Separating NP-Completeness Notions under Strong Hypotheses *

Klaus Ambos-Spies Levke Bentzien
Mathematisches Institut
Universität Heidelberg
D-69120 Heidelberg, Germany

Abstract

Lutz [16] proposed the study of the structure of the class **NP** = **NTIME**(*poly*) under the hypothesis that **NP** does not have *p*-measure 0 (with respect to Lutz's resource bounded measure [15]). Lutz and Mayordomo [18] showed that, under this hypothesis, **NP**-*m*-completeness and **NP**-*T*-completeness differ, and they conjectured that further **NP**-completeness notions can be separated. Here we prove this conjecture for the bounded-query reducibilities. In fact we consider a new weaker hypothesis, namely the assumption that **NP** is not *p*-meager with respect to the resource bounded Baire category concept of Ambos-Spies et al. [2]. We show that this category hypothesis is sufficient to get:

(i) For $k \geq 2$, **NP**-$btt(k)$-completeness is stronger than **NP**-$btt(k+1)$-completeness.

(ii) For $k \geq 1$, **NP**-$bT(k)$-completeness is stronger than **NP**-$bT(k+1)$-completeness.

(iii) For every $k \geq 2$, **NP**-$bT(k-1)$-completeness is not implied by **NP**-$btt(k+1)$-completeness and **NP**-$btt(2^k-2)$-completeness is not implied by **NP**-$bT(k)$-completeness.

(iv) **NP**-*btt*-completeness is stronger than **NP**-*tt*-completeness.

1 Introduction

Since it is commonly believed that **NP** differs from **P**, the internal structure of **NP** has been studied under the hypothesis that **NP** $\neq$ **P**. Classical results in this direction are Mahaney's theorem stating that the **NP**-*m*-complete (i.e. Karp-complete) sets are not sparse ([19]) and Ladner's result that, for any of

*Supported by the Human Capital and Mobility Program of the European Community under grant CHRX-CT93-0415 (COLORET).

the standard polynomial-time reducibilities, there are sets in **NP** which are neither complete nor in **P** ([13]). Many of the fundamental questions on the structure of **NP**, however, remained open when working with the $\mathbf{P} \neq \mathbf{NP}$ hypothesis. This made researchers work with stronger hypotheses. Natural and useful assumptions made in the literature include stronger separation hypotheses for the polynomial hierarchy: For instance, Karp and Lipton [12] expanded Mahaney's theorem from the **NP**-m-complete sets to the **NP**-T-complete (i.e. Cook-complete) sets under the assumption that $\Sigma_2^P \neq \Pi_2^P$. Since even these stronger assumptions on the polynomial-time hierarchy did not help in answering some of the fundamental questions on the structure of **NP**, Lutz [16] suggested to work with another hypothesis concerned with the relation between **NP** and deterministic exponential time.

Note that, for the localization of **NP** in the deterministic time hierarchy, it is only known that $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP} = \mathbf{DTIME}(2^{poly})$, and there are relativizations realizing the two extremes. Assuming $\mathbf{NP} = \mathbf{EXP}$, most of the fundamental questions on the structure of **NP** can be resolved since the structure of **EXP** is well understood. This assumption, however, has some consequences, like $\mathbf{NP} = co\text{-}\mathbf{NP}$, which are widely disbelieved, whence this assumption is not considered to be plausible. Therefore Lutz [16] proposed to adopt a weaker assumption, namely that **NP** contains a non-negligible part of $\mathbf{E} = \mathbf{DTIME}(2^{lin})$. Using his resource-bounded measure theory ([15]), Lutz formalized this "non-smallness" hypothesis by saying that **NP** does not have polynomial-time measure 0 ($\mu_P(\mathbf{NP}) \neq 0$). This hypothesis proved to suffice to settle some of the questions on the structure of **NP**, which could not be answered by using only the weaker hypothesis $\mathbf{NP} \neq \mathbf{P}$ (see [18] or Section 12 of [17] for more details). The probably most interesting consequence of Lutz's non-smallness hypothesis obtained so far, is the separation of **NP**-m-completeness and **NP**-T-completeness (Lutz and Mayordomo [18]). In fact, Lutz and Mayordomo showed that there is a set A which is both **NP**-$btt(3)$-complete and **NP**-$bT(2)$-complete, i.e. a set which is complete for **NP** under both polynomial-time truth-table reducibility of norm 3 (that is a non-adaptive reduction allowing 3 queries) and polynomial-time Turing reducibility of norm 2 (that is an adaptive reduction allowing 2 queries), but which is not **NP**-m-complete. In [20] Mayordomo further improved this result by showing that the set A is not even **NP**-$btt(2)$-complete. Lutz and Mayordomo conjectured that this separation result can be extended to other standard polynomial time reducibilities.

The goal of this paper is to prove some of the most natural instances of this conjecture. Assuming $\mu_P(\mathbf{NP}) \neq 0$ we show:

(1) For every $k \geq 2$, **NP**-$btt(k)$-completeness is stronger than **NP**-$btt(k+1)$-completeness.

(2) For every $k \geq 1$, **NP**-$bT(k)$-completeness is stronger than **NP**-$bT(k+1)$-completeness.

(3) For $k \geq 2$, $\mathbf{NP}$-$bT(k-1)$-completeness is not implied by $\mathbf{NP}$-$btt(k+1)$-completeness and $\mathbf{NP}$-$btt(2^k-2)$-completeness is not implied by $\mathbf{NP}$-$bT(k)$-completeness.

(4) $\mathbf{NP}$-*btt*-completeness is stronger than $\mathbf{NP}$-*tt*-completeness.

In fact, we obtain these results under some weaker hypothesis. We use the resource-bounded Baire category concept of Ambos-Spies, Fleischhack and Huwig [2] for describing the size of complexity classes and we express non-smallness of $\mathbf{NP}$ in this context by the hypothesis that $\mathbf{NP}$ is not p-meager (in the sense of [2]). Though, classically, Baire category and Lebesgue measure are two classification schemes which in general are incompatible, the resource-bounded category concept of [2] is sufficiently weak to become compatible with the resource-bounded measure concept of [15]: For any class $\mathbf{C}$, $\mu_P(\mathbf{C}) \neq 0$ implies that $\mathbf{C}$ is not p-meager (Ambos-Spies et al. [4]), whence Lutz's non-smallness axiom for $\mathbf{NP}$ based on measure implies our category-based non-smallness axiom. In fact, as we will show below, Lutz's hypothesis is strictly stronger than our hypothesis in the sense that, relative to some oracle, $\mathbf{NP}$ satisfies the latter but not the former hypothesis.

The outline of the paper is as follows.

In Section 2 we introduce the fragment of Ambos-Spies et al. resource-bounded category concept necessary for stating our non-smallness hypothesis for $\mathbf{NP}$ and for working with this hypothesis, and we compare it to Lutz's measure hypothesis for $\mathbf{NP}$.

In Section 3 we prove the separation theorems (1)–(3) assuming that $\mathbf{NP}$ is not p-meager. The proof of (1) uses some ideas of Lutz and Mayordomo's separation result but requires some additional new features. The proofs of the other two theorems follow the same pattern. Therefore, to a certain extent, we establish them in parallel.

In Section 4, building upon (1), we prove the separation theorem (4), and in Section 5 we state some open problems.

In our notation we do not distinguish between numbers and strings, and sets and their characteristic sequences. I.e. n is identified with the n-th string so that $|n| \approx log(n)$, and $X \upharpoonright n = X(0) \dots X(n-1)$ is the characteristic string of the initial segment of X of length n. $X_{=n}$ denotes the set of strings in X of length n. In general, our notation follows the setting of Ambos-Spies in [1] where also more details on and motivation for the category and genericity concepts used here can be found. For more background information on resource-bounded measure and randomness we refer to the recent survey by Lutz [17]. Another recent survey by Ambos-Spies and Mayordomo [3] focuses on the relations between resource-bounded measure and category. There also a preliminary account of the category-based non-smallness axiom introduced here is given, and some relations to other strong hypotheses for $\mathbf{NP}$ are discussed. The polynomial time reducibilities discussed here were introduced by Ladner et al. in [14] and we recommend the survey of Buhrman and Torenvliet [8] for an overview on completeness notions.

2 Largeness Axioms for NP

In this section we present some material on resource-bounded category and genericity required for introducing (and working with) our category-based non-smallness hypothesis for **NP**. We first define the general category concept developed by Ambos-Spies [1] (for polynomial time) which is an amalgamation of the category concepts of Ambos-Spies, Fleischhack and Huwig [2] and Fenner [9], and then define the category concept of [2] by restricting the general category concept.

Definition 2.1 *([1], [2], [9])*

1. *An* n^k-extension function f *is a partial function* $f\colon \{0,1\}^* \to (\{0,1\}^* \times \{0,1\})^*$ *such that* $f \in \mathbf{DTIME}(n^k)$ *and, whenever* f *is defined on input* $X{\restriction}n = X(0)\dots X(n-1)$*, then*

   (5) $f(X{\restriction}n) = (y_0, i_0), \dots, (y_m, i_m)$

   for some $m \geq 0$*, some strings* y_j *with* $n \leq y_0 < \dots < y_m$*, and some* $i_j \in \{0,1\}$ $(0 \leq j \leq m)$*.*

2. *A* p-extension function *is an* n^k*-extension function for some* $k \geq 1$*.*

3. *A set* A meets *the* n^k*-extension function* f at n *if* $f(A{\restriction}n){\downarrow}$*, say (5) holds for* $X = A$*, and* $A(y_j) = i_j$ *for* $j \leq m$*; and* A meets f *if* A *meets* f *at some* n*.*

4. *The* n^k*-extension function* f *is* dense along A *if* $f(A{\restriction}n){\downarrow}$ *for infinitely many* n*.*

5. *A set* G *is* general n^k-generic *if* G *meets every* n^k*-extension function* f *which is dense along* G*; and* G *is* general p-generic *if* G *is general* n^k*-generic for all* $k \geq 1$*.*

6. *A class* $\mathbf{M}$ *is* general p-meager *if, for some* $k \geq 1$*,* $\mathbf{M}$ *does not contain any* n^k*-generic set.*

By considering only extension functions specifying the next bit, we obtain the category concept of Ambos-Spies et al. [2]:

Definition 2.2 *([2])*

1. *A* simple n^k-extension function f *is an* n^k*-extension function* f *such that, whenever* $f(X{\restriction}x){\downarrow}$*, then* $f(X{\restriction}x) = (x, i)$ *for some* $i \leq 1$*, in which case we also write* $f(X{\restriction}x) = i$*.*

2. *A set G is* n^k-generic *if G meets every simple n^k-extension function which is dense along G; and G is* p-generic *if G is n^k-generic for all $k \geq 1$.*

3. *A class* $\mathbf{M}$ *is* p-meager *if, for some $k \geq 1$,* $\mathbf{M}$ *does not contain any n^k-generic set.*

By applying a general observation on classes closed under $\leq_m^P$, non-p-meagerness of $\mathbf{NP}$ can be expressed in terms of genericity as follows:

Lemma 2.3 *([1])* $\mathbf{NP}$ *is not p-meager iff* $\mathbf{NP}$ *contains a p-generic set.*

So our non-smallness assumption for $\mathbf{NP}$ based on the category concept of [2] can be stated as follows:

(G) $\exists\, G \in \mathbf{NP}$ (G p-generic) (Genericity Hypothesis)

By elementary properties of the n^k-generic sets, many properties of $\mathbf{NP}$ derived from Lutz's measure hypothesis also easily follow from *(G)*. E.g. any n^2-generic set is $\mathbf{P}$-bi-immune (see [4]), whence *(G)* implies

(B) $\exists\, B \in \mathbf{NP}$ (B $\mathbf{P}$-bi-immune) (Bi-immunity Hypothesis)

For a $\mathbf{P}$-bi-immune $B \in \mathbf{NP}$ and, for an infinite set $C \in \mathbf{P}$, $B \cap C \in \mathbf{NP} - \mathbf{P}$, whence, by letting $C = \{0\}^*$ or $C = \{0^{2^n} : n \geq 0\}$ the Bi-immunity Hypothesis *(B)* implies

(T) $\exists\, A \in \mathbf{NP} - \mathbf{P}$ (A tally)

and

(TT) $\exists\, A \in \mathbf{NP} - \mathbf{P}$ (A exptally)

These assumptions are equivalent to $\mathbf{E} \neq \mathbf{NE}$ and $\mathbf{EE} \neq \mathbf{NEE}$, respectively, and the existence of sets in $\mathbf{NP} - \mathbf{P}$ which are p-selective (see Selman [22]) is among the consequences of *(T)*, while *(TT)* e.g. implies the existence of search problems in $\mathbf{NP}$ which cannot be reduced to their corresponding decision problems (Bellare and Goldwasser [6]). A more detailed discussion of the consequences of the Bi-immunity Hypothesis can be found in Lutz and Mayordomo [18].

