Technical Reports on Mathematical and Computing Sciences: TR-C143
title: On the difference between polynomial-time many-one and truth-table reducibilities on distributional problems
author: Shin Aida[1], Rainer Schuler[2], Tatsuie Tsukiji[1], and Osamu Watanabe[2]
affiliation:
1. School of Informatics and Sciences, Nagoya University, Nagoya 464-8601.
2. Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology
Meguro-ku Ookayama, Tokyo 152-8552.

email: watanabe@is.titech.ac.jp
acknowledgments to financial supports:
Supported in part by JSPS/NSF cooperative research: Complexity Theory for Strategic Goals, 1998–2001.

Abstract. In this paper we separate many-one reducibility from truth-table reducibility for distributional problems in Dist$\mathcal{NP}$ under the hypothesis that $\mathcal{P} \neq \mathcal{NP}$. As a first example we consider the 3-Satisfiability problem (3SAT) with two different distributions on 3CNF formulas. We show that 3SAT using a version of the standard distribution is truth-table reducible but not many-one reducible to 3SAT using a less redundant distribution unless $\mathcal{P} = \mathcal{NP}$.

We extend this separation result and define a distributional complexity class $\mathcal{C}$ with the following properties:
(1) $\mathcal{C}$ is a subclass of Dist$\mathcal{NP}$, this relation is proper unless $\mathcal{P} = \mathcal{NP}$.
(2) $\mathcal{C}$ contains Dist$\mathcal{P}$, but it is not contained in Ave$\mathcal{P}$ unless Dist$\mathcal{NP} \subseteq$ Ave$\mathcal{ZPP}$.
(3) $\mathcal{C}$ has a $\leq_{\mathrm{m}}^{p}$-complete set.
(4) $\mathcal{C}$ has a $\leq_{\mathrm{tt}}^{p}$-complete set that is not $\leq_{\mathrm{m}}^{p}$-complete unless $\mathcal{P} = \mathcal{NP}$.

This shows that under the assumption that $\mathcal{P} \neq \mathcal{NP}$, the two completeness notions differ on some non-trivial subclass of Dist$\mathcal{NP}$.

1 Introduction

Since the discovery of $\mathcal{NP}$-complete problems by Cook and Levin [Coo71, Lev73], a considerable number of $\mathcal{NP}$-complete problems have been reported from various areas in computer science. It is quite interesting and even surprising that most of these $\mathcal{NP}$-completeness results, except only few cases [VV83], have been proven by showing a polynomial-time *many-one* reduction from some other known $\mathcal{NP}$-complete problems. Recall that there are various reducibility types (among polynomial-time deterministic reducibilities) and that polynomial-time many-one reducibility is of the most restrictive type. For example, polynomial-time truth-table reducibility is, by definition, more general than polynomial-time many-one reducibility, and in fact, it has been shown [LLS75] that these two reducibilities differ on some problems. Nevertheless, no $\mathcal{NP}$-complete problem is known that requires (even seems to require) polynomial-time truth-table reducibility for proving its $\mathcal{NP}$-completeness.

Many researchers have studied the difference between these polynomial-time reducibility types; see, e.g., [LY90, Hom97]. Notice first that showing the difference between many-one and more stronger reducibilities on $\mathcal{NP}$ implies that $\mathcal{P} \neq \mathcal{NP}$ (because if $\mathcal{P} = \mathcal{NP}$, then any nontrivial set in $\mathcal{NP}$ is $\mathcal{NP}$-complete under many-one reducibility). Thus, it is more reasonable to assume (at least) $\mathcal{P} \neq \mathcal{NP}$ and to ask about the difference between, e.g., many-one and truth-table reducibilities on $\mathcal{NP}$ under this assumption. Unfortunately, however, the question is still open even assuming that $\mathcal{P} \neq \mathcal{NP}$. Maybe the difference is too subtle to see it in $\mathcal{NP}$ by only assuming $\mathcal{P} \neq \mathcal{NP}$. In this paper we show that this subtle difference appears when we use reducibility for analyzing distributional $\mathcal{NP}$ problems.

The notion of "distributional problem" has been introduced by Levin [Lev86] in his framework for studying average-case complexity of $\mathcal{NP}$ problems. A distributional problem is a pair (A, μ) of a decision problem A (as usual, A is a set of positive instances of the problem) and an input distribution μ. Intuitively (see below for the formal definition), by the complexity of (A, μ), we mean the complexity of A when inputs are given under the distribution μ. Analog to the class $\mathcal{NP}$, Levin proposed to study a class Dist$\mathcal{NP}$, the class of all distributional problems (A, μ) such that $A \in \mathcal{NP}$ and μ can be computed in polynomial-time. Also he introduced a class Ave$\mathcal{P}$, the class of distributional problems solvable in polynomial-time on average. Then the question analog to the $\mathcal{P}$ versus $\mathcal{NP}$ question is whether Dist$\mathcal{NP} \subseteq$ Ave$\mathcal{P}$. Levin also extended the notion of reducibility for distributional problems, and somewhat surprisingly, he proved that distributional problem $(\mathrm{BH}, \mu_{\mathrm{st}})$, where BH is a canonical $\mathcal{NP}$-complete set and μ_{st} is a standard uniform distribution, is complete in Dist$\mathcal{NP}$ by using many-one reducibility. (See, e.g., [Gur91, Wang97] for detail explanation and basic results on Levin's average-case complexity theory.)

Unlike the worst-case complexity, only a small number of "natural" distributional problems have been shown as complete for Dist$\mathcal{NP}$. Intuitively, it seems that most $\mathcal{NP}$ problems are not hard enough to become complete under natural distributions. More technically, the condition required for the reducibility (in the average-case framework) is strong, it is affected by even some small change of distribution. Aida and Tsukiji [AT00] pointed out that this sensitivity could be used to show the subtle difference between many-one and more general reducibilities. They showed two problems (A, μ_A) and (B, μ_B) in Dist$\mathcal{NP}$ such that $(A, \mu_A) \leq^{p}_{\mathrm{tt}} (B, \mu_B)$ but $(A, \mu_A) \not\leq^{p}_{\mathrm{m}} (B, \mu_B)$ unless $\mathcal{P} = \mathcal{NP}$. Unfortunately, though, these distributions μ_A and μ_B are so small that these two problems are trivially in Ave$\mathcal{P}$. It has been left open to show such difference on nontrivial distributional $\mathcal{NP}$ problems.

We solve this open question in this paper. We separate many-one reducibility from truth-table reducibility for nontrivial problems in Dist$\mathcal{NP}$ under the hypothesis that $\mathcal{P} \neq \mathcal{NP}$. Furthermore, we show some nontrivial subclass of Dist$\mathcal{NP}$ in which many-one and truth-table completeness notions differ unless $\mathcal{P} = \mathcal{NP}$.

