbf{N}$. Then, there exists a $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete set for the class $\{Q\}\mathrm{P}$ that is 1-wd self-reducible.

Lemma 2.8 For any $k \geq 2$, there exists a $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete set for the class $\mathrm{MOD}_k\mathrm{P}$ that is strictly 1-wd self-reducible.

Lemma 2.9 There exists a $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete set for the class NP that is strictly 1-wd self-reducible.

[1] Throughout the present paper, we assume that each two-place predicate is not trivial; that is, Q is neither constantly *true* nor constantly *false*.

3 The Main Technical Theorems

In this section, we prove that every one word-decreasing self-reducible set (respectively, strict one word-decreasing self-reducible set) which is $\leq^{\mathrm{SN}}_{\mathrm{btt}}$-reducible (respectively, $\leq^{\mathrm{P}}_{\mathrm{btt}}$-reducible) to some sparse set is already in $\mathrm{NP} \cap \text{co-NP}$ (respectively, P). The proof is inductive; that is, we will show that if an 1-wd self-reducible set is $\leq^{\mathrm{SN}}_{k+1-\mathrm{tt}}$-reducible to a sparse set S for some $k \in \mathbf{N}$, then the set is $\leq^{\mathrm{SN}}_{k-\mathrm{tt}}$-reducible to S. The same scheme applies in the case of strict 1-wd self-reducibility. Then, by using the above argument repeatedly and thereby reducing the number of queries to 0, we obtain the results.

Theorem 3.1 Let A be an 1-wd self-reducible set. If for some $k \in \mathbf{N}$, A is $\leq^{\mathrm{SN}}_{k+1-\mathrm{tt}}$-reducible to a sparse set S, then, A is $\leq^{\mathrm{SN}}_{k-\mathrm{tt}}$-reducible to S.

Proof Let A be an 1-wd self-reducible set and S be a sparse set to which A is $\leq^{\mathrm{SN}}_{k+1-\mathrm{tt}}$-reducible. There exist a polynomial time deterministic Turing transducer M and a polynomial time nondeterministic Turing transducer N which witnesses that A is 1-wd self-reducible and A is $\leq^{\mathrm{SN}}_{k+1-\mathrm{tt}}$-reducible to S, respectively. We will construct a polynomial time Turing machine N_0 which $\leq^{\mathrm{SN}}_{k-\mathrm{tt}}$ reduces A to S. Let $x \in \Sigma^*$ and $\Lambda = [(y_1, \sigma_1), \cdots, (y_m, \sigma_m)]$ be a list of pairs of a string and a $k+1$-tt condition. Λ is called *an (M,N)-chain w.r.t. x* if the following conditions are satisfied:

(c1) $x = y_1$ and for every $i, 1 \leq i < m$, $y_{i+1} < y_i$,

(c2) for every $i, 1 \leq i \leq m$, $\sigma_i \in N(y_i)$,

(c3) for every $i, 1 < i \leq m$, either

- $M(y_{i-1}) = (\alpha, y_i)$ for some $\alpha \in \{\mathrm{id}, \neg\}$ or
- $\sigma_i \in \{\sigma_1, \cdots, \sigma_{i-1}\}$, and

(c4) for every $i, 1 \leq i < m$, if $\sigma_i \in \{\sigma_1, \cdots, \sigma_{i-1}\}$, then $\sigma_{i+1} \notin \{\sigma_1, \cdots, \sigma_i\}$.

Moreover, a list $\Lambda = [(y_1, \sigma_1), \cdots, (y_m, \sigma_m)]$ is called *a full (M, N)-chain w.r.t. x* if Λ is an (M, N)-chain w.r.t. x and

(c5) $M(y_m) \in \{true, false\}$.

The following facts are easy to prove.

Fact 1 The sets

$$C = \{\langle x, \Lambda\rangle : \Lambda \text{ is an } (M,N)\text{-chain w.r.t. } x\} \text{ and}$$
$$\widehat{C} = \{\langle x, \Lambda\rangle : \Lambda \text{ is a full } (M,N)\text{-chain w.r.t. } x\}$$

are in NP.

Fact 2
Let $x \in \Sigma^*$ and $\Lambda = [(y_1, \sigma_1), \cdots, (y_m, \sigma_m)]$ be an (M, N)-chain w.r.t. x. Then, $|\langle x, \Lambda\rangle|$ is bounded above by some polynomial in $|x| + m$.