In order to relate *(G)* to Lutz's hypothesis we need the observation of Ambos-Spies et al. [5] that every p-random set is p-generic and the analog of Lemma 2.3 in the measure setting.

Lemma 2.4 *([5]) Every p-random set is p-generic.*

Lemma 2.5 *([5])* $\mu_p(\mathbf{NP}) \neq 0$ *iff* $\mathbf{NP}$ *contains a p-random set.*

So Lutz's non-smallness assumption that $\mu_p(\mathbf{NP}) \neq 0$ can be rephrased by

(R) $\exists\, R \in \mathbf{NP}$ (R p-random) (Randomness Hypothesis)

Proposition 2.6 *(R) implies (G). Hence any property of* **NP** *which can be obtained from (G) can also be obtained from (R).*

Proof. By Lemma 2.4. □

In fact, the implication *(R)* $\Rightarrow$ *(G)* is strict in the following sense:

(i) *(R)* $\Rightarrow$ *(G)* holds relative to every oracle.

(ii) *(G)* $\Rightarrow$ *(R)* fails relative to some oracle.

Note that (i) holds by relativizing Lemma 2.4 and (ii) holds by the following theorem:

Theorem 2.7 *There is an oracle A such that, relative to A,* **NP** *contains a p-generic set but no p-random set.*

Proof. We will construct a set A with the required properties in stages. I.e. we will effectively enumerate a sequence of finite characteristic functions $(\alpha_s)_{s\geq 0}$ which has the characteristic function of A as its limit.

In order to guarantee that $\mathbf{NP}^A$ does not contain any p^A-random set we will ensure that every set in $\mathbf{NP}^A$ agrees with some $\mathbf{P}^A$-set on all strings of length n for infinitely many numbers n. This will be achieved by letting the oracle A look like the canonical $\mathbf{NP}^A$-complete set K^A on sufficiently large intervals infinitely often. Let N_e be the e-th nondeterministic oracle Turing machine with respect to some standard numbering and let

$$K^A = \{\langle 0^e, x, 0^n\rangle : N_e^A \text{ accepts } x \text{ in } \leq n \text{ steps}\}$$

(where, for technical convenience, we assume that $|\langle x, y, z\rangle|$ is odd for all strings x, y, z). Then the construction of A will ensure that

$$\textit{(6)} \qquad \exists^\infty n\, \forall x\, (n \leq |x| \leq 2^n \;\Rightarrow\; A(x) = K^A(x))$$

That this suffices to eliminate random sets in $\mathbf{NP}^A$ is shown as follows.

Claim 1. Assume that (6) holds. Then $\mathbf{NP}^A$ does not contain any p^A-random set.

Proof. Given $B \in \mathbf{NP}^A$, there is an index e and a polynomial p such that $x \in B$ iff $\langle 0^e, x, 0^{p(|x|)}\rangle \in K^A$. Since $|\langle 0^e, x, 0^{p(|x|)}\rangle|$ is polynomially bounded in $|x|$, it follows from (6) that there are infinitely many numbers n such that, for the $\mathbf{P}^A$-set $\hat{B} = \{x : \langle 0^e, x, 0^{p(|x|)}\rangle \in A\}$, $B_{=n} = \hat{B}_{=n}$. But the observation that, for any p-random set R, $R_{=n} \neq \emptyset$ for all sufficiently large n (see e.g. Lemma 2.10 in [1]), can be easily extended to show that, for any $\mathbf{P}$-set C, $R_{=n} \neq C_{=n}$ a.e. and that this fact relativizes. This completes the proof of Claim 1.

In order to achieve the second goal of the construction we will ensure that the set

$$G^A = \{x : \exists y\, (|y| = |x|^2 \;\&\; xy \in A)\}$$

will be p^A-generic. Fix a recursive enumeration $(f_e)_{e\geq 0}$ of the oracle dependent simple p-extension functions such that w.l.o.g. f_e is an n^e-extension function. Then to make G^A p-generic relative to A it suffices to meet the requirements

$$R_e : f_e^A \text{ dense along } G^A \Rightarrow G^A \text{ meets } f_e^A$$

for $e \geq 0$. These requirements will be met by a so-called slow diagonalization, a variant of the finite extension method: If the hypothesis of R_e will hold then at some stage s of the construction we will choose the finite extension α_s of the previously specified finite part α_{s-1} of A in such a way that α_s will force that $f_e^A(G^A{\restriction}s)\downarrow = G^A(s)$. For each requirement this action has to be taken at most once and if this action becomes necessary then, by the hypothesis of the requirement, there will be inifinitely many stages at which we can take this action. This will allow us to spread out the actions for meeting the requirements R_e in such a way that, by letting $A(x) = K^A(x)$ in the intermediate phases, condition (6) will be satisfied. In the construction below this will be implemented by allowing only requirements R_e to act at stage s for which $e < d(|s|)$ for some slowly growing function d.

If we take action for requirement R_e and choose α_s to force $f_e^A(G^A{\restriction}s)\downarrow = G^A(s)$ we have to make sure that this action will not simultaneously force $f_{e'}^A(G^A{\restriction}s')\downarrow \neq G^A(s')$ for some $e' < e$ and s'. Otherwise, for some requirement $R_{e'}$, the actions of the requirements R_e with $e > e'$ could force $f_{e'}^A$ to be dense along G^A in such a way that we will not be able to ensure that G^A meets $f_{e'}^A$. So the combined actions of the lower priority requirements could cause the failure of R_e. This problem is overcome by allowing a requirement $R_{e'}$ of higher priority to *complain* about the extension α_s proposed by a lower priority requirement R_e. If this happens R_e will not become active but instead we will choose the extension $\alpha_{s'}, s' \geq s$, in such a way that $f_e^A(G^A{\restriction}s)$ will be undefined, thereby eliminating the reason for R_e becoming active at stage s.

For stating the formal construction we need the following notation.

In the following, lower case Greek letters will denote finite partial functions from Σ^* to $\{0,1\}$. We say β *extends* α ($\alpha \subseteq \beta$) if the graph of α is contained in the graph of β and we say β *extends* α *along* γ (denoted by $\beta = \alpha \sqcup \gamma$) if $\beta(x) = \alpha(x)$ for $x \in dom(\alpha)$ and $\beta(x) = \gamma(x)$ for all $x \in dom(\gamma) \setminus dom(\alpha)$. Similarly, a set X extends α ($\alpha \subset X$) if α coincides with the characteristic function of X on $dom(\alpha)$. If used as an oracle, a finite function α is interpreted as the finite set $\{x \in dom(\alpha) : \alpha(x) = 1\}$. So, for a query $x \notin dom(\alpha)$, the oracle α returns the answer 0.

For a string x call the set $code(x) = \{xy : |y| = |x|^2\}$ the *coding region of* x, let $m(x)$ be the least element of $code(x)$, and call a finite function α *x-honest* if, for all $z \in dom(\alpha) \cap code(x)$, $\alpha(z) = 0$. Then $x \in G^A$ iff $A \cap code(x) \neq \emptyset$, $A{\restriction}m(x)$ is x-honest, and, for any x-honest α with $code(x) \not\subseteq dom(\alpha)$ we can find extensions β_i forcing $G^X(x) = i$ for all extensions X of β_i by letting β_0 be the extension of α along $\{(z,0) : z \in code(x)\}$ and by letting β_1 be the extension of α along $\{(z,1)\}$ for some string $z \in code(x) - dom(\alpha)$. In the construction of

A below, the part α_{s-1} of A specified by the end of stage $s-1$ will be chosen to extend $A\restriction m(s)$ whence $G^A\restriction s$ will be determined by the end of stage $s-1$, namely $G^A\restriction s = G^{\alpha_{s-1}}\restriction s$.

For describing the dependence of $f_e^X(G^X\restriction x)$ on the oracle X let $\varphi(X,e,x)$ be the use function of this computation, i.e. the finite characteristic function determining the oracle queries in the computation of $f_e^X(G^X\restriction x)$. Then, for any oracles X and Y such that $X\restriction m(x) = Y\restriction m(x)$ (whence $G^X\restriction x = G^Y\restriction x$) and $\varphi(X,e,x) = \varphi(Y,e,x)$, $f_e^X(G^X\restriction x) = f_e^Y(G^Y\restriction x)$. Moreover, since f_e is an n^e-extension function, $|dom(\varphi(X,e,x))| \leq |G^X\restriction x|^e \leq 2^{e\cdot|x|}$ and, for any $z \in dom(\varphi(X,e,x))$, $|z| \leq 2^{e\cdot|x|}$. This implies that for any oracle X the set

$$Q_s^X = \{z : \exists x \leq s\ \exists e < d(|s|)\ (z \in dom(\varphi(X,e,x)))\}$$

has at most

$$q_s = 2^{|s|+1}\cdot d(|s|)\cdot 2^{d(|s|)\cdot|s|}$$

elements. Note that $q_s < 2^{|s|^2}$ for almost all numbers s if $d(|s|) \leq \log(|s|)$ whereas $|code(s)| = 2^{|s|^2}$. This will ensure that in the construction we can choose α_s to force a computation $f_e^A(G^A\restriction s)$ to converge without exhausting $code(s)$ completely. We assume that the function d limiting the number of requirements which are considered at some stage is chosen so that $q_s < 2^{|s|^2}$ holds for all s and such that, for all $e < d(|s|)$, $2^{e\cdot|s|} < 2^{2^{|s|}}$. Moreover, in order to guarantee (6), we choose d to be nondecreasing and to satisfy $d(n) \leq \log(n)$ and

(7) $\quad n_j = \mu l(d(l) > j) \Rightarrow n_{j+1} > \delta(n_j)$ for all $j \geq 0$

where $\delta(m) = 2^{2^{\cdot^{\cdot^{2^m}}}}\}6m$-times.

Stage s ($s \geq 0$) of the construction of A consists of two parts where the first part contains the action for meeting the requirements R_e while in the second part condition (6) will be ensured. In stage s we do not only define the initial part α_s of A but also sets Sat_s and $Undef_s$ where Sat_s contains the indices of the requirements which have been satisfied by the end of stage s while, for a pair $(e,x') \in Undef_s$, $f_e^A(G^A\restriction x')$ will be made undefined unless some requirement $R_{e'}$ with $e' < e$ will be satisfied at a stage $> x'$. The initial values of these parameters are $\alpha_{-1} = Sat_{-1} = Undef_{-1} = \emptyset$.

Stage s of the construction of A:

Step 1: Requirement R_e *requires attention via* β if $e < d(|s|)$, $e \notin Sat_{s-1}$, β is an s-honest extension of α_{s-1} and $f_e^\beta(G^\beta\restriction s)$ is defined. Requirement R_e *requires attention* if R_e requires attention via some β. Requirement $R_{e'}$ *complains about* $\beta \supseteq \alpha_{s-1}$ if $e' \notin Sat_{s-1}$ and there are strings x, z such that $z \in code(x)$, $z \in dom(\beta) \setminus dom(\alpha_{s-1})$, $\beta(z) = 1$ and there is an extension γ of β such that

(*) $f^{\gamma}_{e'}(G^{\gamma}\upharpoonright x) = 0$ or

(**) $\exists (e', x') \in Undef_{s-1}$ ($f^{\gamma}_{e'}$ queries z on input $G^{\gamma}\upharpoonright x'$).