First we define two versions of the distributional 3-Satisfiability problem (3SAT) by considering different distributions on 3CNF formulas. The first distribution μ is defined by modifying a standard uniform distribution on 3CNF formulas. Here the standard distribution gives each formula the probability that it is generated by a random process, where every literal is chosen randomly from the set of variables and their complements. For the second distribution ν, we consider less redundant 3CNF representation. Note that a 3CNF formula F usually has many equivalent formulas; for example, permuting the order of clauses in F, we can easily get a different but equivalent formula. We consider some restriction on the form of formulas to reduce this redundancy, and define the second distribution ν so that non-zero probability is given only on such formulas that satisfy our restriction. By this way, the probability of each formula (of the required form) gets increased considerably (compared with ν). By using this increase, we prove that $(3\mathrm{SAT}, \nu)$ is not many-one reducible to $(3\mathrm{SAT}, \mu)$ unless $\mathcal{P} = \mathcal{NP}$. On the other hand, by using the self-reducibility of 3SAT, we prove that even $(3\mathrm{SAT}, \nu)$ is truth-table reducible $(3\mathrm{SAT}, \mu)$.

Next we extend this separation technique and define a subclass $\mathcal{C}$ of Dist$\mathcal{NP}$ in which many-one and truth-table completeness notions differ unless $\mathcal{P} = \mathcal{NP}$. Furthermore, we can show that $\mathcal{C}$ is not contained in Ave$\mathcal{P}$ (thus it is not trivial) unless all Dist$\mathcal{NP}$ are solvable in polynomial-time on average by randomized zero-error computation.

2 Preliminaries

We use standard notations and definitions from computability theory, see, e.g., [BDG88]. We briefly recall the definitions of the average-case complexity classes used in the following. For definitions and discussion, see [Gur91].

A distributional problem consists of a set L and a distribution on strings defined by the distribution function μ, i.e., a (real) valued function such that $\sum_x \mu(x) = 1$. A distribution μ is called *polynomial-time computable* if the binary expansion of the distribution function μ^*, defined by $\mu^*(x) = \sum_{y \le x} \mu(x)$ for all x, is polynomial-time computable in the sense that for any x and n, the first n bits of $\mu^*(x)$ is computable within polynomial time w.r.t. $|x|$ and n.

Let Dist$\mathcal{NP}$ denote the class of all distributional problems (L, μ) such that $L \in \mathcal{NP}$ and μ is polynomial-time computable. Similarly, Dist$\mathcal{P}$ denotes the class of distributional problems $(L, \mu) \in$ Dist$\mathcal{NP}$ such that L is in $\mathcal{P}$.

The average-case analog of $\mathcal{P}$ is denoted by Ave$\mathcal{P}$ and defined as follows. A distributional problem (L, μ) is decidable in polynomial-time on average, if L is decidable by some t-time bounded Turning machine, and t is *polynomial on μ-average*, which means that t, a function from $\Sigma^* \to \mathbf{N}$, satisfies the following for some constant $\epsilon > 0$ [Lev86, Gur91].

$$\sum_x \frac{t^\epsilon(x)}{|x|}\mu(x) \;<\; \infty.$$

Let Ave$\mathcal{P}$ denote the class of all distributional problems that are decidable in polynomial-time on average. Similarly let Ave$\mathcal{ZPP}$ denote the class of all distributional problems that are decidable in polynomial-time on average by randomized Turing machines (without error), see, e.g., [Imp95]. Here we have to be a little careful defining average polynomial-time for randomized computation [Gur91]. Let $t(x, r)$ denote the running time of M on input x using random bits r. We say that M is *polynomial-time* on average if

$$\sum_x \sum_r 2^{-|r|} \frac{t^\epsilon(x,r)}{|x|} \mu(x) \;<\; \infty,$$

where r ranges over all binary strings such that M on input x halts using r but it does not halt using any prefix r' of r.

Finally, we define "reducibility" between distributional problems. A distributional problem (A, μ) is *polynomial-time reducible* to (B, ν), if there exists an oracle Turing machine M and a polynomial p such that the following three conditions hold.

(1) The running time of M (with oracle B) is polynomially bounded.

(2) For every x, we have $x \in A \Leftrightarrow x \in L(M, B)$, where $L(M, B)$ is the set of strings accepted by M with oracle B.

(3) For any x, let $Q(M, B, x)$ denote the set of oracle queries made by M with oracle B and input x. The following condition holds for every y.

$$\nu(y) \;\geq \sum_{x\,:\,y \in Q(M,B,x)} \frac{\mu(x)}{p(|x|)}.$$

From these three conditions, any problem (A, μ) that is polynomial-time reducible to some problem in Ave$\mathcal{P}$ also belongs to Ave$\mathcal{P}$ [Lev86, Gur91]. The above condition (3) is called a *dominance condition*.

By restricting the type of queries, we can define finer reducibilities. A reduction M is called a *truth-table reduction* if for every x, the oracle queries of M on input x are made nonadaptively, i.e., they are independent of the oracle set. M is a *many-one reduction* if for every x, M on input x makes exactly one query, and it accepts x iff the query is in the oracle set. We can define more general reduction types by considering randomized computation. That is, a reduction is called a *randomized reduction* if the oracle Turing machine is randomized. In this paper, we consider the most restrictive randomized reduction type that requires "zero error" to the oracle Turing machine M, i.e., M is correct and polynomial-time bounded for all inputs and all possible random bits. The dominancy condition needs to be revised for randomized reductions. For any x and any r, let $Q(M, B, x, r)$ denote the set of oracle queries made by $M^B(x; r)$, i.e., the execution of M with oracle B on input x using random bits r. Here we assume that $M^B(x; r)$ halts consuming all bits of r and $M^B(x; r')$ does not halt for any prefix r' of r. (If r does not satisfy this condition, then we simply define $Q(M, B, x, r)$ to be empty.) Then our dominancy condition is stated as follows.

(3) For every y, we have

$$\nu(y) \;\geq \sum_{x,r\,:\,y \in Q(M,B,x,r)} \frac{\mu(x) \cdot 2^{-|r|}}{p(|x|)}.$$

3 Separation on 3SAT

Our first separation is on 3SAT, i.e., the set of all *satisfiable* 3CNF formulas F. We recall some basic definitions on 3SAT. A formula F is in *3CNF* if F is a conjunction of *clauses* which contain at most 3 *literals*, i.e., F is of the form $C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_i = l_{j_1} \vee l_{j_2} \vee l_{j_3}$ and l_{j_k} is either the variable v_{j_k} or its negation. (We use the index of j_k of each literal l_{j_k} to denote that of its variable.) We use $\mathcal{F}^{(n,m)}$ to denote the set of 3CNF formulas with n variables and m clauses. (We assume that $m \leq 8n^3$.)

The standard distribution μ_{st} assigns to any formula F in $\mathcal{F}^{(n,m)}$ the probability

$$\frac{1}{n(n+1)} \cdot \frac{1}{8n^3} 2^{-3m(1+\lceil \log n \rceil)}.$$

That is, we have the following random experiment in mind.

> Choose n (number of variables) randomly. Choose $m \in \{1, \cdots, 8n^3\}$ (number of clauses) randomly. Choose each of the $3m$ literals l randomly from the set of variables and negated variables of size $2n$. Let F denote the resulting formula. Output F.