The machine N_0 performs in the following way:

{ The Description of N_0 }

(1) For a given input $x \in \Sigma^*$, N_0 nondeterministically guesses $m, 1 \leq m \leq r(|x|)$ and a list $\Lambda = [(y_1, \sigma_1), \cdots, (y_m, \sigma_m)]$ w.r.t. x, and N_0 nondeterministically checks that Λ is an (M, N)-chain and that

($\clubsuit$) if $m < r(|x|)$, then Λ is full,

where r is a polynomial defined later. If the check fails, then N_0 halts immediately. Otherwise, N_0 proceeds to the next step.

(2) If Λ is full, N_0 deterministically computes $\chi_A(x)$ from Λ and outputs *true* if $\chi_A(x) = true$ and *false* otherwise. Otherwise, N_0 proceeds to the next step.

(3) If Λ is not full (therefore, $m = r(|x|)$), then N_0 deterministically computes an h-tt condition ζ with $h \leq k$ such that $x \in A$ if and only if ζ is satisfied by S, outputs ζ, and halts.

{ End of the Description of N_0 }

First we show that for every $x \in \Sigma^*$, there exists at least one computation path which leads to step (2). Let x be an input to N_0 and let T be the set of all (M, N)-chains w.r.t. x. Note that there exists $\Lambda_0 = [(z_1, \tau_1), \cdots, (z_{m_0}, \tau_{m_0})]$ in T such that for every $\Lambda = [(z'_1, \tau'_1), \cdots, (z'_\ell, \tau'_\ell)]$ in T, it holds that $z_{m_0} \leq z'_\ell$. We claim that Λ_0 is full. This is seen as follows: Assume to the contrary that Λ_0 is not full. For some $\alpha \in \{\mathrm{id}, \neg\}$ and $w < z_{m_0}$, $M(z_{m_0})$ is (α, w). Let ξ be any tt-condition in $N(w)$. Suppose that $\xi \in \{\tau_i : 1 \leq i \leq m_0\}$ and let n_0 be the smallest $i, 1 \leq i \leq m_0$ such that $\tau_i = \xi$. Obviously, $\Lambda = [(z_1, \tau_1), \cdots, (z_{n_0}, \tau_{n_0}), (w, \xi)]$ is an (M, N)-chain w.r.t. x, and this yiedls a contradiction. On the other hand, suppose that $\xi \notin \{\tau_i : 1 \leq i \leq m_0\}$. Clearly, $\Lambda = [(z_1, \tau_1), \cdots, (z_{m_0}, \tau_{m_0}), (w, \xi)]$ is an (M, N)-chain w.r.t. x, and this yiedls a contradiction. Therefore, Λ_0 is full.

For a given polynomial r, define $m_1 = \max\{m_0, r(|x|)\}$ and $\Lambda_1 = [(z_1, \tau_1), \cdots, (z_{m_1}, \tau_{m_1})]$. Obviously, Λ_1 satisfies ($\clubsuit$). Thus, for every polynomial r, there exists at least one computation path which leads to step (2). Furthermore, from Fact 1 and 2, step (1) can be executed within polynomial time in $|x|$. Therefore, in order to establish the theorem, we only have to show that there exist two polynomial time algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ which satisfy the following conditions:

1. for a given x and a full (M, N)-chain Λ w.r.t. x, $\mathcal{A}_1$ computes $\chi_A(x)$, and
2. for a given x and an (M, N)-chain Λ of length $r(|x|)$, $\mathcal{A}_2$ computes an h-tt condition ζ with

$h \leq k$ such that $x \in A$ if and only if ζ is satisfied by S.

For two strings x and y, $\oplus_A[x,y]$ denotes the relationship between $\chi_A(x)$ and $\chi_A(y)$. More precisely, for every x and $y \in \Sigma^*$,

$$\oplus_A[x,y] = \begin{cases} \mathrm{id} & \text{if } \chi_A(x) = \chi_A(y), \\ \neg & \text{if } \chi_A(x) \mathrm{NE} \chi_A(y). \end{cases}$$

It is not hard to see that the following lemma holds.