Now if there is a requirement R_e and a finite function β such that R_e requires attention via β and no $R_{e'}$ with $e' < e$ complains about β then fix the least such e and β (in this order), let γ be the extension of β along the union of the use functions $\varphi(\beta, e', x)$ for all numbers $e' \leq e$ and strings x such that $x = s$ or $(e', x) \in Undef_{s-1}$, let z be the least string in $code(x) \setminus dom(\gamma)$, and let

$$\delta = (\gamma \cup (z, f^{\beta}_{e}(G^{\beta}\upharpoonright s))) \sqcup (code(s) \times \{0\}).$$

In this case say that R_e *receives attention* (*via* β) and let

$$\begin{array}{rcl} Sat_s & = & Sat_{s-1} \cup \{e\} \\ Undef_s & = & Undef_{s-1}. \end{array}$$

Otherwise, let $\delta = \alpha_{s-1} \sqcup (code(s) \times \{0\})$ and let

$$\begin{array}{rcl} Sat_s & = & Sat_{s-1} \\ Undef_s & = & Undef_{s-1} \cup \{(e, s) : R_e \text{ requires attention at stage } s\} \end{array}$$

Step 2: For the extension δ of α_{s-1} defined in Step 1 let

$$\alpha_s = \delta \sqcup (K^{\alpha_s}\upharpoonright m(s+1)).$$

This completes the construction. Note that $K^{\beta}(x)$ only depends on β for the strings in $dom(\beta)$ which are less than x. So in Step 2 α_s and $K^{\alpha_s}\upharpoonright m(s+1)$ can be inductively defined by fixing $\alpha_s(y)$ for the strings $y < m(s+1)$ with $y \notin dom(\delta)$ in order.

The correctness of the construction is established by the following claims. Let $Sat = \bigcup_{s \geq 0} Sat_s$ and $Undef = \bigcup_{s\geq 0} Undef_s$.

Claim 2. For all $s \geq 0$, α_s is well defined, α_s extends α_{s-1}, $dom(\alpha_s)$ contains all strings less than $m(s+1)$, and, for $z \in dom(\alpha_s)$ with $z \not\leq m(s+1)$, $|z| < 2^{2^{|s|}}$. Moreover, for any $z \in code(s)$, $z \in dom(\alpha_{s-1})$ implies that $z \in dom(\varphi(\alpha_{s-1}, e, s'))$ for some $e < d(|s-1|)$ and $s' < s$.

Proof. The proof, which is by induction on s, easily follows from the following observations: If no requirement receives attention at stage s then there is no string $z \in code(s')$ for $s' > s$ such that $z \in dom(\alpha_s) \setminus dom(\alpha_{s-1})$. If R_e receives attention at stage s via β then minimality of β and the inductive hypothesis for the second part of the claim ensure that, for the corresponding γ, $dom(\gamma) \cap code(s) \subseteq Q^{\gamma}_s$ whence the string $z \in code(s) \setminus dom(\gamma)$ required for the definition of the δ-part of α_s exists.

By Claim 2 the sequence $(\alpha_s)_{s\geq 0}$ is well defined and determines a unique set A. Moreover, for any s and $\beta \supseteq \alpha_{s-1}$, $G^{\alpha_{s-1}}\restriction s = G^{\beta}\restriction s = G^A\restriction s$, which will be tacitly used below. It remains to show that A satisfies condition (6) and the requirements $R_e (e \geq 0)$ are met.

Claim 3. Every requirement R_e receives attention at most once and if R_e receives attention then R_e is met.

Proof. Assume that R_e receives attention via β at stage s. Then $e \in Sat_{s'}$ for all $s' \geq s$ whence R_e does not require attention after stage s. Moreover, by choice of β and definition of α_s, $f_e^{\beta}(G^{\beta}\restriction s) = f_e^{\alpha_s}(G^{\alpha_s}\restriction s)\downarrow$ since α_s extends $\varphi(\beta, e, s)$, and there is a string $z \in code(s)$ with $\alpha_s(z) = f_e^{\alpha_s}(G^{\alpha_s}\restriction s)$ and $\alpha_s(z') = 0$ for all $z' \in code(s) - \{z\}$. Hence $f_e^A(G^A\restriction s) = G^{\alpha_s}(s) = G^A(s)$.

Claim 4. Condition (6) is satisfied.

Proof. Let x_n be the least string x with $d(|x|) = n$ and let $l(n) = |x_n|$. We will show that for all sufficiently large n there is a number m with $l(n) \leq m < 2^m \leq l(n+1)$ such that $A(x) = K^A(x)$ for all strings x with $m \leq |x| \leq 2^m$.

Note that $l(n) \geq n$ whence, by (7), $l(n+1) \geq 2^{2^{\cdot^{\cdot^{2^{l(n)}}}}}$ where the exponential function is iterated $6n$ times. Hence, for l_i $(i = 0, \ldots, 3n)$ defined by $l_0 = l(n)$ and $l_{i+1} = 2^{2^{l_i}}$, $l(n) = l_0 < l_1 < \ldots < l_{3n} \leq l(n+1)$. On the other hand, only the n requirements $R_0, \ldots, R_{n-1}$ can receive attention at a stage $s < x_{n+1}$ and, by Claim 3, each requirement receives attention at most once. So we can pick $k \leq 2n$ such that no requirement receives attention at any stage s with $l_k \leq |s| < l_{k+2}$. It follows by construction that $A(x) = K^A(x)$ for all strings $x < m(0^{l_{k+2}})$ with $x \notin dom(\alpha_{s_0})$ where $s_0 = 1^{l_k - 1}$. Since $|x| < l_{k+1}$ for $x \in dom(\alpha_{s_0})$ by Claim 2, this implies the claim.

Claim 5. Assume that R_e never receives attention and that no $R_{e'}$ with $e' < e$ receives attention after stage s_e. Then, for any stage $s > s_e$ at which R_e requires attention, $f_e^A(G^A\restriction s)$ is undefined.

Proof. For a contradiction assume that R_e requires attention at stage $s > s_e$ and $f_e^A(G^A\restriction s)$ is defined. Let $\beta = \alpha_{s-1} \sqcup \varphi(A, e, s)$ and fix $\hat{s} \geq s$ minimal such that $\beta \subseteq \alpha_{\hat{s}}$. Then $f_e^{\beta}(G^{\beta}\restriction s) = f_e^A(G^A\restriction s)\downarrow$ via the same computation (i.e. $\varphi(\beta, e, s) = \varphi(A, e, s)$). We will show that

(8) β is s-honest and no $R_{e'}$ with $e' < e$ complains about β at stage s.

Since $f_e^{\beta}(G^{\beta}\restriction s)\downarrow$ this will imply that R_e requires attention via β at stage s and no higher priority requirement complains about β. So R_e or a requirement $R_{e'}$ with $e' < e$ will receive attention at stage s contrary to the choice of s_e.

Note that α_{s-1} is s-honest since R_e requires attention at stage s. Moreover, for $s' \geq s$ such that no requirement receives attention at a stage s'' with $s \leq s'' \leq s'$, *(9)* $\forall x\ (\alpha_{s-1}\ x\text{-honest} \Rightarrow \alpha_{s'}\ x\text{-honest})$

Namely, in Step 1 of such stages s'' only pairs $(z, 0)$ are added to $\alpha_{s''-1}$ and, by convention, $|\langle u, v, w\rangle|$ is odd for all strings u, v, w while any string z in a coding region has even length, whence $K^{\alpha_{s''}}(z) = 0$. So $\alpha_{s''}(z) = 0$ for all strings z from a coding region added to $\alpha_{s''}$ in Step 2.

Now, if we assume that (9) holds for $s' = \hat{s}$ then (8) is obvious. Hence w.l.o.g. we may fix s' with $s \leq s' \leq \hat{s}$ minimal such that some requirement $R_{\hat{e}}$ receives attention at stage s', say via $\hat{\beta}$. By construction, either $s' = s$ or (e, s) is put into $Undef_s$ at stage s. In any case, $\alpha_{s'-1}$ and $\hat{\beta}$ are s-honest and $\alpha_{s'}$ extending $\hat{\beta}$ is chosen in such a way that the computation $f_e^{\hat{\beta}}(G^{\hat{\beta}}\restriction s)$ is preserved by the choice of γ. Hence $f_e^{\hat{\beta}}(G^{\hat{\beta}}\restriction s) = f_e^A(G^A\restriction s) = f_e^{\beta}(G^{\beta}\restriction s)$ and $\beta(x) = 0$ for all $x \in dom(\beta) \setminus dom(\hat{\beta})$. Therefore β is s-honest and if there is a $z \in dom(\beta) \setminus dom(\alpha_{s-1})$ to complain about then $z \in dom(\hat{\beta}) \setminus dom(\alpha_{s'-1})$ by minimality of s'. Hence, if $e' < e$ and $R_{e'}$ complains about β at stage s then $e' < e < \hat{e}$ and $R_{e'}$ complains about $\hat{\beta}$ at stage s', too, contrary to the assumption about $\hat{\beta}$.

This completes the proof of Claim 5.

Claim 6. Every requirement R_e is met.

Proof. For a contradiction assume that R_e is not met. Then $f_e^A(G^A\restriction s)$ is defined for infinitely many strings s and, by Claim 3, $e \notin Sat$. Moreover, again by Claim 3, we may fix s_e such that no requirement $R_{e'}$, $e' \leq e$, receives attention after stage s_e. Now take $s > s_e$ such that $z < s$ for all $z \in dom(\alpha_{s_e})$ and such that $f_e^A(G^A\restriction s)$ is defined.

Then, by Claim 5, R_e does not require attention at stage s. On the other hand, for $\beta = \alpha_{s-1} \sqcup \varphi(A, e, s)$, $\beta \subset A$ and $f_e^{\beta}(G^{\beta}\restriction s) = f_e^A(G^A\restriction s)\downarrow$. Hence β is not s-honest. By $\beta \subset A$ this implies that $G^A(s) = 1$ whence $f_e^A(G^A\restriction s) = 0$ since otherwise R_e will be met. Moreover, since $\beta \subset A$ and $A\restriction m(s+1) \subseteq \alpha_s$, α_s is not s-honest, whence we may fix $s' \leq s$ minimal such that $\alpha_{s'}$ is not s-honest. Note that $s' > s_e$ since $\alpha_{s_e} \subseteq A\restriction m(s)$.

Now, as shown in the proof of Claim 5 already, the extension α_{t+1} of an x-honest α_t is x-honest again unless a requirement receives attention at stage $t+1$. So, by $s_e < s'$, a requirement $R_{e'}$ with $e' > e$ receives attention at stage s', say via $\hat{\beta}$. Note that $\hat{\beta} \subseteq \alpha_{s'} \subseteq \alpha_s \subset A$ and distinguish the following two cases.

If $s' < s$ then non-s-honesty of $\alpha_{s'}$ implies that $\hat{\beta}$ is not s-honest since for $z \in code(s)$ with $z \in dom(\alpha_{s'}) \setminus dom(\hat{\beta})$, $\alpha_{s'}(z) = 0$. On the other hand, by $\hat{\beta} \subset A$, $f_e^{\gamma}(G^{\gamma}\restriction s) = f_e^A(G^A\restriction s) = 0$ for $\gamma = \hat{\beta} \sqcup \beta$, whence requirement R_e will complain about $\hat{\beta}$ at stage s'. So $R_{e'}$ will not receive attention via $\hat{\beta}$ contrary to assumption.

If $s' = s$ then $\hat{\beta}$ is s-honest since $R_{e'}$ requires attention via $\hat{\beta}$ at stage s. Moreover, by $e' > e$, $\alpha_s \supseteq \hat{\beta}$ is chosen to preserve $f_e^{\hat{\beta}}(G^{\hat{\beta}}\restriction s)$, i.e., $f_e^{\hat{\beta}}(G^{\hat{\beta}}\restriction s) =$

$f_e^A(G^A \upharpoonright s)$ and $\varphi(\hat{\beta}, e, s) = \varphi(A, e, s) \subseteq \alpha_s$. So, by choice of β, $\beta \subseteq \hat{\beta} \sqcup \varphi(\hat{\beta}, e, s)$ whence, by s-honesty of $\hat{\beta}$, β is s-honest too. A contradiction.