In order to simplify our discussion, we restrict the form of formulas so that $m = f_0(n)$, where $f_0(n) = \lceil n \log n \rceil$. Since m is determined from n, the standard distribution is modified as follows.

$$\mu_{\mathrm{st0}}(F) \;=\; \begin{cases} 8n^3 \cdot \mu_{\mathrm{st}}(F), & \text{if } F \in \mathcal{F}^{(n,f_0(n))}, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

We should note here that the same result holds by considering any "smooth" function for f such that $n \leq f(n) \leq n \log n$ for all n. Here a function f is *smooth* if there is no big jump from $f(n-1)$ to $f(n)$; more precisely, there exists constants $c_f > 1$ and $d_f > 0$ such that for any sufficiently large n and for some $k < d_f \log n$, we have $f(n) - c_f \log n < f(n-k) < f(n) - \log n$. For example, consider $f(n) = n\lceil \log n \rceil$. While this function satisfies our smoothness condition for most n, we have $f(n) \geq f(n-k) + \log n$ for any $k = O(\log n)$ if n is sufficiently large and $\lceil \log n \rceil = 1 + \lceil \log(n-1) \rceil$. On the other hand, a function like $f(n) = \lceil n \log n \rceil$ satisfies this smoothness condition.

Note that it is still open whether $(\mathrm{3SAT}, \mu)$ is in Ave$\mathcal{P}$, i.e., polynomial-time solvable on average. On the other hand, it has been shown that $(\mathrm{3SAT}, \mu)$ defined using $f(n) \geq dn^2$ for some $d > 0$ is indeed in Ave$\mathcal{P}$ [KP92].

Now define the first distribution. From some technical reason, we consider 3CNF formulas with some additional clauses. For any n, a 3CNF $P = C_1 \wedge \cdots C_{2\lceil \log(n+1) \rceil - 1}$ is called a *type-I prefix for n* if each C_i is of the form $C_i = (v_{j_i} \vee v_{j_i} \vee v_{j_i})$ for some $j_i \in \{3(i+1), 3(i+1)+1, 3(i+1)+3\}$. Note that there are $3^{2\lceil \log(n+1) \rceil - 1} \leq n^4$ type-I prefices for n. We consider only formulas G that is of the form $P \wedge F$ for some type-I prefix P for n and $F \in \mathcal{F}^{(n,f_0(n))}$. We use $\mathcal{G}^{(n)}$ to denote the set of such formulas. This somewhat artificial requirement is just to simplify our analysis of a truth-table reduction defined in Lemma 3.

Our first distribution is defined as follows.

$$\mu(G) \;=\; \begin{cases} \frac{1}{n(n+1)} 3^{-(2\lceil \log(n+1)\rceil -1)} \cdot 2^{-3f_0(n)(1+\log n)}, & \text{if } G \text{ is in } \mathcal{G}^{(n)}, \text{ and} \\ , & \text{otherwise.} \end{cases}$$

Next we define the second distribution. As mentioned in Introduction, the 3CNF representation has redundancy; i.e., a 3CNF formula (usually) has many equivalent formulas. Here we introduce one restriction on the form of formulas for reducing some redundancy, which is not essential for the hardness of the satisfiability problem.

For any n, a 3CNF $P = C_1 \wedge \cdots C_{2\lceil \log n\rceil}$ is called the *type-II prefix for n* if each C_i is of the form $C_i = (v_{3i} \vee v_{3i+1} \vee v_{3i+3})$. Note that for each n, the type-II prefix for n is uniquely determined. We consider only formulas F in $\mathcal{F}^{(n,f_0(n))}$ such that (i) F is *ordered*, i.e., (a) every clause C of F is of the form[1] $(l_{j_1} \vee l_{j_2} \vee l_{j_3})$ with $j_1 \le j_2 \le j_3$, and (b) clauses are placed in the (lexicographic) order of the indices of variables in each clause, and (ii) the first $2\lceil \log n\rceil$ clauses of F are the type-II prefix for n. Let $\mathcal{F}^{(n)}$ denote the set of such formulas. Note that $\mathcal{F}^{(n)}$ is a subset of $\mathcal{F}^{(n,f_0(n))}$, (*Cf.* $\mathcal{G}^{(n)}$ is an extension of $\mathcal{F}^{(n,f_0(n))}$.) As shown below, this restriction is not essential.

Lemma 1. For any 3CNF formula $F \in \mathcal{F}^{(n,f_0(n))}$, we can either convert it to an equivalent formula $F' \in \mathcal{F}^{(n)}$ by renaming variables and/or changing their signs, or determine the satisfiability of F in polynomial-time.

Proof. Let F be any 3CNF formula in $\mathcal{F}^{(n)}$. From this F, we construct the type-II prefex for n, i.e., the clauses C_1, C_2, ..., $C_{2\lceil \log n\rceil}$ of the specific form. They are constructed one by one by choosing some appropriate clause from (the remaining part of) F and renaming it. Assume that we have already constructed clauses C_1, ..., C_i, and let E be the remaining formula. So far we have used variables $v_1, ..., v_{3i}$. For obtaining the next C_{i+1}, we find from E some clause C consisting of three new variables $v_{j_1}, v_{j_2}, v_{j_3}$. Then we simply rename these variables to $v_{3i+1}, v_{3i+2}, v_{3i+3}$ and change their sings (if necessary) in E so that C is now represented as a clause $(v_{3i+1} \vee v_{3i+2} \vee v_{3i+3})$. If we can keep finding appropriate clauses, we would be able to construct required C_1, ..., $C_{2\lceil \log n\rceil}$. Finally, change the order of literals in each clause of E and the order of clauses so that E becomes ordered. Then F' is obtained as $C_1 \wedge \cdots \wedge C_{2\lceil \log n\rceil} \wedge E$.

Our construction fails if we cannot find any new appropriate clause from the remaining part E after constructing $C_1, ..., C_i$. But this means that after determining values of variables $v_1, ..., v_{3i}$ in $C_1, ..., C_i$, E becomes a 2CNF formula. Hence, we can determine its satisfiability in polynomial-time. Thus, by trying all possible assignments for $v_1, ..., v_{3i}$, we can determine whether F is satisfiable or not in polynomial-time. $\square$

Now our distribution is defined as follows.

[1] Recall that the index of each literal is used to denote that of its variable. That is, a literal l_{j_k} is either the variable v_{j_k} or its negation.

$$\nu(F) = \begin{cases} \frac{1}{n(n+1)} 2^{-3(f_0(n)-\lceil \log n \rceil)(1+\log n)}, & \text{if } F \in \mathcal{F}^{(n)}, \\ p_0, & \text{if } F = F_0, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

Here let F_0 is any fixed formula not in $\mathcal{F}^{(n)}$. and define $p_0 = 1 - \sum_{F \neq F_0} \nu(F)$. From some technical reason, for any formula $F \in \mathcal{F}^{(n)}$, $\nu(F)$ is defined $1/\lceil \log n \rceil$ smaller than the probability of choosing F uniformly at random from all formulas in $\mathcal{F}^{(n)}$. We will use this point for showing $\leq^p_{\rm tt}$-reducibility from $(\text{3SAT}, \nu)$ to $(\text{3SAT}, \mu)$. Intuitively, nu corresponds to the following random generation.

> Choose n (number of variables) randomly. Let $m = f_0(n)$ (number of clauses). Fix first $\lceil \log n \rceil$ clauses as required for the type-II prefix. Then choose the remaining $3m - 2\lceil \log n \rceil$ clauses as in the standard distribution. If F does not belong to $\mathcal{F}^{(n)}$, then just output F_0. Otherwise output F.