Lemma 3.2 There exists a deterministic polynomial time algorithm $\mathcal{A}_1$ which, for a given x and a full (M,N)-chain Λ, computes $\chi_A(x)$.

Next we define the polynomial r. Since N runs in polynomial time and S is sparse, there exists a polynomial q such that for every $x \in \Sigma^*$ and for every (M,N)-chain $\Lambda = [(y_1,\sigma_1),\cdots,(y_m,\sigma_m)]$ w.r.t. x,

$$\begin{aligned} &\| \{w \in S : w \text{ appears as an argument in } \sigma_i \\ &\qquad \text{for some } i, 1 \leq i \leq m \} \\ &\leq q(|x|). \end{aligned}$$

Now define $r(n) = 4 \cdot 2^{2^{k+1}} \cdot q(n)^{k+1} + 1$.

Now our goal is to prove the following lemma.

Lemma 3.3 There exists a deterministic polynomial time algorithm $\mathcal{A}_2$ which, given $x \in \Sigma^*$ and an (M,N)-chain $\Lambda = [(y_1,\sigma_1),\cdots,(y_m,\sigma_m)]$ w.r.t. x with $m = r(|x|)$, computes an h-tt condition ζ with $h \leq k$ such that $x \in A$ if and only if ζ is satisfied by S.

Proof of Lemma 3.3 Let K denote the set of indices $\{1,\cdots,k+1\}$. For a $k+1$-tt condition $\sigma = (\beta, w_1, \cdots, w_{k+1})$, let $\beta(\sigma)$ denote β, for $\ell \in K$, let $\sigma[\ell]$ denote w_ℓ, and for a set $Q \subseteq K$, let $\sigma[Q]$ denote $\{\sigma[\ell] : \ell \in Q\}$. $\mathcal{A}_2$ performs in the following way:

{ The Description of $\mathcal{A}_2$ }

For a given $x \in \Sigma^*$ and a given (M,N)-chain $\Lambda = [(y_1,\sigma_1),\cdots,(y_m,\sigma_m)]$ w.r.t. x with $m = r(|x|)$, do the following:

(1) Find $I \subseteq \{1,\cdots,m\}$ with $\| I \geq q(|x|)+1$, $h \in \{0,\cdots,k\}$, $Q \subset K$ with $\| Q = h$, $\alpha_0 \in \{\mathrm{id}, \neg\}$, and a $k+1$-truth-table β_0 such that

(a) for every $i \in I$, $\oplus_A[x,y_i] = \alpha_0$,

(b) for every $i \in I$, $\beta(\sigma_i) = \beta_0$,

(c) for every i and $j \in I$ and for every $\ell \in Q$, $\sigma_i[\ell] = \sigma_j[\ell]$, and

(d) for every distinct i and $j \in I$, $\sigma_i[K\backslash Q] \cap \sigma_j[K\backslash Q] = \emptyset$.

(2) Compute an h-truth-table $\widehat{\beta}$ from β_0 by substituting every argument at the position $\ell \in K\backslash Q$ with *false* at the same time.

(3) Compute an h-truth-table $\widetilde{\beta}$ such that for every $b_1,\cdots,b_h \in \{true, false\}$ it holds that

$$\widetilde{\beta}(b_1,\cdots,b_h) = \alpha_0(\widehat{\beta}(b_1,\cdots,b_h)).$$

Set ζ to $(\widetilde{\beta}, w_{\ell_1},\cdots,w_{\ell_h})$, where $\ell_1,\cdots,\ell_h$ is an enumeration of all indices in Q in increasing order and for every $t, 1 \leq t \leq h$, $w_{\ell_t} = \sigma_i[\ell_t]$ for every $i \in I$.

{ End of the Description of $\mathcal{A}_2$ }

Notice that in order to execute step (1), we have only to do a brute-force search method over Q and $i_0 = \min\{i \in I\}$. That is, we only have to move Q over elements in $2^K - \{K\}$ and i_0 from 1 to m and enumerate all indices $i \neq i_0$ satisfying

(a') $\oplus_A[x,y_i] = \oplus_A[x,y_{i_0}]$,

(b') $\beta(\sigma_i) = \beta(\sigma_{i_0})$,

(c') for every $\ell \in Q$, $\sigma_i[\ell] = \sigma_{i_0}[\ell]$, and

(d') $\sigma_i[K\backslash Q] \cap \sigma_{i_0}[K\backslash Q] = \emptyset$,

and test whether the number of such indices is $\geq q(|x|)$ or not. Obviously, this search can be executed within polynomial time in $|x|$. Since step (2) and (3) can be done within polynomial time in $|x|$, $\mathcal{A}_2$ runs in polynomial time in $|x|$.