This completes the proof of Theorem 2.7. □

We close this section with a property of the p-generic sets needed in the following. The concept of a simple extension function underlying the genericity concept of [2] considered here may appear quite weak. Ambos-Spies [1], however, has shown that in the definition of p-genericity simple extension functions can be replaced by *bounded* extension functions, where an extension function f is *k-bounded* ($k \geq 1$) if, whenever $f(X \upharpoonright x)$ is defined, then $f(X \upharpoonright x) = (y_0, i_0), \ldots, (y_m, i_m)$ for some $m < k$.

Lemma 2.8 *([1]) Let G be p-generic and let f be a k-bounded n^c-extension function which is dense along G ($k, c \geq 1$). Then G meets f.*

Here we will need a somewhat stronger observation which can be proved in a similar way: For $f(X \upharpoonright x) = (y_0, i_0), \ldots, (y_m, i_m)$ it is not necessary that $f(X \upharpoonright x)$ can be computed in $|f(X \upharpoonright x)|^c = O(2^{c \cdot |x|})$ steps but it suffices that, for $j \leq m$, $(y_0, i_0), \ldots, (y_j, i_j)$ can be computed in $O(2^{c \cdot |y_j|})$ steps. In the following lemma we state a special case of this observation which will be sufficient for our investigations here.

Lemma 2.9 *Let $l, c \geq 1$ and let f be an extension function such that, for almost all initial segments $\alpha = X \upharpoonright 0^n$ of length $2^n - 1$ ($n \geq 1$), $f(\alpha)$ is defined and*

$$f(\alpha) = (y_{\alpha,1}, i_{\alpha,1}), \ldots, (y_{\alpha,l_\alpha}, i_{\alpha,l_\alpha})$$

where $l_\alpha \leq l$, $pos(\alpha) = (y_{\alpha,1}, \ldots, y_{\alpha,l_\alpha})$ is computable in $2^{c \cdot n}$ steps and $i_{\alpha,j}$ is computable in $2^{c \cdot |y_{\alpha,j}|}$ steps. Then every p-generic set meets f.

Proof. W.l.o.g. we may assume that $l_\alpha = l$ for all strings α on which the extension function f is defined. We split f into l simple n^{c+1}-extension functions $f_1, \ldots, f_l$ as follows: Given k with $1 \leq k \leq l$ and a string $X \upharpoonright y$ let $f_k(X \upharpoonright y) = i_{\alpha,k}$, where α is the shortest initial segment $X \upharpoonright 0^n$ of $X \upharpoonright y$ such that, for $f(\alpha) = (y_{\alpha,1}, i_{\alpha,1}), \ldots, (y_{\alpha,l}, i_{\alpha,l})$, $y = y_{\alpha,k}$ and $(X \upharpoonright y)(y_{\alpha,j}) = i_{\alpha,j}$ for $1 \leq j < k$. If no such α exists, $f_k(X \upharpoonright y)$ is undefined. Then, as one can easily check, f_1 is dense along all sets, f_{k+1} is dense along all sets which meet f_k infinitely often ($1 \leq k < l$), and a set which meets f_l meets the extension function f, too. Since a p-generic set G meets any simple n^{c+1}-extension function which is dense along G not just once but infinitely often (see [1], Proposition 6.11), the above implies that any p-generic set G will meet f. □

3 Separating NP-Completeness Notions of Bounded Query Reducibilities

In this section we compare the **NP**-completeness notions induced by the bounded query reducibilities of fixed norm under the genericity assumption (G). We first consider the case of nonadaptive reductions. Recall that a *polynomial-time bounded-truth-table reduction of norm k* (*P-btt*(k)*-reduction* for short) of a set A to a set B is given by polynomial-time functions $h : \Sigma^* \to \Sigma$ (*evaluator)* and $g_1, \ldots, g_k : \Sigma^* \to \Sigma^*$ (*selectors)* such that $A(x) = h(x, B(g_1(x)), \ldots, B(g_k(x)))$ for all x. We write $A \leq^P_{btt(k)} B$ if there is a *P-btt*(k)-reduction from A to B, and a set $B \in \mathbf{NP}$ is $\mathbf{NP}$-*btt*(k)-*complete* if $A \leq^P_{btt(k)} B$ for all $A \in \mathbf{NP}$.

Theorem 3.1 *Assume (G) and let* $k \geq 2$. *There is an* $\mathbf{NP}$-*btt*$(k+1)$-*complete set* A *which is not* $\mathbf{NP}$-*btt*(k)-*complete.*

Proof. Let $G \in \mathbf{NP}$ be p-generic and let C be an $\mathbf{NP}$-m-complete set such that $C \in DTIME(2^n)$. Before going into details we want to give the idea of the proof. Note that p-generic sets are designed to share the properties that can be forced by diagonalization strategies within the range of given bounds. This can be viewed as "built-in diagonalizations" in the p-generic sets: every describable diagonalization strategy with the given resources will succeed.

Therefore we will consider a set $\widehat{G}$ that is *P-btt*$(k-1)$-reducible to the p-generic set G. The set A will be the disjoint union of the $k-1$ parts of G that determine $\widehat{G}$ and of the sets $\widehat{G} \cap C$ and $\widehat{G} \cup C$. By this construction A will be $\mathbf{NP}$-*btt*$(k+1)$-complete, since membership in $\widehat{G}$ can be detected by $k-1$ queries and membership in C can be detected by two extra queries. Allowing only k queries will cause some lack of information, either about $\widehat{G}$ or about the relation between $\widehat{G}$ and C. This fact will be used in the proof of A not being $\mathbf{NP}$-*btt*(k)-complete.

There we will take a subproblem B of G which is independent of the parts of G used in the definition of A and, for a contradiction, we will assume that a *P-btt*(k)-reduction from B to A is given. Then, exploiting the "lack of information" in this reduction, there will be a strategy forcing A and B to disagree on the reduction.

Formally this will be achieved by defining a bounded extension function f describing the diagonalization strategy, and, by G meeting f, we will argue that this strategy will succeed. Since the diagonalization strategy has to pay attention to all possible kinds of missing information, the definition of f will require a careful and somewhat tedious distinction of cases.

The desired set A is defined as follows. Let $z_1, \ldots, z_{k+1}$ be the first $k+1$ strings of length k, let

$$\widehat{G}_m = \{x : xz_m \in G\} \quad (1 \leq m \leq k)$$

and let

$$\widehat{G} = \bigcup_{m=1}^{k-1} \widehat{G}_m$$

be the union of the first $k-1$ of these sets. Then A is the disjoint union of the $k+1$ sets $\widehat{G}_1, \ldots, \widehat{G}_{k-1}, \widehat{G} \cap C$ and $\widehat{G} \cup C$:

$$A = \bigcup_{m=1}^{k-1} \{xz_m : x \in \widehat{G}_m\} \cup \{xz_k : x \in \widehat{G} \cap C\} \cup \\ \cup \{xz_{k+1} : x \in \widehat{G} \cup C\}.$$

Since G and C are **NP**-sets, it follows by standard closure properties of **NP** that $A \in \mathbf{NP}$, too. Moreover, for any string x,

$$\begin{array}{lllll} x \in C & \Leftrightarrow & x \in \widehat{G} \cap C & \text{or} & [x \notin \widehat{G} \;\&\; x \in \widehat{G} \cup C] \\ & \Leftrightarrow & x \in \widehat{G} \cap C & \text{or} & [\forall 1 \leq i < k\ (x \notin \widehat{G}_i) \;\&\; x \in \widehat{G} \cup C] \\ & \Leftrightarrow & xz_k \in A & \text{or} & [\forall 1 \leq i < k\ (xz_i \notin A) \;\&\; xz_{k+1} \in A] \end{array}$$

whence $C \leq^P_{btt(k+1)} A$. So, by **NP**-m-completeness of C, A is **NP**-$btt(k+1)$-hard.

It remains to show that A is not **NP**-$btt(k)$-complete, which will be the most involved part of the proof. Since $\widehat{G}_k \in \mathbf{NP}$ it suffices to show that $\widehat{G}_k \not\leq^P_{btt(k)} A$. For a contradiction assume that $\widehat{G}_k \leq^P_{btt(k)} A$ via $(h, g_1, \ldots, g_k)$, i.e.

(10) $\widehat{G}_k(x) = h(x, A(g_1(x)), \ldots, A(g_k(x)))$

for all strings x.

In the following we will use p-genericity of G to refute (10). We will define a bounded extension function f such that f will satisfy the hypothesis of Lemma 2.9 and such that (10) will fail for $x = 0^n$ if G will meet f at 0^{n+k}.

For this sake, given n and $X \restriction 0^{n+k}$ we will define a set $COND = \{(y_l, i_l) : l \leq m\}$ (where $0^{n+k} \leq y_0 < y_1 < \ldots < y_m$) of *forcing conditions* in such a way that, firstly,

(11) $X \restriction 0^{n+k} = G \restriction 0^{n+k} \;\&\; \forall (y,i) \in COND(G(y) = i)$

will imply

(12) $\widehat{G}_k(0^n) \neq h(0^n, A(g_1(0^n)), \ldots, A(g_k(0^n)))$

and, secondly, $|COND| \leq c$ for some constant c (not depending on $X \restriction 0^{n+k}$ and n) and there are uniform procedures for computing the set $\{y_l : l \leq m\}$ of the positions of the forcing conditions in $O(2^n)$ steps and for computing the value i of some forcing condition (y, i) in $O(2^{|y|})$ steps.

Then, by letting

$$f(X \restriction 0^{n+k}) = (y_0, i_0), \ldots, (y_m, i_m),$$

the latter ensures that f fulfills the premises of Lemma 2.10. Moreover, G meeting f at 0^{n+k} implies (11). So, by p-genericity of G, (12) will hold for some n.

In the remainder of the proof we define the condition set $COND$ for given n and $X\lceil 0^{n+k}$ and we show that it has the desired properties. Since $COND$ will be chosen so that (11) ensures (12), the motivating remarks on the definition of $COND$ are made under the asumption that (11) holds. So, by $G\lceil 0^{n+k} = X\lceil 0^{n+k}$, for strings y with $|y| < n+k$ the value of $G(y)$ is determined by $X\lceil 0^{n+k}$. For strings y with $|y| \geq n+k$ and for $i \leq 1$ we can assume that $G(y) = i$ can be forced by adding the condition (y, i) to $COND$. I.e. the set $COND$ should be viewed as a tool allowing us to fix the values of G on a constant number of strings of length at least $n+k$.

Using the dependence of $\widehat{G}_k$ and A on G, by forcing values for G we can also force values for $\widehat{G}_k$ and, to a certain extent, for A. E.g., since $\widehat{G}_k(0^n) = G(0^n z_k)$ the condition $(0^n z_k, i)$ forces $\widehat{G}_k(0^n) = i$ (for $i = 0, 1$).

We will use this observation to force values $A(w) = \alpha(w)$ for the queries

$$w \in QUERY = \{g_1(0^n), \ldots, g_k(0^n)\}$$

on the right side of (12). Then we can compute

$$\textit{(13)}\quad j_0 = h(0^n, \alpha(g_1(0^n)), \ldots, \alpha(g_k(0^n)))$$

whence, by adding the condition $(0^n z_k, 1-j_0)$ to $COND$ we can force $\widehat{G}_k(0^n) = 1 - j_0$ thereby satisfying (12).