We observe (without proof) that the distributions μ and ν defined above are polynomial time computable. Thus, both distributional problems $(\text{3SAT}, \mu)$ and $(\text{3SAT}, \nu)$ belong to Dist$\mathcal{NP}$.

For our separation result, we first show that $(\text{3SAT}, \nu)$ is not $\leq^p_{\rm m}$ to $(\text{3SAT}, \mu)$ unless $\mathcal{P} = \mathcal{NP}$.

Lemma 2. If $(\text{3SAT}, \nu) \leq^p_{\rm m} (\text{3SAT}, \mu)$, then we have $\text{3SAT} \in \mathcal{P}$ and hence $\mathcal{P} = \mathcal{NP}$.

Proof. Assume there exists a many-one reduction R from $(\text{3SAT}, \nu)$ to $(\text{3SAT}, \mu)$. Then we can define the the following 3SAT solver. The correctness is clear by the definition of the many-one reducibility. The polynomial-time bound of this algorithm is guaranteed as follows. The reduction R reduces (in each iteration) a formula of F in $\mathcal{F}^{(n,f_0(n))}$ to a formula F' in $\mathcal{F}^{n',f_0(n')}$ with $n' \leq n-1$. That is, the number of variables reduces at least one at each while-iteration.

Algorithm 3SAT Solver
input F in $\mathcal{F}^{(n,f_0(n))}$
$F' \leftarrow F$; $n' \leftarrow n$;
while $n' > \log n$ **do**
  modify F' to an equivalent formula F in $\mathcal{F}^{(n')}$;
  % The procedure given in Lemma 1 is used.
  % If the procedure fails, then the satisfiability of F' can be determined directly.
  $G \leftarrow R(F)$; $n' \leftarrow$ the number of variables in G;
  % G is in $\mathcal{G}^{(n')}$; i.e., $G = P \wedge F'$ with some type-I prefix P for n' and $F' \in {}^{(n',f_0(n'))}$.
  remove each clause $(v_{k_i} \vee v_{k_i} \vee v_{k_i})$ of P by assinging $v_{k_i} = 1$ in F';
  % F' may be reduced to a simpler formula.
  (if necessary) add redundant variables or clauses so that F' belongs to $\mathcal{F}^{n',f_0(n')}$;
end-while

output 1 if the final F' is satisfiable, and output 0 otherwise;
end-algorithm.

This claim is proved by using the domination condition. From the domination condition, for some constant $c > 0$ and for any sufficiently large n, we have

$$\begin{aligned} &\frac{1}{n(n+1)} 2^{-3(f_0(n)-\lceil \log n \rceil)(1+\log n)} \\ &\quad = \nu(F) \ \leq \ n^c \cdot \mu(F') \\ &\qquad = \frac{n^c}{n'(n'+1)} 3^{-(2\lceil \log n'+1 \rceil -1)} \cdot 2^{-3f_0(n')(1+\log n')}, \end{aligned}$$

which implies

$$\frac{1}{n(n+1)} 2^{-3(f_0(n)-\lceil \log n \rceil)(1+\log n)} \ \leq \ \frac{n^c}{(n')^5(n'+1)} 2^{-3f_0(n')(1+\log n')}.$$

Now suppose that $n' \geq n$. Then from the above, it should hold that $c \log n > 3 \log^2 n$, which is impossible for sufficiently large n. Therefore, we have $n' < n$. $\square$

On the other hand, some $\leq_{\rm tt}^p$-reduction exists from $(\mathrm{3SAT}, \nu)$ to $(\mathrm{3SAT}, \mu)$.

Lemma 3. $(\mathrm{3SAT}, \nu) \ \leq_{\rm tt}^p \ (\mathrm{3SAT}, \mu)$.

Proof. We define a truth-table reduction from $(\mathrm{3SAT}, \nu)$ to $\mathrm{3SAT}, \mu)$. For our discussion, consider any formula F in $\mathcal{F}^{(n)}$. Recall that $F = C_1 \wedge \cdots \wedge C_{2\lceil \log n \rceil} \wedge E$, where (i) E is an ordered formula with $f_0(n) - 2\lceil \log n \rceil$ clauses, and (ii) each C_i, $1 \leq i \leq 2\lceil \log n \rceil$, is the form $(v_{3i} \vee v_{3i+1} \vee v_{3i+3})$. We would like to solve the satisfiability of F by asking polynomially many nonadaptive queries to 3SAT. Note that queried formulas are of some appropriate form, more precisely, they should belong to $\mathcal{G}^{(n')}$ for some n'. Furthermore, since $\nu(F)$ (for $F \in \mathcal{F}^{(n)}$) is much bigger than $\mu(G)$ (for $G \in \mathcal{G}^{(n)}$), we need to reduce the size of queried formulas. Our idea is simple. We delete the first $2\lceil \log n \rceil$ clauses $C_1, ..., C_{2\lceil \log n \rceil}$ by considering all possible partial assignments satisfying all these clauses. Since each C_i is $(v_{3i} \vee v_{3i+1} \vee v_{3i+3})$, we only have to assign 1 to one of three variables $v_{3i}, v_{3i+1}, v_{3i+3}$ for satisfying C_i. That is, for every partial assingment, which assings 1 to one of three variables $v_{3i}, v_{3i+1}, v_{3i+3}$ for each i, $1 \leq i \leq 2\lceil \log n \rceil$, we can remove the first $2\lceil \log n \rceil$ clauses (and $2\lceil \log n \rceil$ variables). Thus, the resulting formula has at most $f_0(n) - 2\lceil \log n \rceil \leq f_0(n-1)$ clauses and less than $n-1$ variables; hence, by adding some redundant variables and clauses (if necessary), we can modify it to a formula F' in $\mathcal{F}^{(n-1, f_0(n-1))}$. Note that there are $3^{2\lceil \log n \rceil} \leq n^4$ such partial assignments and that F is satisfiable if and only if one of the obtained F' is satisfiable. Therefore, the above procedure is indeed a disjunctive truth-table reduction that asks polynomial number of formulas of smaller size.

Though the idea is simple, a bit more complicated argument is necessary for the formal proof. For showing the dominancy condition, it would be convenient if any query formula is asked from only polynomially many formulas. Let us see how to construct query formulas satisfying this requirement from $F = C_1 \wedge \cdots \wedge C_{2\lceil \log n \rceil} \wedge E$. For each i, $1 \leq i \leq 2\lceil \log n \rceil$, let k_i

be a number in $\{3i+1, 3i+2, 3i+3\}$ denoting the index of the variable to which we assign 1 for deleting C_i. For each choice of $(k_1, ...k_{2\lceil \log n \rceil})$, we compute a query formula F' as follows.

(1) Delete all clauses $C_1, ..., C_{2\lceil \log n \rceil}$ from F (with no change in the E part). Instead add clauses $C'_i = (v_{k_i} \vee v_{k_i} \vee v_{k_i})$. That is, at this stage F is transformed to $G' = C'_1 \wedge \cdots \wedge C'_{2\lceil \log n \rceil} \wedge E$.

(2) We want to remove v_{k_1} in order to reduce (at least) one variable. Though we can remove v_{k_1} by simply replacing all v_{k_1} with v_{k_2} (or, v_{k_3}, etc.), we do this more carefully for achieving our requirement.