It remains to show that $\mathcal{A}_2$ works correctly. This is seen as follows. Define $I_{\rm init}$ to be the set of all $i \in \{1, \cdots, m\}$ such that $\sigma_i \notin \{\sigma_1, \cdots, \sigma_{i-1}\}$. It is not hard to see that

- $\| I_{\rm init} \geq 2 \cdot 2^{2^{k+1}} \cdot q(|x|)^{k+1} + 1$, and
- for every distinct i and j in $I_{\rm init}$, $\sigma_i {\rm NE} \sigma_j$.

For each $\alpha \in \{{\rm id}, \neg\}$, define

$$I_\alpha = \{i \in I_{\rm init} : \oplus_A[x, y_i] = \alpha\}.$$

Obviously, exactly one of $\| H_{\rm id}$ and $\| H_\neg$ is $\geq 2^{2^{k+1}} \cdot q(|x|)^{k+1} + 1$. Let α_0 be such a truth-table.

For each $k+1$-truth-table β, define

$$I(\beta) = \{i \in I_{\alpha_0} : \beta(\sigma_i) = \beta\}.$$

Since there are $2^{2^{k+1}}$ possible β, there is at least one β such that $\| I(\beta) \geq q(|x|)^{k+1} + 1$. Let β_0 be one of such $k+1$-truth-tables.

Define $G = \{\sigma_i : i \in I(\beta_0)\}$. G can be viewed as a family of ordered sets, each one with cardinality $k+1$. By a simple modification of the theorem in [10], we obtain the following proposition.

Proposition 3.4 Let $\mathcal{F}$ be a family of ordered sets, each one with cardinality t. If $\| \mathcal{F} \geq d^t + 1$, then there exist $\mathcal{G} \subset \mathcal{F}$ with $\| \mathcal{G} \geq d+1$ and a set $Q \subseteq \{1, \cdots, t\}$ such that

(i) for every $\ell \in Q$ and for every U and V in $\mathcal{G}$, $U(\ell) = V(\ell)$, and

(ii) for every distinct U and V in $\mathcal{G}$, $\{U(\ell) : \ell \in \{1, \cdots, t\} \setminus Q\} \cap \{V(\ell) : \ell \in \{1, \cdots, t\} \setminus Q\} = \emptyset$,

where $U(\ell)$ (respectively, $V(\ell)$) denotes the ℓ-th component of U (respectively, V).

From the above proposition, there exist a set $I \subseteq I(\beta_0)$ with $\| I \geq q(|x|)+1$, $h \in \{0, \cdots, k\}$ and a set $Q \subset K$ with $\| Q = h$ satisfying the conditions (c) and (d). Since $I \subseteq I(\beta_0)$, conditions (a) and (b) are satisfied. Therefore, the search procedure is always successful.

On the other hand, $\{\sigma_i[K \backslash Q] : i \in I\}$ is a family of disjoint sets in Σ^* with $\| I \geq q(|x|) + 1$. Since $q(|x|)$ is an upper bound for the number of strings in S appearing as an argument in σ_i, $1 \leq i \leq m$, there exists at least one i such that $\sigma_i[K \backslash Q] \subseteq \overline{S}$. Let ν denote one of such i. It holds that

$$\begin{aligned} & y_\nu \in A \\ \iff & \beta_0(\chi_S(\sigma_\nu[1]), \cdots, \chi_S(\sigma_\nu[k+1])) = \mathit{true} \\ \iff & \widehat{\beta}(\chi_S(\sigma_\nu[\ell_1]), \cdots, \chi_S(\sigma_\nu[\ell_h])) = \mathit{true}. \end{aligned}$$

By combining this with $\oplus_A[x, y_\nu] = \alpha_0$, we have

$$\begin{aligned} & x \in A \\ \iff & \alpha_0(\widehat{\beta}(\chi_S(\sigma_\nu[\ell_1]), \cdots, \chi_S(\sigma_\nu[\ell_h]))) = \mathit{true} \\ \iff & \widetilde{\beta}(\chi_S(\sigma_\nu[\ell_1]), \cdots, \chi_S(\sigma_\nu[\ell_h])) = \mathit{true} \\ \iff & \zeta \text{ is satisfied by } S. \end{aligned}$$

Therefore, ζ is an h-tt condition such that $x \in A$ if and only if ζ is satisfied by S. Hence $\mathcal{A}_2$ works correctly. This proves the lemma, and consequently, this proves the theorem.