Since the dependence of A on G is less straightforward and since, in fact, A does not only depend on G but also on the complete set C, next we will discuss to what extent values for A can be forced. This requires a closer look at the definition of A. Corresponding to the parts $\widehat{G}_1, \ldots, \widehat{G}_{k-1}$, $\widehat{G} \cap C$ and $\widehat{G} \cup C$ of which A is composed, a string $w \in A$ is of the form vz_p, $1 \leq p \leq k+1$, where p indicates that membership of w in A is determined by the pth component. For $w = vz_p$ we call v the *value* and p the *index* of w. Then the relation between $A(w)$ and G depends on the index of w as follows

$$\textit{(14)}\qquad 1 \leq p \leq k-1 \;\Rightarrow\; A(vz_p) = G(vz_p)$$

$$\textit{(15)}\qquad \begin{aligned} vz_k \in A \;&\Leftrightarrow\; v \in C \;\&\; \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in A) \\ &\Leftrightarrow\; v \in C \;\&\; \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in G) \end{aligned}$$

$$\textit{(16)}\qquad \begin{aligned} vz_{k+1} \in A \;&\Leftrightarrow\; v \in C \;\vee\; \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in A) \\ &\Leftrightarrow\; v \in C \;\vee\; \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in G) \end{aligned}$$

$$\textit{(17)}\qquad p > k+1 \;\Rightarrow\; vz_p \notin A$$

By (14) we can force $A(w) = i$ (for $i = 0, 1$) for a string w of index $p \in \{1, \ldots, k-1\}$ by the condition (w, i). If the index of w is k then we can only control the second conjunct on the right side of (15): taken together the conditions $(vz_1, 0), \ldots, (vz_{k-1}, 0)$ will ensure that this conjunct fails whence $A(vz_k) = 0$. By the dependence on C, however, here in general it is impossible to force $A(vz_k) = 1$. Dually, by (16), for w with index $k+1$, adding $(vz_l, 1)$ to $COND$

for some $l \in \{1, \ldots, k-1\}$ will force $A(vz_{k+1}) = 1$, whereas $A(vz_{k+1}) = 0$ in general cannot be forced. So, for queries $w, w' \in QUERY$ with index k and $k+1$, we have to force $A(w) = 0$ and $A(w') = 1$, respectively.

For strings $w = vz_k$ and $w' = vz_{k+1}$ of index k and $k+1$ with the same value v the above strategies for forcing $A(w) = 0$ and $A(w') = 1$ are not compatible. In order to force these values here, we have to know the value of $C(v)$: If $C(v) = 0$ then $A(w) = 0$ by failure of the first conjunct, whence we can force $A(w') = 1$ as above. Dually, for $C(v) = 1$, $A(w') = 1$ is immediate by the first disjunct in (16) whence here it suffices to force $A(w) = 0$. For the following it will be important to note that if this situation applies to strings $w, w' \in QUERY$ and we have to know $C(v)$ in order to decide whether some string xz_l, $1 \leq l \leq k-1$, has to be forced into G (and thereby into A!) then we can choose l so that $xz_l \notin QUERY$. Namely, since vz_k and vz_{k+1} are both among the k queries at least one of the $k-1$ strings $xz_1, \ldots, xz_{k-1}$ will not be queried. (This will guarantee that the values $\alpha(w)$ for $w \in QUERY$ can be computed in $O(2^n)$ steps, whence, by (13), the condition $(0^n z_k, 1-j_0)$ forcing the desired disagreement in (12) will obey the required time bounds for $COND$.)

Next we define the function $\alpha : QUERY \rightarrow \{0,1\}$ specifying the intended values for A on these queries. Fix $w = vz_p \in QUERY$. For the definition of $\alpha(w)$ we distinguish between short and long strings w. If $|w| < n+k$ let

$$\text{(18)}\quad \alpha(w) = \begin{cases} X(w) & \text{if } 1 \leq p \leq k-1 \\ 1 & \text{if } [p = k \;\&\; v \in C \;\&\; \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in X)] \\ & \quad \text{or} \\ & [p = k+1 \;\&\; (v \in C \vee \exists l(1 \leq l \leq k-1 \;\&\; vz_l \in X))] \\ 0 & \text{otherwise} \end{cases}$$

It is easy to check that in this case the hypothesis $X \upharpoonright 0^{n+k} = G \upharpoonright 0^{n+k}$ in (11) implies $A(w) = \alpha(w)$ by (14)–(17). For w with $|w| \geq n+k$ let

$$\text{(19)}\quad \alpha(w) = \begin{cases} 1 & \text{if } p = k+1 \\ & \quad \text{or} \\ & [p = k-1 \;\&\; \forall l \in \{1, \ldots, k-1, k+1\}(vz_l \in QUERY)] \\ 0 & \text{otherwise} \end{cases}$$

Here, assuming (11), $A(w) = \alpha(w)$ will be ensured by the definition of the set $COND$ of forcing conditions which we will give next.

For any value v, $|v| \geq n$, such that there is a query $vz_p \in QUERY$ for some p with $1 \leq p \leq k+1$, we will define a set $COND_v$ of forcing conditions which will be part of $COND$. Given such a value v let

$$IND(v) = \{p : 1 \leq p \leq k+1 \;\&\; vz_p \in QUERY\}$$

be the set of indices of queried strings with this value. The definition of $COND_v$ depends on this index set and on the parameter r defined by

$$r = \mu s \geq 1(s \notin IND(v) \;\vee\; s = k-1)$$

as follows:

(20) $k+1 \notin IND(v) \quad \Rightarrow \quad COND_v = \{(vz_1, 0), \ldots, (vz_{k-1}, 0)\}$

(21) $k+1 \in IND(v) \quad \& \quad k \notin IND(v)$
$\Rightarrow \quad COND_v = \{(vz_i, j) : 1 \le i \le k-1 \;\&\; (j = 1 \Leftrightarrow i = r)\}$

(22) $k+1 \in IND(v) \quad \& \quad k \in IND(v)$
$\Rightarrow \quad COND_v = \{(vz_i, j) :$
$1 \le i \le k-1 \;\&\; (j = 0 \text{ if } i \ne r) \;\&\; (j = 1 - C(v) \text{ if } i = r)\}$

Note that in case of (22), $r \notin IND(v)$, i.e., $vz_r \notin QUERY$, since, by $|IND(v)| \le |QUERY| = k$, $k \in IND(v)$ and $k+1 \in IND(v)$ imply that $s \notin IND(v)$ for some $s \in \{1, \ldots, k-1\}$.

To show that, assuming (11), this part of the definition of $COND$ ensures that $A(w) = \alpha(w)$ for strings $w \in QUERY$ with $|w| \ge n+k$, fix such a string $w = vz_p$ and distinguish the following cases depending on the index of x.

Case 1: $p < k-1$. Then $\alpha(w) = 0$ and $(vz_p, 0) \in COND_v$ whence $A(w) = 0$ by (11).

Case 2: $p = k-1$ and $IND(v) = \{1, \ldots, k-1, k+1\}$. Then $\alpha(w) = 1$. Moreover, $r = k-1$ whence, by (21), $(vz_{k-1}, 1) \in COND$. So $A(w) = 1$ by (11).

Case 3: $p = k-1$ and $IND(v) \ne \{1, \ldots, k-1, k+1\}$. Then $\alpha(w) = 0$. Moreover, either $k+1 \notin IND(v)$ whence case (20) applies or $r \ne k-1$. So in any case $(vz_{k-1}, 0) \in COND_v$ whence $A(w) = 0$.

Case 4: $p = k$. Then $\alpha(w) = 0$. If $v \notin C$ then $A(w) = 0$ is immediate by (15). So assume $v \in C$, i.e., $1 - C(v) = 0$. Since for the definition of $COND_v$ (20) or (22) applies, it follows that $COND_v = \{(vz_1, 0), \ldots, (vz_{k-1}, 0)\}$ whence $G(vz_l) = 0$ for all $l \in \{1, \ldots, k-1\}$ by (11). Hence $A(w) = 0$ by (15).

Case 5: $p = k+1$. Then $\alpha(w) = 1$. If $v \in C$ then $A(w) = 1$ is immediate by (16). So assume $v \notin C$, i.e., $1 - C(v) = 1$. Since case (21) or (22) applies, $(vz_r, 1) \in COND_v$ whence $vz_r \in G$ by (11). It follows that $A(w) = 1$ by (16).

This completes the proof that, assuming (11), the above defined parts of $COND$ force $A(w) = \alpha(w)$ for all queries w. So, by adding $(0^{n+k}, 1 - j_0)$ for j_0 defined by (13) as final condition, (11) will imply (12). Hence it only remains to show that the condition set satisfies the bounds specified above.

For this sake, first observe that $|COND| \le k^2$ since besides the last condition $(0^{n+k}, 1 - j_0)$ forcing the desired value of the left side of (12), for any of the at most k different values v attained by some query in $QUERY$, $k-1$ conditions were added to $COND$. Moreover, the positions $y_0, \ldots, y_m$ of the conditions can be computed in $poly(n)$ steps. Finally, given $X \restriction 0^{n+k}$, for the computation of the value i of a condition (y, i) $O(2^{|y|})$ steps suffice: If (y, i) is added to $COND$ via (20), (21) or (22) then obviously $poly(|y|)$ steps suffice unless $(y, i) =$

$(vz_r, 1-C(v))$ in case (22) where the claim follows from $C \in DTIME(2^n)$. This leaves the final condition $(0^{n+k}, 1-j_0)$. Here, by (13), it suffices to show that $\alpha(w)$ for $w \in QUERY$ can be computed in $O(2^n)$ steps. But for $\alpha(w)$ defined by (18) this follows by $|w| < n+k$ and $C \in DTIME(2^n)$ while for $\alpha(w)$ defined by (19) obviously $\alpha(w)$ is computable in $poly(n)$ steps.
This completes the proof of Theorem 3.1. □

Next we will prove the analog of Theorem 3.1 for the adaptive reducibilities. A *polynomial-time bounded Turing reduction of norm* k ($P-bT(k)$-*reduction* for short) is a polynomial-time bounded Turing reduction M in which for any oracle set X and any input x the number of oracle queries is bounded by k. The queries may depend on the oracle, i.e. on the answers of the previous queries, whence the computation or query tree of $M(x)$ where the nodes are labelled with the queries has depth $\leq k-1$ but may have size $2^k - 1$. We say that A is $P-bT(k)$-*reducible* to B ($A \leq^P_{bT(k)} B$) via M if $A = M^B$ for the $P-bT(k)$-reduction M, and $B \in \mathbf{NP}$ is $\mathbf{NP}$-$bT(k)$-*complete* if $A \leq^P_{bT(k)} B$ for all $A \in \mathbf{NP}$.

Obviously,

(23) $A \leq^P_{btt(k)} B \Rightarrow A \leq^P_{bT(k)} B$

and, by the above remark on the size of the query tree of a $P-bT(k)$-reduction,

(24) $A \leq^P_{bT(k)} B \Rightarrow A \leq^P_{btt(2^k-1)} B$,

and both implications are optimal (see [13]).

In order to distinguish $\mathbf{NP}$-$bT(k+1)$-completeness from $\mathbf{NP}$-$bT(k)$-completeness assuming (G), we show that the set A constructed in the proof of Theorem 3.1 is $\mathbf{NP}$-$bT(k)$-complete but not $\mathbf{NP}$-$bT(k-1)$-complete. Together with Theorem 3.1 this implies the following stronger result.

Theorem 3.2 *Assume* (G) *and let* $k \geq 2$. *There is a set* A *which is* $\mathbf{NP}$-$bT(k)$-*complete and* $\mathbf{NP}$-$btt(k+1)$-*complete but neither* $\mathbf{NP}$-$bT(k-1)$-*complete nor* $\mathbf{NP}$-$btt(k)$-*complete.*

Proof. Fix A, C and G as in the proof of Theorem 3.1. It remains to show that A is $\mathbf{NP}$-$bT(k)$-hard but not $\mathbf{NP}$-$bT(k-1)$-hard.