   (a) Move all clauses containing v_{k_1} in E in front of E.

   (b) Let C be any clause containing v_{k_1}. If v_{k_2} does not occur in C, then replace v_{k_1} in C with v_{k_2}. If v_{k_2} occurs in C but v_{k_3} does not, then replace v_{k_1} in C with v_{k_3}. Otherwise (i.e., both v_{k_2} and v_{k_3} occur in C), then replace v_{k_1} in C with v_{k_4}. (We may assume that $2\lceil \log n \rceil \geq 4$.) Let E' be the obtained formula by this replacement. Note that we do not change the order of literals.

   (c) Now v_{k_1} is not used in E'; hence, we can change all occurences of v_n with v_{k_1}, thereby reducing the number of variables in E' to $n-1$. Let E'' be the obtained formula. Then add sufficient number of dummy clauses, e.g., $(v_1 \vee v_1 \vee \neg v_1)$, to the end of E'' until it has $f_0(n-1)$ clauses. Let F' be the resulting formula.

(3) Finally, we define $G = C'_2 \wedge \cdots \wedge C'_{2\lceil \log n \rceil} \wedge F'$, which is the query formula corresponding to the partial assingment specified by $(k_1, ...k_{2\lceil \log n \rceil})$.

For the obtained formula G, it is clear that G belongs to $\mathcal{F}^{(n-1)}$ and that G is satisfiable if and only if F with the partial assignment $v_{k_1} = v_{k_2} = \cdots = v_{k_{2\lceil \log n \rceil}} = 1$ is satisfiable. Now consider the dominancy condition. Observe first that $\nu(F) \leq n^c \cdot \mu(G)$ for some $c > 0$. This is immediate by comparing $\nu(F)$ and $\mu(G)$ as follows.

$$
\begin{aligned}
\nu(F) &= \frac{1}{n(n+1)} 2^{-3(f_0(n) - \lceil \log n \rceil)(1+\log n)} \\
&= \frac{1}{n(n+1)} 2^{-3(\lceil n \log n \rceil - \lceil \log n \rceil)(1+\log n)} \leq \frac{1}{n(n+1)} 2^{-3(n \log n - \log n - 1)(1+\log n)}, \quad \text{and} \\
\mu(G) &= \frac{1}{(n-1)n} 3^{-(2\lceil \log n \rceil - 1)} \cdot 2^{-3 f_0(n-1)(1+\log(n-1))} \\
&\geq \frac{1}{(n-1)n^5} 2^{-3(\lceil (n-1)\log(n-1) \rceil)(1+\log n)} \geq \frac{1}{(n-1)n^5} 2^{-3(n \log n - \log n + 2)(1+\log n)}.
\end{aligned}
$$

Then for proving the dominancy condition, it suffices to show that each G is queried from at most polynomially many formulas. To see this, we check the above procedure backwards from a query formula G and estimate the number of possible ways to go back to the origial formula F.

Recall that G is of the form $C'_2 \wedge \cdots \wedge C'_{2\lceil \log n \rceil} \wedge F'$, where $F' \in \mathcal{F}^{(n-1, f_0(n-1))}$. (Note that n itself is uniquely determined from the number of clauses in G.) From F', we can recover E'' uniquely by removing dummy clauses. (Recall that the number of clauses in E'' is $m' = f_0(n) - 2\lceil \log n \rceil$.) At this stage, we cannot determine which v_{k_1} of $\{v_1, v_2, v_3\}$ are replaced with v_n when transforming E' to E''. That is, there are three choices. But once we fix k_1, it is clear that E' is uniquely determined from E''. To get E from E', we first determine which clauses in

E' were modified when transforming E to E'. Note that all modified clauses are placed in front of E; hence, there are at most $m' = O(n \log n)$ choices. Now once those modified clauses are determined, the original E can be uniquely computed from E'. For each modified clause C' in E', the following two cases are possible:

(Case 1) C' has v_{k_2} together with v_{k_3} and/or v_{k_4}.
If C' is ordered, then v_{k_2} is the one replaced with v_{k_1}. Othewise, the one (which must be v_{k_2} or v_{k_3}, but not both) that preceeds v_{k_2} in C is the one replaced with v_{k_1}. Thus, relace all occurences of such variable back to v_{k_1}.

(Case 2) C' has no v_{k_3} nor v_{k_4}, but it has v_{k_2}.
v_{k_2} is the one replaced with v_{k_1}. Thus, relace all occurences of v_{k_2} back to v_{k_1}.

Finally, permuting clauses for satisfying the order, we obtain E. Then the original F is obtained by just adding the type-II prefix for n in front of E.

From this argument, it is now clear that for each query formula $G \in \mathcal{G}^{(n)}$, the number of formulas in $\mathcal{F}^{(n)}$ from which G is queried is polynomially bounded. $\square$

From above two lemmas, we have the following separation result.

Theorem 4. There exist polynomial time distributions ν and μ such that $(\mathrm{3SAT}, \nu) \leq^p_{\mathrm{tt}} (\mathrm{3SAT}, \mu)$, but $(\mathrm{3SAT}, \nu) \not\leq^p_{\mathrm{m}} (\mathrm{3SAT}, \mu)$ unless $\mathcal{P} = \mathcal{NP}$.

4 Separating Completeness Notions

In this section we define some subclass of Dist$\mathcal{NP}$ in which we can show the difference between many-one and truth-table completeness notions. More specifically, we will define a distributional complexity class $\mathcal{C}$ with the following properties:

(1) $\mathcal{C}$ is a subclass of Dist$\mathcal{NP}$, and furthermore, the relation is proper unless $\mathcal{P} = \mathcal{NP}$.
(2) $\mathcal{C}$ contains Dist$\mathcal{P}$, but it is not contained in Ave$\mathcal{P}$ unless Dist$\mathcal{NP} \subseteq$ Ave$\mathcal{ZPP}$.
(3) $\mathcal{C}$ has a $\leq^p_{\mathrm{m}}$-complete set.
(4) There exists a problem $C \in \mathcal{C}$ that is $\leq^p_{\mathrm{tt}}$-complete in $\mathcal{C}$ but that is not $\leq^p_{\mathrm{m}}$-complete in $\mathcal{C}$ unless $\mathcal{P} = \mathcal{NP}$.

That is, if $\mathcal{P} \neq \mathcal{NP}$, then two completeness notions differ on some subclass of Dist$\mathcal{NP}$.

First we define the complexity class $\mathcal{C}$. For this purpose, we consider the following version of bounded halting problem, which we call *B*ounded *H*alting problem with *P*adding. Here for some technical reason, we consider only Turing machines M using one tape as both an input and a work tape. We also assume that M's tape alphabet is $\{0, 1, \mathrm{B}\}$ and that M cannot go beyond the cells containing 0 or 1. Note that this is not an essential restriction if we assume that M's reachable tape cells are initially filled by 0. On the other hand, with this assumption, we can represent the content of the whole tape of M by a string in $\{0, 1\}^*$ of fixed length.