Proof of Lemma 3.3
Proof of Theorem 3.1

From Theorem 3.1, we obtain the following theorem.

Theorem 3.5 If a set A is 1-wd self-reducible and $\leq^{\rm SN}_{\rm btt}$-reducible to some sparse set, then A is in NP$\cap$ co-NP.

For $\leq^{\rm P}_{\rm btt}$-reducibility of strict 1-wd self-reducible sets to sparse sets, we have the following theorems.

Theorem 3.6 Let A be a strict 1-wd self-reducible set. If for some $k \in \mathbf{N}$, A is $\leq^{\mathrm{P}}_{k+1-tt}$-reducible to a sparse set S, then A is $\leq^{\mathrm{P}}_{k-tt}$-reducible to S.

Theorem 3.7 If a set A is strictly 1-wd self-reducible and $\leq^{\mathrm{P}}_{btt}$ reducible to some sparse set, then A is in P.

4 Sparse Bounded Truth-Table Hard Sets for Counting Classes

In this section, we will consider the possibility of the existence of sparse bounded truth-table hard sets for counting classes by using the theorems we showed in the previous sections. It is not hard to prove the theorems and corllaries here so omit them. For $\leq^{\mathrm{SN}}_{btt}$-reducibility to sparse sets, we obtain the following theorem.

Theorem 4.1 Let $K = \{Q\}\mathrm{P}$ for some polynomial time decidable two-place predicate on $\mathbf{N} \times \mathbf{N}$. If there is a sparse $\leq^{\mathrm{SN}}_{btt}$-hard set for K, then $K \subseteq \mathrm{NP} \cap \text{co-NP}$.

For $\leq^{\mathrm{P}}_{btt}$-reducibility to sparse sets, from Theorem 3.7 and the existence of strictly 1-wd self-reducible sets that have been shown to be complete for different classes in section 3, we obtain the following theorems.

Theorem 4.2 Let $K = \{Q\}\mathrm{P}$ for some polynomial time decidable two-place predicate Q on $\mathbf{N} \times \mathbf{N}$ such that $\mathrm{NP} \subseteq K \cup \mathrm{co}K$. If there is a sparse $\leq^{\mathrm{P}}_{btt}$-hard set for K, then $K = \mathrm{P}$.

Theorem 4.3 For any $k \geq 2$, if $\mathrm{MOD}_k\mathrm{P}$ has a sparse $\leq^{\mathrm{P}}_{btt}$-hard set then $\mathrm{MOD}_k\mathrm{P} = \mathrm{P}$.

Next we consider the consequences of theorems 4.1, 4.2 and 4.3. From Theorem 4.1, we obtain the following corollaries first stated in [23].

Corollary 4.4 [23] If there is a sparse $\leq^{\mathrm{SN}}_{btt}$-hard set for NP, then PH = NP.

Corollary 4.5 [23] If there is a sparse $\leq^{\mathrm{SN}}_{btt}$-hard set for PP, then PH = NP = PP.

Moreover, we have a similar result for $\mathrm{C}_{=}\mathrm{P}$.

Corollary 4.6 If there is a sparse $\leq^{\mathrm{SN}}_{btt}$-hard set for $\mathrm{C}_{=}\mathrm{P}$, then $\mathrm{PH} = \mathrm{NP} = \mathrm{PP} = \mathrm{C}_{=}\mathrm{P}$.

On the other hand, from Theorem 4.2, we obtain the following corollaries.

Corollary 4.7 [24] If there is a sparse $\leq^{\mathrm{P}}_{btt}$-hard set for NP, then NP = P.

Corollary 4.8 [23] If there is a sparse $\leq^{\mathrm{P}}_{btt}$-hard set for PP, then PP = P.