For a proof of the former note that, by definition of A,

$$\begin{aligned} x \in C \quad &\Leftrightarrow \quad [x \in \widehat{G} \;\&\; x \in \widehat{G} \cap C] \text{ or } [x \notin \widehat{G} \;\&\; x \in \widehat{G} \cup C] \\ &\Leftrightarrow \quad [A \cap \{xz_l : 1 \leq l \leq k-1\} \neq \emptyset \;\&\; xz_k \in A] \\ &\qquad \text{or} \\ &\qquad [A \cap \{xz_l : 1 \leq l \leq k-1\} = \emptyset \;\&\; xz_{k+1} \in A] \end{aligned}$$

whence $C \leq^P_{bT(k)} A$. By $\mathbf{NP}$-m-completeness of C this implies that A is $\mathbf{NP}$-$bT(k)$-hard. The proof that A is not $\mathbf{NP}$-$bT(k-1)$-hard is similar to the proof that A fails to be **NP**-$btt(k)$-complete. Hence we will only sketch the proof where we will use the notation introduced there.

Given a P-$bT(k-1)$-reduction M it suffices to show that

(25) $\widehat{G}_k(0^n) \neq M^A(0^n)$

for some n. As in the proof of Theorem 3.1, given a number n and an initial segment $X\upharpoonright 0^{n+k}$ it suffices to define a set $COND = \{(y_l, i_l) : l \leq m\}$ of forcing conditions such that $0^{n+k} \leq y_0 < y_1 < \ldots < y_m$, the sizes of the sets $COND$ are unifomly bounded by a constant c, the forcing locations $\{y_0, \ldots, y_m\}$ and the forced values i_l are uniformly computable in $O(2^n)$ and $O(2^{|y_l|})$ steps, respectively, and to ensure that the assumption (11) will imply (25).

As in the proof of Theorem 3.1 this is achieved by first defining a function α specifying the intended A-values for the oracle queries in $M^A(0^n)$ where α will be computable in $O(2^n)$ steps. Then, by appropriate forcing conditions it will be ensured that, assuming (11), the intended values are actually attained for some n whence $M^\alpha(0^n) = M^A(0^n)$. Hence by adding a condition forcing $\widehat{G}_k(0^n) = 1 - M^\alpha(0^n)$ the desired diagonalization (25) is achieved.

For the definition of the function $\alpha : \Sigma^* \rightarrow \{0,1\}$ we distinguish between short and long input strings. For w with $|w| < n+k$, $\alpha(w)$ is defined by (18) and, for $w = vz_p$ with $|w| \geq n+k$ we let $\alpha(w) = 1$ iff the index p of w is $k+1$.

Next, by simulating M on input 0^n define the set $QUERY$ of the queries asked by M if the previous queries were answered by $\alpha(w)$, and let $M^\alpha(0^n)$ denote this computation.

Note that $|QUERY| \leq k-1$ since M is $(k-1)$-bounded and, as one can easily show,

(26) $j_0 = M^\alpha(0^n)$

can be computed in $O(2^n)$ steps by definition of α.

In order to ensure that, assuming (11), $M^A(0^n) = M^\alpha(0^n)$ and $\widehat{G}_k(0^n) = 1 - M^\alpha(0^n)$ we define the set $COND$ of forcing conditions as in the proof of Theorem 3.1 (using the set $QUERY$ defined above and j_0 as in (26)). For the proof of correctness it is crucial to note that now $|QUERY| \leq k-1$ whence not only in (22) but also in (21) $r \notin IND(v)$, i.e., the string forced into A there is not a member of the set $QUERY$. Hence, assuming (11), for $w \in QUERY$ with $|w| \geq n+k$, $A(w) = \alpha(w) = 0$ for strings of index $\leq k-1$ is immediate while, for strings of index k or $k+1$, $A(w) = \alpha(w)$ is shown as in the proof of Theorem 3.1. □

Corollary 3.3 *Assume (G) and let $k \geq 1$. There is an* **NP**-$bT(k+1)$*-complete set which is not* **NP**-$bT(k)$*-complete.*

Corollary 3.4 *Assume (G) and let $k \geq 2$. There is an* **NP**-$btt(k+1)$*-set which is not* **NP**-$bT(k-1)$*-complete.*

While the separation for the bounded Turing reducibilities above is optimal, the corresponding Theorem 3.1 for the truth-table reducibilities leaves a small gap for $k = 1$:

Question 1. Assuming (G), is there an $\mathbf{NP}$-$btt(2)$-complete set which is not $\mathbf{NP}$-$btt(1)$-complete?

Also it remains the question whether the comparison of bounded truth-table and bounded Turing completeness in Corollary 3.4 can be further improved to obtain optimal bounds:

Question 2. Assume (G) and let $k \geq 1$. Is there an $\mathbf{NP}$-$btt(k+1)$-complete set which is not $\mathbf{NP} - bT(k)$-complete?

Under the stronger hypothesis that there is a p-generic set G in $\mathbf{NP} \cap co\mathbf{NP}$ we can give affirmative answers to these questions. This stronger hypothesis allows the construction of witness sets A for the separations which also use negative information on G. This greatly simplifies the above proofs where the access of the constructed set A to the generic set G was only positive in order to guarantee membership of A in $\mathbf{NP}$.

For the comparison of bounded truth-table and bounded Turing completeness in the other direction we can prove a bound which is optimal by (24). Here again the lack of direct access to negative information on the generic set G requires a quite involved definition of the witness set.

Theorem 3.5 *Assume (G) and let $k \geq 2$. There is an $\mathbf{NP}$-$bT(k)$-complete set A which is not $\mathbf{NP}$-$btt(2^k - 2)$-complete.*

Proof. The basic format of the proof is the same as that of Theorem 3.1. As there we fix an $\mathbf{NP}$-m-complete set $C \in DTIME(2^n)$ and a p-generic set $G \in \mathbf{NP}$. From these sets we define a set A of which we will show that it has the desired properties. Again, there will be a part of A for which we can force $A(w) = i$ by fixing $G(v) = i$ for some string v and $i \leq 1$, a second part for which we can force $A(w) = 0$ and a third part for which we can force $A(w) = 1$ by fixing G on a constant number of strings.

For the definition of A we need the following notation. For any string y let

$$I_i(y) = \{z : zi \sqsubseteq y\} \qquad (i = 0, 1)$$

be the set of proper initial segments z of y such that the i-extension zi is still extended by y. Note that $I_0(y)$ and $I_1(y)$ are disjoint and the union of these sets consists of all proper initial segments of y, whence $|I_0(y)| + |I_1(y)| = |y|$.

Using this notion we define sets $C_{\&}$ and $C_{\vee}$ depending on C and G as follows:

$$C_{\&} = \{\overline{x}y : |y| = k - 2 \,\&\, [x \in C \,\&\, \overline{x}y \in G \,\&\, \forall z \in I_1(y)(\overline{x}z \in G)]\}$$

$$C_{\vee} = \{\overline{x}y : |y| = k - 2 \,\&\, [x \in C \,\vee\, \overline{x}y \in G \,\vee\, \exists z \in I_0(y)(\overline{x}z \in G)]\}$$

where $\overline{x}y$ denotes the coded pair (x, y) defined by $\overline{x}y = 1^{|x|+1}0xy$. (For the following note that $\overline{x}y \in \Sigma^* \setminus \{0\}^*$ and, for $|y| = k-2$, $|\overline{x}y| = 2|x| + k$.)

Then the desired set A is defined as the disjoint union of $G^- = G \setminus \{0\}^*$, $C_\&$ and $C_\vee$:

$$A = \{v\underline{1} : v \in G^-\} \cup \{v\underline{2} : v \in C_\&\} \cup \{v\underline{3} : v \in C_\vee\}$$

where $\underline{1}$, $\underline{2}$, $\underline{3}$ are the first three strings of length 2. By standard closure properties of **NP**, $A \in$ **NP** whence it suffices to show that A is **NP**-$bT(k)$-hard but not **NP**-$btt(2^k - 2)$-hard.

In order to do so we first point out some basic relations among the parts of A and the sets C and G. For any string x let y_x be the unique string of length $k-2$ satisfying

(27) $\forall z \sqsubset y_x \quad (\overline{x}z \in G \Leftrightarrow z1 \sqsubseteq y_x)$

Then $\overline{x}z \in G$ for all $z \in I_1(y_x)$ and $\overline{x}z \notin G$ for all $z \in I_0(y_x)$ whence, by definition of $C_\&$ and $C_\vee$,

(28) $\overline{x}y_x \in G \Rightarrow C(x) = C_\&(\overline{x}y_x) \;\&\; C_\vee(\overline{x}y_x) = 1$

and

(29) $\overline{x}y_x \notin G \Rightarrow C(x) = C_\vee(\overline{x}y_x) \;\&\; C_\&(\overline{x}y_x) = 0.$

For strings $y \neq y_x$ of length $k-2$ we have

(30)
$$\begin{array}{lll} y <_L y_x & \Rightarrow & C_\vee(\overline{x}y) = 1 \\ y_x <_L y & \Rightarrow & C_\&(\overline{x}y) = 0 \end{array}$$

(where $u <_L v$ denotes that u is to the left of v, i.e., $u = w_0 0 w_1$ and $v = w_0 1 w_2$ for some strings w_0, w_1, w_2). This follows from the fact that, by definition of y_x, for the longest common initial segment z of y and y_x, $G(\overline{x}z) = 1$ and $z \in I_0(y)$ if $y <_L y_x$ while $G(\overline{x}z) = 0$ and $z \in I_1(y)$ if $y_x <_L y$. Finally note that, for any string y of length $k-2$,

(31) $C_\&(\overline{x}y) \leq G(\overline{x}y) \leq C_\vee(\overline{x}y)$.

Now, for a proof of **NP**-$bT(k)$-hardness of A, we show that the **NP**-m-complete set C is P-$bT(k)$-reducible to A as follows. Given x, first by $k-2$ adaptive queries to the G^--part of A produce the string $y_x = y_x(0) \dots y_x(k-3)$ satisfying (27) by letting $y_x(l) = G(\overline{x}(y_x \lceil l))$ for $l \leq k-3$. Then, by (28) and (29), a $(k-1)$th query to the G^--part of A will tell how a final query to the $C_\&$-part respectively $C_\vee$-part will give the desired answer.

In the remainder of the proof we will demonstrate that $G \not\leq^P_{btt(2^k-2)} A$ whence A is not **NP**-$btt(2^k - 2)$-hard. Given a p-$btt(r)$-reduction $(h, g_1, \dots, g_r)$ with $r \leq 2^k - 2$ it suffices to show that

(32) $G(0^n) \neq h(0^n, A(g_1(0^n)), \dots, A(g_r(0^n)))$

for some n. As in the two preceding proofs this can be established by defining, for any given number n and any initial segment $X \upharpoonright 0^n$, a set $COND = \{(y_l, i_l) : l \leq m\}$ of forcing conditions such that $0^n \leq y_0 < y_1 < \ldots < y_m$ and the following properties hold: The size of $COND$ is uniformly bounded by a constant, there are uniform procedures for computing the set $\{y_l : l \leq m\}$ in $O(2^n)$ steps and for computing i_l from y_l in $O(2^{|y_l|})$ steps, and, finally, the condition (11) will imply that (32) holds.

Fix n and $X \upharpoonright 0^n$. For $QUERY = \{g_1(0^n), \ldots, g_r(0^n)\}$ we define a function $\alpha : QUERY \to \{0, 1\}$ such that, assuming (11), $A(w) = \alpha(w)$ will either hold by the first part of (11) or will be forced by the definition of $COND$ and the second part of (11). Moreover, α will be computable in $O(2^n)$ steps. Hence adding the condition $(y, i) = (0^n, 1 - h(0^n, \alpha(g_1(0^n)), \ldots, \alpha(g_r(0^n))))$ to $COND$ will ensure that (32) holds, and i will be computable in $O(2^n) = O(2^{|y|})$ steps as required above.

For the definition of α and $COND$ we have to distinguish the different types of elements of A. A string w is called *relevant* if $w = \overline{x}y\underline{j}$, $j \in \{1, 2, 3\}$, $|y| \leq k - 2$, and $|y| = k - 2$ if $j \in \{2, 3\}$. And w is called *simple* if $w = v\underline{1}$, $v \notin \{0\}^*$, and w is not relevant. For a relevant string $w = \overline{x}y\underline{j}$, $1 \leq j \leq 3$, j is called the *index* of w and x is called the *value* of w.