Below we use ϕ to denote any fixed function on $\mathbf{N}$ such that $n \leq \phi(n) \leq p(n)$ for some polynomial p and $\phi(n)$ is computable within polynomial-time in n.

$$
\begin{array}{rcll}
\mathrm{BHP}_\phi & = & \{ & \langle M, q, i, w, y \rangle \ : \\
& & & \text{(i)} \quad M \text{ is NDTM, } q \text{ is its state, } i,\ 1 \leq i \leq |w|, \text{ is a head position, and} \\
& & & \qquad w, y \in \{0,1\}^*, \text{ where } w \text{ is } M\text{'s tape and } y \text{ is padding,} \\
& & & \text{(ii)} \quad |y| = \phi(|M| + |w| + t) \text{ for some } t \in \mathbf{N}, \text{ and} \\
& & & \text{(iii)} \quad M \text{ has an accepting path of length } t \text{ from configuration } (q, i, w).\ \}
\end{array}
$$

Notice here that w represents the content of the whole M's tape. We assume that M's tape head does not go outside of w. We assume some reasonable encoding of M and its state q, and $|M|$ and $|q|$ are the length of the descriptions of M and q under this encoding. Again for simplifying our discussion below, we assume that for each M and w, the length of $|q|$ and $|i|$ is fixed.

In the literature, the following versions of the halting problem BH and its padded version BH$'$ have been studied [Gur91]. Our BHP_ϕ is regarded a variation of of BH$'$ when ϕ is defined as $\phi(n) = n$.

$$
\begin{array}{rcl}
\mathrm{BH} & = & \{ \langle M, x, 0^t \rangle \ : \ M \text{ accepts } x \text{ in } t \text{ steps. } \}, \quad \text{and} \\
\mathrm{BH}' & = & \{ \langle M, x, y \rangle \ : \ M \text{ accepts } x \text{ in } |y| \text{ steps. } \}.
\end{array}
$$

As a distribution we consider the standard distribution extended on tuples, e.g., every instance $\langle M, q, i, x, y \rangle$ of BHP_ϕ has the following probability.

$$
\mu_{\mathrm{st}}(\langle M, q, i, w, y \rangle) \ = \ \frac{1}{\alpha(|M|, |q|, |i|, |w|, |y|)} \cdot 2^{-(|M|+|q|+|i|+|w|+|y|)},
$$

where $\alpha(n_1, n_2, \ldots, n_k) = \prod_{i=1}^{k} n_i(n_i + 1)$. Note however that a unary padding string has probability inverse polynomial to its length; for example, for any instance $\langle M, x, 0^t \rangle$ for BH, we have

$$
\mu_{\mathrm{st}}(\langle M, x, 0^t \rangle) \ = \ \frac{1}{\alpha(|M|, |x|, t)} \cdot 2^{-(|M|+|x|)},
$$

First it should be mentioned that $(\mathrm{BH}, \mu_{\mathrm{st}})$ is reducible to $(\mathrm{BHP}_\phi, \mu_{\mathrm{st}})$ via a randomized reduction of the strongest type, i.e., the one with no error.

Proposition 5. For any polynomially bounded $\phi(n)$ that is polynomial-time computable w.r.t. n, there is a polynomial-time randomized reduction (with no error) from $(\mathrm{BH}, \mu_{\mathrm{st}})$ to $(\mathrm{BHP}_\phi, \mu_{\mathrm{st}})$.

Proof. There is a standard technique [Gur91] for reducing the bounded halting problem BH to its padding version BH$'$ via a randomized reduction. This technique can be also used here. One point we should clarify is a way to adjust instances for our specific requirements.

Consider any instance $\langle M, x, 0^t \rangle$ for BH. We may assume that M uses only one tape as both an input and a work tape and that M's tape alphabet is $\{0, 1, \mathrm{B}\}$. Also we assume that M does not halt unless it enters an accepting state. Clearly, we do not need more than $\max\{|x|, t\}$ tape cells for simulating M. Now we can easily modify M to some machine M' with the following properties.

(1) Whenever M' moves to a cell containing a blank symbol, it moves back to the previous position.

(2) M' has special states q'_0 and q'_1 such that M' from (q'_0, u, v) moves to $(q'_1, \lambda, d(u)(01)^m)$ in $O((|u|+|v|)^2)$ steps, where u and v represent respectively the content of the tape left and right of the tape head, $d(u)$ is a string obtained by duplicating each bit of u, and $m = (|u|+|v|-2|u|)/2$.

(3) Starting from the state q'_1, M' simulates the execution of M by interpreting each 00, 11, and 01 as 0, 1, and B of M's symbol. When the simulation enters an accepting state of M, M' also goes into an accepting state (and stays in it forever).

Then it is clear that $\langle M, x, 0^t \rangle$ is in BH if and only if $\langle M', q'_0, |x|+1, xz, y \rangle$ is in BHP_ϕ, where z and y are any strings in $\{0,1\}^*$ such that (i) $|xz| = 2\max\{|x|, t\}$, (ii) $|y| = \phi(|M| + |xz| + t')$ for some time bound t' that is large enough for the simulation of M for t steps. Therefore, our reduction produces, for given $\langle M, x, 0^t \rangle$, an instance $\langle M', q'_0, |x|+1, xz, y \rangle$ with randomly generated z and y of appropriate length. The dominancy condition can be checked easily and it is left to the reader. □

Since $(\mathrm{BH}, \mu_{\mathrm{st}})$ a complete problem in Dist$\mathcal{NP}$ [Gur91], this proposition shows that $(\mathrm{BHP}_\phi, \mu_{\mathrm{st}})$ is complete in Dist$\mathcal{NP}$ under the zero-error randomized reducibility. On the other hand, since $(\mathrm{BHP}_\phi, \mu_{\mathrm{st}})$ is a distributional problem with a flat distribution, as we will see below, $(\mathrm{BH}, \mu_{\mathrm{st}})$ is not $\leq^p_{\mathrm{m}}$-reducible to $(\mathrm{BHP}_\phi, \mu_{\mathrm{st}})$ unless $\mathcal{P} = \mathcal{NP}$.

We may use any reasonable function for ϕ. Here for the following discussion, we fix $\phi(n) = n \log n$, by which we formally mean that $\phi(n) = \lceil n \log n \rceil$ (see the smoothness discussion in the previous section). Let BHP denote the class BHP_ϕ with this ϕ. Now our class $\mathcal{C}$ is defined as a class of distributional problems (L, μ) such that (i) μ is polynomial-time computable, and (ii) (L, μ) is $\leq^p_{\mathrm{m}}$-reducible to $(\mathrm{BHP}, \mu_{\mathrm{st}})$.

Note first that if (L, μ) is $\leq^p_{\mathrm{m}}$-reducible to $(\mathrm{BHP}, \mu_{\mathrm{st}})$, then L must be in $\mathcal{NP}$. Thus, $\mathcal{C}$ is contained in Dist$\mathcal{NP}$. But $(\mathrm{BH}, \mu_{\mathrm{st}})$ is not $\leq^p_{\mathrm{m}}$-reducible to $(\mathrm{BHP}, \mu_{\mathrm{st}})$ unless $\mathcal{P} = \mathcal{NP}$. Thus, if $\mathcal{P} \neq \mathcal{NP}$, then $\mathcal{C}$ is a proper subclass of Dist$\mathcal{NP}$ because $(\mathrm{BH}, \mu_{\mathrm{st}})$ does not belong to $\mathcal{C}$. On the other hand, since $(\mathrm{BHP}, \mu_{\mathrm{st}})$ is complete in Dist$\mathcal{NP}$ under the zero-error randomized reducibility, it cannot be in Ave$\mathcal{P}$ unless Dist$\mathcal{NP} \subseteq$ Ave$\mathcal{ZPP}$; that is, $\mathcal{C} \not\subseteq$ Ave$\mathcal{P}$ unless Dist$\mathcal{NP}$ $\subseteq$ Ave$\mathcal{ZPP}$.