Corollary 4.9 If there is a sparse $\leq^{\mathrm{P}}_{btt}$-hard set for $\mathrm{C}_{=}\mathrm{P}$, then $\mathrm{PP} = \mathrm{C}_{=}\mathrm{P} = \mathrm{P}$.

Finally, we consider $\mathrm{MOD}_k\mathrm{P}$. For a class K, a set L is in $\mathrm{BP} \cdot K$ if there exist a set A in K and a polynomial p such that for every $x \in \Sigma^*$,

$$\| \{y \in \Sigma^{=p(|x|)} : \chi_A(\langle x, y\rangle) = \chi_L(x)\} \geq \frac{1}{3} \cdot 2^{p(|x|)}.$$

The BP-operator is first introduced in [27]. Also, BPP is the class of sets for which there exists a probabilistic polynomial time acceptor with error probability $\leq 1/3$ [12]. The following relationships between BP-operator and the polynomial time hierarchy are widely known.

Theorem 4.10
1. [27] For every $k \geq 1$, $\mathrm{BP} \cdot \Sigma^{\mathrm{P}}_k \subseteq \Pi^{\mathrm{P}}_{k+1}$.
2. [32] $\mathrm{PH} \subseteq \mathrm{BP} \cdot \mathrm{C}_{=}\mathrm{P}$, $\mathrm{PH} \subseteq \mathrm{BP} \cdot \mathrm{PP}$, and $\mathrm{PH} \subseteq \mathrm{BP} \cdot \mathrm{MOD}_k\mathrm{P}$, where $k \geq 2$.
3. [27] $\mathrm{BPP} = \mathrm{BP} \cdot \mathrm{P}$.

4. [18,29] BPP $\subseteq \Sigma_2^P \cap \Pi_2^P$.

From Proposition 2.6, Theorem 4.1 and Theorem 4.10, we obtain the following corollaries.

Corollary 4.11 Let K be any class chosen from $\{MOD_2P, MOD_3P, \cdots\}$. If K has sparse $\leq_{btt}^{SN}$ hard sets, then PH $\subseteq \Sigma_2^P \cap \Pi_2^P$.

Corollary 4.12 Let K be any class chosen from $\{MOD_2P, MOD_3P, \cdots\}$. If K has sparse $\leq_{btt}^{P}$ hard sets, then PH = BPP.

Since there are complete sets for PSPACE that are 1-wd self-reducible [21], we can also derive the following result from Theorem 3.5.

Corollary 4.13 If there is a sparse $\leq_{btt}^{P}$-hard set for PSPACE, then PSPACE = NP.

Note that this is the biggest class to which we can apply our technique, since all 1-wd self-reducible sets can be decided in PSPACE, and then they cannot be complete for any class bigger than PSPACE.

Acknowledgments The authors are very grateful to Richard Beigel, Ricard Gavaldà and Jacobo Torán for their careful reading of the manuscript and many valuable comments, which made the manuscript more readable and understandable. Also, they would like to thank Masao Ikekawa, who let them know the paper [10]. The first author would like to thank Professor Kojiro Kobayashi for his valuable suggestions.

References

[1] L. Adleman and K. Manders, Reducibility, randomness, and intractability, *9th STOC* (ACM, 1977) 151-163.

[2] J. L. Balcázar, Self-reducibility, *4th STACS* (LNCS **247**, Springer-Verlag, 1987) 136-147; also, to appear in *JCSS*.

[3] J. L. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I* (EATCS monographs on Theoretical Computer Science 11, Springer-Verlag, 1988).

[4] R. Beigel, J. Gill, and U. Hertrampf, Counting classes: Thresholds, parity, mods, and fewness, *7th STACS* (LNCS **415**, Springer-Verlag, 1990) 49-57.

[5] R. Beigel, L. A. Hemachandra, and G. Wechsung, On the power of probabilistic polynomial time $P^{NP}[\log] \subseteq PP$, *4th SICT* (IEEE, 1989) 225-227.

[6] R. Beigel, N. Reingold, and D. Spielman, PP is closed under intersection, Tech. Rep. 803, Yale University, June 1990.

[7] L. Berman and J. Hartmanis, On isomorphism and density of NP and other sets, *SIAM J. Comput.* 6 (1977) 305-322.

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

[9] J. Cai and L. A. Hemachandra, On the power of parity polynomial time, *6th STACS* (LNCS 349, Springer-Verlag, 1989) 229-240.