Note that any element of A is relevant or simple. For a relevant string $w = v\underline{1}$ with index 1 or a simple string, $A(w) = G(v)$ whence, assuming (11), we can force $A(w) = i$ ($i \in \{0, 1\}$) by the condition (v, i) if $|v| \geq n$, i.e., $|w| \geq n + 2$.

For a relevant string $w = \overline{x}y\underline{2}$ with index 2, $A(w) = C_{\&}(\overline{x}y)$, and the latter conjunctively depends on $C(x)$ and $G(\overline{x}z)$ for some $z \sqsubseteq y$. By the conjunctive dependence on C, manipulating G only suffices to force $A(w) = 0$. Similarly, for relevant strings with index 3, by disjunctivity of the dependence only $A(w) = 1$ can be forced. Moreover, forcing these values will require to fix G on certain strings with value x, whence forcing the values of A for strings with the same value has to be coordinated.

Based on these observations we define α and the parts of $COND$ forcing A to agree with α as follows. For any irrelevant, non-simple string $w \in QUERY$, $\alpha(w) = 0$, and, for any simple $w = v\underline{1} \in QUERY$ let $\alpha(w) = (X \upharpoonright 0^n)(v)$ if $|v| < n$ and let $\alpha(w) = 0$ if $|v| \geq n$. Moreover, in the latter case add the condition $(v, 0)$ to $COND$. Note that, assuming (11), in any of the above cases this ensures that $A(w) = \alpha(w)$.

For the remaining cases let x be a string such that there is a relevant query w in $QUERY$ with value x. Then $\alpha(w)$ is simultaneously defined for all such strings w and a part $COND_x$ of $COND$ is specified which, assuming (11), guarantees the correctness of $\alpha(w)$. The definition depends on the length of x.

Case 1: $2|x| + k < n$. Then, for any relevant string $w = \overline{x}y\underline{j}$ with value x, $|\overline{x}y| < n$ whence for $w \in QUERY$ the definition of $\alpha(w)$ can be based on the initial segment $X\lceil 0^n$ as follows:

$$\alpha(\overline{x}y\underline{j}) = \begin{cases} X(\overline{x}y) & \text{if} \quad j = 1 \\ C_{\&}^X(\overline{x}y) & \text{if} \quad j = 2 \\ C_{\vee}^X(\overline{x}y) & \text{if} \quad j = 3 \end{cases}$$

where $C_{\&}^X$ and $C_{\vee}^X$ are defined as $C_{\&}$ and $C_{\vee}$, respectively, with $X\lceil 0^n$ in place of G. Moreover, we let $COND_x = \emptyset$.

In this case, $A(w) = \alpha(w)$ by the first part of (11) and $\alpha(w)$ can be computed in $O(2^n)$ steps by $C \in DTIME(2^n)$.

Case 2: $2|x| + k \leq n + k$. Then let $COND_x$ consist of the conditions $(\overline{x}y, 0)$ for all strings y with $|y| \leq k - 2$ and $|\overline{x}y| \geq n$ and let $X\lceil 0^{n+k}$ be the extension of $X\lceil 0^n$ with $X(z) = 0$ for all strings z with $0^n \leq z \leq 0^{n+k}$. For the relevant strings $w \in QUERY$ with value x, $\alpha(w)$ is defined as in Case 1 with $X\lceil 0^{n+k}$ replacing $X\lceil 0^n$.

Note that, by the second part of (11), $COND_x$ forces $G(\overline{x}y) = 0$ for all $\overline{x}y$ of length $\geq n$ which are relevant for the value of $A(w)$ for the above queries w. So computing $\alpha(w)$ requires information on C only for the string x and $|x| \leq n$ whence $\alpha(w)$ can be computed in $O(2^n)$ steps.

Case 3: $n + k < 2|x| + k$. Then, for any string $v = \overline{x}z$ with $|z| \leq k - 2$, $|v| \geq n$ whence $G(v) = i$ $(i \in \{0, 1\})$ can be forced by adding the condition (v, i) to $COND_x$. We will force these values in such a way that for all strings y of length $k-2$, $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$, whence $A(w) = 0$ and $A(w) = 1$ for relevant strings w with value x which have index 2 and 3. Hence, correspondingly, we let $\alpha(w) = 0$ and $\alpha(w) = 1$, respectively, for those of these strings w which are queried. For the definition of $COND_x$ and α for the index-1 strings with value x we distinguish the following cases.

Case 3.1: $\exists z \ (|z| \leq k - 2 \ \& \ \overline{x}z\underline{1} \notin QUERY)$. Then let z_0 be the least such string and let

$$\begin{aligned} COND_x \quad = \quad & \{(\overline{x}z, 0) \ : \ |z| \leq k - 2 \ \& \ z <_L z_0 1\} \\ & \cup\{(\overline{x}z, 1) \ : \ |z| \leq k - 2 \ \& \ z_0 0 <_L z\} \\ & \cup\{(\overline{x}z, i) \ : \ i \leq 1 \ \& \ z \in I_i(z_0)\} \\ & \cup\{(\overline{x}z_0, 1 - C(x))\}. \end{aligned}$$

For any relevant string $\overline{x}z\underline{1} \in QUERY$ with value x and index 1, let $\alpha(\overline{x}z\underline{1})$ be the unique number i with $(\overline{x}z, i) \in COND_x$. By (11) this will obviously imply that $\alpha(\overline{x}z\underline{1}) = A(\overline{x}z\underline{1})$. Moreover, by choice of z_0, the last part of $COND_x$ is not used for determining $\alpha(\overline{x}z\underline{1}) = i$ whence this value can be computed in $poly(n)$ steps. To show that $COND_x$ forces the intended values for strings with index 2 and 3 too, fix y with $|y| = k-2$. It suffices to show that, assuming (11), $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$. First observe that, by the third part of $COND_x$, $z_0 \sqsubseteq y_x$ for the unique string y_x satisfying (27), and that the first two parts of $COND_x$ imply that

(33) $\quad (z <_L z_0 \Rightarrow G(\overline{x}z) = 0) \,\&\, (z_0 <_L z \Rightarrow G(\overline{x}z) = 1)$

for all strings z with $|z| \leq k-2$. So if $y <_L z_0$ then, by (33) and (31), $C_{\&}(\overline{x}y) = 0$ and, by $z_0 \sqsubseteq y_x$ and (30), $C_{\vee}(\overline{x}y) = 1$. For $z_0 <_L y$, $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$ are shown similarly. This leaves the case where $z_0 \sqsubseteq y$, for which it is crucial to note that $C(x) \neq G(\overline{x}z_0)$ by the fourth part of $COND_x$. So, if $z_0 = y$ then $y = y_x$ and $C(x) \neq G(\overline{x}y_x)$ whence the claim follows from (28) and (29). Otherwise, $z_0 \sqsubset y$, say $p = |y| - |z_0|$. Here the proof depends on the value of $C(x)$. If $C(x) = 0$ then $C_{\&}(\overline{x}y) = 0$ is immediate by (31). Moreover, by $z_0 \sqsubseteq y_x$ and, by $C(x) = 0$ and by the definition of $COND_x$, $G(\overline{x}z_0 1^l) = 1$ for all $l \leq p$ whence $y_x = z_0 1^p$ and $G(\overline{x}y_x) = 1$. So, either $y <_L y_x$, whence $C_{\vee}(\overline{x}y) = 1$ by (30), or $y = y_x$, whence $C_{\vee}(\overline{x}y) = 1$ by $G(\overline{x}y_x) = 1$ and (31). Finally if $C(x) = 1$ then, by similar arguments, $C_{\vee}(\overline{x}y) = 1$ is immediate and $y_x = z_0 0^p$ and $G(\overline{x}y_x) = 0$. So $y_x <_L y$ or $y_x = y$ whence $C_{\&}(\overline{x}y) = 0$ by (30) or (31), respectively.

Case 3.2: $\forall z\ (|z| \leq k-2 \Rightarrow \overline{x}z\underline{1} \in QUERY)$. Then there are at least $2^{k-1} - 1$ strings with index 1 in $QUERY$. Since

$$2^{k-1} - 1 + 2^{k-2} + 2^{k-2} = 2^k - 1$$

whereas $|QUERY| \leq r \leq 2^k - 2$, for one of the 2^{k-2} strings y of length $k-2$ $\overline{x}y\underline{2} \notin QUERY$ or $\overline{x}y\underline{3} \notin QUERY$. For the following fix y with $|y| = k-2$ and j with $1 < j \leq 3$ minimal such that $\overline{x}y\underline{j} \notin QUERY$. Let

$$\begin{aligned} COND_x \;=\; & \{(\overline{x}z, 0) \,:\, |z| \leq k-2 \,\&\, z <_L y\} \\ & \cup \{(\overline{x}z, 1) \,:\, |z| \leq k-2 \,\&\, y <_L z\} \\ & \cup \{(\overline{x}z, i) \,:\, i \leq 1 \,\&\, z \in I_i(y)\} \\ & \cup \{(\overline{x}y, i) \,:\, (i = 1 \Leftrightarrow j = 2) \,\&\, (i = 0 \Leftrightarrow j = 3)\} \end{aligned}$$

and, for any relevant string $\overline{x}z\underline{1} \in QUERY$ with value x and index 1 let $\alpha(\overline{x}z\underline{1})$ be the unique number i with $(\overline{x}z, i) \in COND_x$. Then, obviously, $\alpha(\overline{x}z\underline{1})$ can be computed in $poly(n)$ steps and, by (11), $A(\overline{x}z\underline{1}) = \alpha(\overline{x}z\underline{1})$. It remains to show that, assuming (11), $COND_x$ forces the intended values for relevant queries with value x and index 2 or 3. Note that the third part of $COND_x$ ensures that $y = y_x$ whence, by the first two parts, for any string y' of length $k-2$,

$$(y' <_L y_x \Rightarrow G(\overline{x}y') = 0) \,\&\, (y_x <_L y' \Rightarrow G(\overline{x}y') = 1)$$

holds. As in Case 3.1 this implies $A(\overline{x}y'\underline{2}) = C_{\&}(\overline{x}y') = 0$ and $A(\overline{x}y'\underline{3} = C_{\vee}(\overline{x}y') = 1$ for $y' \neq y$. Finally, if $y' = y$, then $\overline{x}y\underline{j} \notin QUERY$. So, for $j = 2$, it suffices to show that $A(\overline{x}y\underline{3}) = C_{\vee}(\overline{x}y) = 1$. In this case the fourth part of $COND_x$ consists of the condition $(\overline{x}y, 1)$ whence $G(\overline{x}y) = 1$. So the claim follows from (31). Similarly, if $j = 3$, then the final condition of $COND_x$ is $(\overline{x}y, 0)$ whence $G(\overline{x}y) = 0$ and therefore $A(\overline{x}y\underline{2}) = C_{\&}(\overline{x}y) = 0$ by (31). This completes the definition of α. The condition set $COND$ consists of the parts specified above together with the final condition

$$(0^n, 1 - h(\alpha(g_1(0^n)), \ldots, \alpha(g_r(0^n))))\,.$$

Then, assuming (11), $COND$ forces A to agree with α on the query set $QUERY$ whence, by the final condition, (32) holds. The proof of the required bounds on $COND$ is straightforward by the above remarks. In particular, $|COND| \leq 2^{2k-1}$ since for any of the $r \leq 2^k - 2$ queries at most $2^{k-1} - 1$ conditions are added to $COND$.
This completes the proof of Theorem 3.5. □

4 Further Separation Results

In the preceding section we gave separations of the **NP**-completeness notions for the bounded query reducibilities of fixed norm (under the assumption (G)). By exploiting the unifomity (in k) of the proof of Theorem 3.1, here we separate **NP**-*btt*-completeness from **NP**-*tt*-completeness. This requires the following diagonalization lemma.