Proposition 6. The class $\mathcal{C}$ defined above has the following complexity.
(1) It is a subclass of Dist$\mathcal{NP}$, and the relation is proper unless $\mathcal{P} = \mathcal{NP}$.
(2) It contains Dist$\mathcal{P}$, but it is not contained in Ave$\mathcal{P}$ unless Dist$\mathcal{NP} \subseteq$ Ave$\mathcal{ZPP}$.

Clearly, the class $\mathcal{C}$ has $\leq^p_{\mathrm{m}}$-complete sets, e.g., $(\mathrm{BHP}, \mu_{\mathrm{st}})$ is one of them. On the other hand, we can define some $\leq^p_{\mathrm{tt}}$-complete problem in $\mathcal{C}$ that is not $\leq^p_{\mathrm{m}}$-complete unless $\mathcal{P} = \mathcal{NP}$.

Theorem 7. Define BHP$'$ as follows with $\phi'(n) = n \log n + \log^2 n$ (or, more formally, $\phi'(n) = \lceil n \log n + \log^2 n \rceil$). Then we have $(\mathrm{BHP}, \mu_{\mathrm{st}}) \leq^p_{\mathrm{tt}} (\mathrm{BHP}', \mu_{\mathrm{st}})$, but $(\mathrm{BHP}, \mu_{\mathrm{st}}) \not\leq^p_{\mathrm{m}} (\mathrm{BHP}', \mu_{\mathrm{st}})$

unless $\mathcal{P} = \mathcal{NP}$. That is, $(\mathrm{BHP}', \mu_{\rm st})$ is $\leq_{\rm tt}^{p}$-complete in $\mathcal{C}$ but it is not $\leq_{\rm m}^{p}$-complete unless $\mathcal{P} = \mathcal{NP}$.

$$
\begin{array}{lcll}
\mathrm{BHP}' & = & \{ & \langle M, q, i, w, u, v \rangle \ : \\
& & & \text{(i)} \quad M \text{ is NDTM, } q \text{ is its state, } i \text{ is a head position, and } w, u, v \in \{0,1\}^*, \\
& & & \text{(ii)} \quad |v| = \phi'(|M| + |w| + t - |u|) \text{ for some } t, \\
& & & \text{(iii)} \quad |u| = \log(|M| + |w| + t), \text{ and} \\
& & & \text{(iv)} \quad M \text{ has an accepting path of length } t - |u| \text{ from configuration } (q, i, w) \\
& & & \qquad \text{whose prefix is } u. \ \}
\end{array}
$$

Proof. First we show that $(\mathrm{BHP}', \mu_{\rm st})$ is $\leq_{\rm m}^{p}$-reducible to $(\mathrm{BHP}, \mu_{\rm st})$. This implies that $(\mathrm{BHP}', \mu_{\rm st})$ is indeed contained in the class $\mathcal{C}$. Let $\langle M, q, i, w, u, v \rangle$ be any instance of BHP' satisfying the syntactic conditions, i.e., the conditions (i) $\sim$ (iii), of BHP' for some number t. Let $m = |M| + |w| + t - |u|$. We map this instance to $\langle M, q', i', w', y' \rangle$, where q', i', w' are respectively M's state, head position, and tape content after executing $|u|$ steps on the path u starting from configuration (q, i, w). In order to satisfy the syntactic conditions of BHP (and keep the consistency as a reduction), y' should be a string of length $\phi(m)$. But since $\phi(m) = \phi'(m) - \log^2 m$ (recall that $|w| = |w'|$), we have $|y'| \leq |v| - \log^2(m)$; hence, we can simply use the prefix of v of appropriate length for y'. Notice that this mapping may not be one-to-one. But first note that

$$
\mu_{\rm st}(\langle M, q', i', w', y' \rangle) \ = \sum_{\tilde{v} \in V(y')} \mu_{\rm st}(\langle M, q, i, w, u, \tilde{v} \rangle),
$$

where $V(y')$ is the set of $\tilde{v}$ of length $\phi'(m)$ whose prefix is y'. Also for considering all configurations reachable to (q', i', w'), let $C(q', i', w')$ be the set of pairs of M's configurations $(\tilde{q}, \tilde{i}, \tilde{w})$ and $\tilde{u}$ of length $\log(|M| + |w| + t)$ such that the configuration (q', i', w') is reached after executing $|\tilde{u}| = \log(|M| + |w| + t)$ steps from $(\tilde{q}, \tilde{i}, \tilde{w})$ following $\tilde{u}$. Since $|\tilde{u}| = \log(|M| + |\tilde{w}| + t) = \log(|M| + |w| + t)$, $C(q', i', w')$ has at most $|M|(|M| + |w| + t)^2 \times (|M| + |w| + t)$ elements. Thus, we have

$$
\frac{\sum_{(\tilde{q}, \tilde{i}, \tilde{w}), \tilde{u} \in C(q', i', w')} \sum_{\tilde{v} \in V(y')} \mu_{\rm st}(\langle M, q, i, w, u, \tilde{v})}{|M|(|M| + |w| + t)^3} \leq \mu_{\rm st}(\langle M, q', i', w', y' \rangle).
$$

Therefore the dominancy condition is satisfied.

We observe here that the many-one reduction decreases the length of the instance by order $(\log)^2$. Let $\ell = |M| + |q| + |i| + |w| + |u| + |v|$ and $\ell' = |M| + |q'| + |i'| + |w'| + |y'|$, then if ℓ is sufficiently large, we have

$$
\ell' \ \leq \ \ell - \log^2(m) \ \leq \ \ell - \log^2(l^{1/2}) \ = \ \ell - \frac{1}{4} \log^2 \ell,
$$

since we may assume that $m^2 \leq m \log m + \log^2 m + (|M| + |w| + |q| + |i| + |u|) = |M| + |q| + |i| + |w| + |u| + |v| = \ell$, for sufficiently large ℓ.

Next suppose that there exists a $\leq_{\mathrm{m}}^{p}$-reduction from $(\mathrm{BHP}, \mu_{\mathrm{st}})$ to $(\mathrm{BHP}', \mu_{\mathrm{st}})$. We will show that this assumption implies that $\mathcal{P} = \mathcal{NP}$. Consider any $\langle M, q, i, w, y\rangle$ satisfying the syntax of BHP, and let $\langle M', q', i', w', u', v'\rangle$ be the instance of BHP$'$ obtained by the assumed reduction. We may assume that $\langle M', q', i', w', u', v'\rangle$ satisfies the syntax of BHP$'$ for some t'; i.e., $|v| = \phi'(|M'| + |w'| + t' - |u'|)$. Let $\ell = |M| + |q| + |i| + |w| + |y|$. By using the reduction from BHP$'$ to BHP explained above, we reduce further the instance $\langle M', q', i', w', u', v'\rangle$ to some instance $\langle M', q'', i'', w'', y''\rangle$ of BHP. Note that $|y''| = \phi(|M'| + |w''| + t'')$ where $t'' = t' - |u'|$.