[10] P. Erdös and R. Rado, Intersection theorems for systems of sets, *J. London Math. Soc.* 35 (1960) 85-90.

[11] S. Fortune, A note on sparse complete sets, *SIAM J. Comput.* 8 (1979) 431-433.

[12] J. Gill, Computational complexity of probabilistic Turing machines, *SIAM J. Comput.* 6 (1975) 675-695.

[13] T. Gundermann, N. A. Nasser, and G. Wechsung, A survey on counting classes, *5th SICT* (IEEE, 1990) 140-153.

[14] J. Kadin, $P^{NP[\log n]}$ and sparse Turing-complete sets for NP, *2nd SICT* (IEEE, 1987) 33-40.

[15] R. M. Karp and R. J. Lipton, Some connections between nonuniform and uniform complexity classes, *12th STOC* (ACM, 1980) 302-309.

[16] K. Ko, Distinguishing bounded reducibilities by sparse sets, *3rd SICT* (IEEE, 1988) 181-191.

[17] K. Ko, Distinguishing conjunctive and disjunctive reducibilities by sparse sets, *Inf. and Comput.* **81** (1989) 62-87.

[18] C. Lautemann, BPP and the polynomial hierarchy, *IPL* **17** (1983) 215-217.

[19] T. J. Long, On γ-reducibility versus polynomial time reducibility, *TCS* **14** (1981) 91-101.

[20] T. J. Long, Strong nondeterministic polynomial time reducibilities, *TCS* **21** (1982) 1-25.

[21] A. Lozano and J. Torán, Self-reducible sets of small density, accepted for publication in *Math. Syst. Theory.*

[22] S. R. Mahaney, Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis, *JCSS* **25** (1982) 130-143.

[23] M. Ogiwara, On sparse bounded truth-table hard sets for PP and NP, Manuscript, June 1990.

[24] M. Ogiwara and O. Watanabe, On polynomial time bounded truth-table reducibility of NP sets to sparse sets, *22nd STOC* (ACM, 1990) 457-467; also, accepted for publication in *SIAM J. Comput.*.

[25] C. Papadimitriou and S. Zachos, Two remarks on the power of counting, *6th GI Conference* (LNCS **145**, Springer-Verlag, 1983) 269-276.

[26] D. A. Russo, Structural properties of complexity classes, Ph D. dissertation, Dept. of Math., Univ. of Cal. at Santa Barbara, 1985.

[27] U. Schöning, Probabilistic complexity classes and lowness, *JCSS* **39** (1988) 84-100.

[28] J. Simon, On the difference between one and many, *4th ICALP* (LNCS **52**, Springer-Verlag, 1977) 480-491.

[29] M. Sipser, A complexity theoretic approach to randomness, *15th STOC* (ACM, 1983) 330-335.

[30] L. J. Stockmeyer, The polynomial-time hierarchy, *TCS* **3** (1977) 1-22.

[31] S. Toda, On the computational power of PP and $\oplus$P, *30th FOCS* (IEEE, 1989) 514-519.

[32] S. Toda and M. Ogiwara, Counting classes are as hard as the polynomial-time hierarchy, Tech. Rep. CSIM 90-09, Univ. of Electro-Communications, July 1990.

[33] J. Torán, An oracle characterization of the counting hierarchy, *3rd SICT* (IEEE, 1988) 213-223.

[34] E. Ukkonen, Two results on polynomial time truth-table reductions to sparse sets, *SIAM J. Comput.* **12** (1983) 580-587.

[35] K. W. Wagner, The complexity of combinatorial problems with succinct input representation, *Acta Inform.* **23** (1986) 325-356.

[36] O. Watanabe, On $\leq^P_{1\text{-}tt}$ sparseness and nondeterministic complexity classes, *15th ICALP* (LNCS **317**, Springer-Verlag, 1988) 697-709.

[37] C. Wrathall, Complete sets and the polynomial-time hierarchy, *TCS* **3** (1977) 23-33.

[38] C. K. Yap, Some consequences of non-uniform conditions on uniform classes, *TCS* **26** (1983) 287-300.

[39] Y. Yesha, On certain polynomial-time truth-table reducibilities of complete sets to sparse sets, *SIAM J. Comput.* **12** (1983) 411-425.