Lemma 4.1 *Let $\mathbf{C}_n$, $n \geq 0$, be uniformly recursively presentable classes which are closed under finite variants, let $D \subseteq \{0\}^* \times \Sigma^*$ be a recursive set such that*

$$D^{[n]} = \{x : \langle 0^n, x\rangle \in D\} \notin \mathbf{C}_n,$$

and let $f : \mathbb{N} \to \mathbb{N}$ be a nondecreasing and unbounded recursive function. Then there exists a set A and a function $g : \mathbb{N} \to \mathbb{N}$ such that

(34) $A \notin \bigcup_{n\geq 0} \mathbf{C}_n$

(35) $\forall\, n(A_{=n} = D^{[g(n)]}_{=n})$

(36) g is polynomial-time computable with respect to the unary representation of numbers.

(37) $\forall n\, (g(n) \leq f(n))$

Note that (35) and (36) imply that $A \leq^P_m D$. So Lemma 4.1 can be viewed as an infinitary version of Schöning's diagonalization lemma in [21].

Proof. The proof is by a standard delayed diagonalization argument similar to the one in [21]. Let U be a recursive presentation of the classes $\mathbf{C}_n$, $n \geq 0$, i.e. $U \subseteq \mathbb{N} \times \mathbb{N} \times \Sigma^*$ is a recursive set such that

$$\mathbf{C}_i = \{U^{[n]}_i : n \geq 0\} \quad \text{where} \quad U^{[n]}_i = \{x \in \Sigma^* : \langle i, n, x\rangle \in U\}.$$

We define the function $h : \mathbb{N} \to \mathbb{N}$ by

$$h(n) = \mu m > n(\forall i \leq n\, \forall l \leq n\, \exists x\, (|x| \in [n, m)\ \&\ D^{[i]}(x) \neq U^{[l]}_i(x)))$$

Then, for $i \leq n$, the length interval $[n, h(n))$ will contain witnesses for the fact that the i-th diagonal set $D^{[i]}$ does not occur under the first n sets in the class $\mathbf{C}_i$. Since the sets D and U are recursive and the classes $\mathbf{C}_i$ are closed under finite variants h will be total recursive. Therefore we may choose a time-constructible function $r \geq h$ and define the intervals

$$I^r(n) := \{x \in \Sigma^* : r^n(0) \leq |x| < r^{n+1}(0)\}$$

where $r^0(0) = 0$, $r^{n+1}(0) = r(r^n(0))$.

Now choose a polynomial-time computable enumeration α of $\mathbb{N}$ in which every number occurs infinitely often and such that $\alpha(n) < f(n)$ for all numbers n. (E.g., given a polynomial-time computable and invertible pairing function $< \cdot, \cdot >$: $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we can define α by letting $\alpha(\langle n, m\rangle) = m$ if the relation $m \leq f(\langle n, m\rangle)$ can be shown in quadratic time by finding a number $k < \langle n, m\rangle$ with $m \leq f(k)$ and by letting $\alpha(\langle n, m\rangle) = 0$ otherwise.)

Finally we define the desired set A by

$$A := \bigcup_{n \geq 0} I^r(n) \cap D^{[\alpha(n)]}$$

For a proof of (34), assume that the claim fails, say $A \in \mathbf{C}_i$, whence $A = U_i^{[k]}$ for some k. Then, by the choice of α, there is an m such that $r^m(0) \geq max\{i, k\}$ and $\alpha(m) = i$. Since $r^{m+1}(0) \geq h(r^m(0))$, by definition of h there exists a string $x \in I^r(m)$ such that $x \in D^{[i]} \Delta U_i^{[k]}$. Since $A(x) = D^{[i]}(x)$ by definition, it follows that $A(x) \neq U_i^{[k]}(x)$ contrary to the assumption.

For a proof of the conditions (35) – (37), note that, by definition of A, there is a unique function $g : \mathbb{N} \to \mathbb{N}$ such that (35) holds. Moreover, since $r(n) > n$, for any number n there is a unique number $s(n) \leq n$ such that $r^{s(n)}(0) \leq n < r^{s(n)+1}(0)$ and, by time constructibility of r, $s(n)$ can be computed in $poly(n)$ steps. Then $g(n) = \alpha(s(n))$, whence (36) holds by polynomial time computability of α. Moreover, by choice of α and by weak monotonicity of f, it follows with $s(n) \leq n$ that

$$g(n) = \alpha(s(n)) \leq f(s(n)) \leq f(n)$$

whence (37) is satisfied, too. □

Proposition 4.2 *A set A is* **NP**-*btt-complete iff A is* **NP**-*btt*(k)-*complete for some $k \geq 1$.*

Proof. For a proof of the nontrivial implication assume that A is **NP**-*btt*-complete and let C be an **NP**-m-complete set. Then $C \leq_{btt}^P A$ whence, by definition, $C \leq_{btt(k)}^P A$ for some k. Hence, for any $B \in$ **NP**, $B \leq_m^P C \leq_{btt(k)}^P A$ by **NP**-m-hardness of C. It follows that $B \leq_{btt(k)}^P A$ whence A is **NP**-*btt*(k)-hard. □

Theorem 4.3 *Assume (G). There is an* **NP**-*tt-complete set A which is not* **NP**-*btt-complete.*

Proof. The set A will be composed of the $btt(k+1)$- but not $btt(k)$-complete sets of Theorem 3.1. Let $\widetilde{C}$ be an **NP**-m-complete set with $\widetilde{C} \in \mathbf{DTIME}(2^n)$ and let $C = \{0^n 1x : n \geq 0 \,\&\, x \in \widetilde{C}\}$ be a padded version of $\widetilde{C}$. Note that C is **NP**-m-complete and $C \in \mathbf{DTIME}(2^n)$, too.

For $k \geq 0$ let A_k be the **NP**-$btt(k+3)$-complete set constructed in the proof of Theorem 3.1 from C as above and some fixed p-generic set G. Then $A_k \in \mathbf{NP}$ uniformly in k, whence $D = \{\langle 0^k, x\rangle : k \geq 0 \,\&\, x \in A_k\} \in \mathbf{NP}$ too. Moreover, for $\mathbf{C}_k = \{B : B \text{ } \mathbf{NP}\text{-}btt(k)\text{-complete}\}$ (for $k \geq 1$ and $\mathbf{C}_0 = \mathbf{C}_1$), the classes $\mathbf{C}_k$ are uniformly recursively presentable and closed under finite variants, and $D^{[k]} = A_k \notin \mathbf{C}_k$. So, by Lemma 4.1, we may fix A and g satisfying (34)$-$(37) for the recursive function $f(n) = \mu m(2m+3 \geq n)$. By (35) and (36), $A \leq_m^P D$ whence $A \in \mathbf{NP}$. Moreover, by (34), A is not **NP**-$btt(k)$-complete for all $k \geq 1$, whence by Proposition 4.2 A is not **NP**-btt-complete.

Finally, for the proof that A is **NP**-tt-hard, it suffices to show that $\widetilde{C} \leq_{tt}^P A$. So fix x and let $|x| = n$. Then, for any numbers $k, m \geq 0$, $\widetilde{C}(x) = C(0^m 1x)$ and, by definition of A_k, $C(0^m 1x)$ can be computed from $(A_k)_{=(m+1+n)+k+2}$ with $k+3$ non-adaptive queries in polynomial time uniformly in x, k and m. On the other hand, by choice of A, f and g, $A_{=2n+3} = (A_{g(2n+3)})_{=2n+3}$ where $g(2n+3) \leq n$ for all numbers n. By the above, $\widetilde{C}(x)$ can be recovered from A as follows: Let $k = g(2n+3)$ and $m = n - k$. Then $A_{=2n+3} = (A_k)_{=2n+3} = (A_k)_{=(m+1+n)+k+2}$ whence $\widetilde{C}(x) = C(0^m 1x)$ can be recovered from $A_{=2n+3}$ with $g(2n+3)+3$ non-adaptive queries. □

5 Open Problems

Assuming Lutz's non-smallness hypothesis *(R)* for **NP** or the weaker category hypothesis *(G)*, we obtained almost complete separation results for the **NP**-completeness notions of the bounded query reducibilities.

The relations among the completeness notions for the standard 1-query reducibilities like $btt(1)$, m and $1(=$ one-one$)$, however, remain open. Since for **EXP** $btt(1)$-, m-, and 1-completeness coincide (see [11] and [7]) and since the hypotheses *(G)* and *(R)* are compatible with **NP** = **EXP**, we cannot use any of these hypotheses for a separation here.

Another interesting open question is the relation between **NP**-T-completeness and **NP**-tt-completeness. The hypotheses *(G)* and *(R)* might not suffice, however, to settle this question. There is some evidence, that *(G)* helps only for bounded query reducibilities (see e.g. Theorem 2.11 in [4]) while *(R)* can deal with unbounded queries (see e.g. Theorem 3.8 in [18]) but (by the Borel-Cantelli-Lemmas) only if the number of queries is growing sufficiently slowly in the input length.

A strengthening of the hypothesis *(G)*, however, namely the assumption that **NP** is not general p-meager in the sense of Definition 2.1, suffices for separating **NP**-T-completeness and **NP**-tt-completeness. This can be seen by an analog construction as in Theorem 3.5.

Along these lines one could be tempted to work with still stronger hypotheses based on the recent category concepts of Fenner [10] and Ambos-Spies ([1], Section 7). As Fenner [10] has shown, however, non-p-meagerness of **NP** for these concepts already implies that **NP** = **EXP**, so that these assumptions are not very plausible.

References

[1] K. Ambos-Spies. Resource-bounded genericity. In Cooper et al., editors, *Computability, Enumerability, Unsolvability: Directions in Recursion Theory*, London Math. Soc. Lect. Notes Series 224, pages 1–59. Cambridge University Press, 1996.

[2] K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizing over deterministic polynomial time. In *Proceedings of the First Workshop on Computer Science Logic, CSL '87*, Lecture Notes in Computer Science 329, pages 1–16, Springer Verlag, 1988.

[3] K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics 187, pages 1–47. Marcel Dekker, 1997.

[4] K. Ambos-Spies, H.-C. Neis, and S. A. Terwijn. Genericity and measure for exponential time. *Theoretical Computer Science*, **168**, 3–19, 1996.

[5] K. Ambos-Spies, S. A. Terwijn, and X. Zheng. Resource bounded randomness and weakly complete problems. *Theoretical Computer Science*, **172**, 195–207, 1997.

[6] M. Bellare and S. Goldwasser. The complexity of decision versus search. *SIAM Journal on Computing*, **23**, 97–119, 1994.

[7] L. Berman. On the structure of complete sets: Almost everywhere complexity and infinitely often speedup. In *Proceedings of the Seventeenth Annual Conference on Foundations of Computer Science*, pages 76–80, IEEE Computer Society Press, 1991.

[8] H. Buhrmann and L. Torenvliet. On the structure of complete sets. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, pages 118–133, IEEE Computer Society Press, 1994.

[9] S. A. Fenner. Notions of resource-bounded category and genericity. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 196–212, IEEE Computer Society Press, 1991.

[10] S. A. Fenner. Resource-bounded category: a stronger approach. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference*, pages 182–192, IEEE Computer Society Press, 1995.

[11] S. Homer, S. Kurtz, and J. Royer. A note on 1-truth-table complete sets. *Theoretical Computer Science*, **115**, 383–389, 1993.

[12] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[13] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, **22**, 155–171, 1975.

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

[15] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, **44**, 220–258, 1992.

[16] J. H. Lutz. The quantitative structure of exponential time. In *Proceedings of the Eighth Annual Conference on Structure in Complexity Theory*, pages 158–175, IEEE Computer Society Press, 1993.

[17] J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity Theory Retrospective, II.* Springer Verlag, 1997.

[18] J. H. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science*, **164**, 141–163, 1996.

[19] S. R. Mahaney. Sparse complete sets for NP: Solution for a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, **25**, 130–143, 1982.

[20] E. Mayordomo. *Contributions to the Study of Resource-Bounded Measure.* PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 1994.

[21] U. Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, **18**, 95–103, 1982.

[22] A. L. Selman. P-selective sets, tally languages, and the behaviour of polynomial time reducibilities on NP. *Mathematical Systems Theory*, **13**, 55–65, 1979.