We estimate $\ell' = |M'| + |q'| + |i'| + |w'| + |u'| + |v'|$ and $\ell'' = |M'| + |q''| + |i''| + |w''| + |y''|$, and prove that $\ell'' < \ell$, i.e., $\langle M', q'', i'', w'', y''\rangle$ is shorter than $\langle M, q, i, w, y\rangle$. First from the above analysis, we have

$$\ell'' \;\leq\; \ell' - \frac{1}{4}\log^2 \ell'$$

Now consider the case that $\ell' < \ell/2$. Then from the above bound, we immediately have $\ell'' < \ell$ for sufficiently large ℓ. Thus, consider the other case, i.e., $\ell' \geq \ell/2$. Even in this case, ℓ' cannot be so large. This is because from the dominancy condition, we have $\ell' \leq \ell + d\log\ell$ for some constant $d > 0$, and hence,

$$\ell'' \;\leq\; (\ell + d\log\ell) - \frac{1}{4}\log^2(\ell + d\log\ell) \;\leq\; (\ell + d\log\ell) - \frac{1}{4}\log^2 \ell,$$

which, by using the assumption that $\ell' \geq \ell/2$, implies $\ell'' < \ell$ if ℓ is large enough.

Therefore, the obtained instance $\langle M', q'', i'', w'', y''\rangle$ is at least one bit shorter than the original instance $\langle M, q, i, w, y\rangle$. Thus, applying this process for enough number of times, which is still polynomially bounded, we can obtain a trivial instance for BHP. Thus BHP is in $\mathcal{P}$, which implies that $\mathcal{P} = \mathcal{NP}$.

Finally, we show a $\leq_{\mathrm{tt}}^{p}$-reduction from $(\mathrm{BHP}, \mu_{\mathrm{st}})$ to $(\mathrm{BHP}', \mu_{\mathrm{st}})$. For a given instance $\langle M, i, q, w, y\rangle$ of BHP with $|y| = \phi(|M| + |w| + t)$ for some t, we only have to ask queries of the form $\langle M, i, q, w, u, v\rangle$ for all $u \in \{0,1\}^{\log m}$, where $m = |M| + |w| + t$, and v is the prefix of y of length $\phi'(|M| + |w| + t - \log m)$. (We will see below that $\phi'(|M| + |w| + t - \log m)$ is smaller than $\phi(|M| + |w| + t)$; hence, this choice of v is possible.)

Clearly, this reduction works as a disjunctive truth-table reduction from BHP to BHP$'$. To check the dominancy condition, consider any $\langle M, i, q, w, u, v\rangle$ satisfying the syntax of BHP$'$, we estimate the probability of instances in BHP that ask $\langle M, i, q, w, u, v\rangle$ in our $\leq_{\mathrm{tt}}^{p}$-reduction. First note that

$$\begin{aligned}
|v| \;&=\; \phi'(|M| + |w| + t - \log m) \\
&=\; (m - \log m)\log(m - \log m) + (\log(m - \log m))^2 \\
&\leq\; m\log m \;=\; \phi(|M| + |w| + t) \;=\; |y|.
\end{aligned}$$

Let I be the set of instances in BHP that ask $\langle M, i, q, w, u, v\rangle$. Then I consists of strings $\langle M, i, q, w, vy'\rangle$ for some y'. Thus, $\mu_{\mathrm{st}}(I)$, the total probability of instances in BHP that ask $\langle M, i, q, w, u, v\rangle$ is estimated as follows.

$$
\begin{aligned}
\mu_{\rm st}(I) &= \sum_{\langle M,i,q,w,vy'\rangle\in I} (1/\alpha)\cdot 2^{-(|M|+|i|+|q|+|w|+|v|+|y'|)} \\
&= 2^{|y'|}\times(1/\alpha)\cdot 2^{-(|M|+|i|+|q|+|w|+|v|+|y'|)} \\
&= (1/\alpha)\cdot 2^{-(|M|+|i|+|q|+|w|+|v|)} \;\leq\; |u|^2 2^{|u|}\cdot(1/\alpha')\cdot 2^{-(|M|+|i|+|q|+|w|+|u|+|v|)} \\
&= (\log m)^2 2^{\log m}\cdot \mu_{\rm st}(\langle M,w,u,v\rangle).
\end{aligned}
$$

Here $\alpha = \alpha(|M|,|q|,|i|,|w|,|v|,|y'|)$ and $\alpha' = \alpha(|M|,|q|,|i|,|w|,|u|,|v|)$. Note that $1/\alpha \leq |u|^2/\alpha'$. Since $(\log m)^2 2^{\log m}$ is bounded by $p(|\langle M,w,u,v\rangle|)$ with some polynomial p, the dominancy condition is satisfied. $\square$

References

[AT00] S. Aida and T. Tsukiji, On the difference among polynomial-time reducibilities for distributional problems (*Japanese*), in *Proc. of the LA Symposium, Winter*, RIMS publication, 2000.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1988.

[Betal92] S. Ben-David, B. Chor, O. Goldreich, and M. Ludy, On the theory of average case complexity, *Journal of Comput. and Syst. Sci.*, 44:193-219, 1992.

[Coo71] S.A. Cook, The complexity of theorem proving procedures, in *the Proc. of the third ACM Sympos. on Theory of Comput.*, ACM, 151-158, 1971.

[Gur91] Y. Gurevich, Average case completeness, *Journal of Comput. and Syst. Sci.*, 42:346–398, 1991.

[Hom97] S. Homer, Structural properties of complete problems for exponential time, in *Complexity Theory Retrospective 2* (A.L. Selman Ed.), Springer-Verlag, 135–154, 1997.

[Imp95] R. Impagliazzo, A personal view of average-case complexity, in *Proc. 10th Conference Structure in Complexity Theory*, IEEE, 134–147, 1995.

[KP92] E. Koutsoupias and C. Papadimitriou, On the greedy algorithm for satisfiability, *Infom. Process. Lett.* 43, 53–55, 1992.

[LLS75] R. Ladner, N. Lynch, and A. Selman, A Comparison of polynomial time reducibilities, *Theoretical Computer Science*, 1:103–123, 1975.

[Lev73] L.A. Levin, Universal sequential search problem, *Problems of Information Transmission*, 9:265–266, 1973.

[Lev86] L.A. Levin, Average case completeness classes, *SIAM J. Comput.*, 15:285–286, 1986.

[LY90] L. Longpré and P. Young, Cook reducibility is faster than Karp reducibility, *J. Comput. Syst. Sci.*, 41, 389–401, 1990.

[VV83] U. Vazirani and V. Vazirani, A natural encoding scheme proved probabilistic polynomial complete, *Theoret. Comput. Sci.*, 24, 291–300, 1983.

[Wang97] J. Wang, Average-case computational complexity theory, in *Complexity Theory Retrospective 2* (A.L. Selman Ed.), Springer-Verlag, 295–328, 1997.