SIAM J. COMPUT.
Vol. 33, No. 6, pp. 1369–1416

DISJOINT NP-PAIRS*

CHRISTIAN GLASSER†, ALAN L. SELMAN‡, SAMIK SENGUPTA‡, AND LIYU ZHANG‡

Abstract. We study the question of whether the class DisjNP of disjoint pairs (A, B) of NP-sets contains a complete pair. The question relates to the question of whether optimal proof systems exist, and we relate it to the previously studied question of whether there exists a disjoint pair of NP-sets that is NP-hard. We show under reasonable hypotheses that nonsymmetric disjoint NP-pairs exist, which provides additional evidence for the existence of P-inseparable disjoint NP-pairs.

We construct an oracle relative to which the class of disjoint NP-pairs does not have a complete pair; an oracle relative to which optimal proof systems exist, and hence complete pairs exist, but no pair is NP-hard; and an oracle relative to which complete pairs exist, but optimal proof systems do not exist.

Key words. disjoint NP-pairs, promise problems, propositional proof systems, oracles, symmetry

AMS subject classification. 68Q15

DOI. 10.1137/S0097539703425848

1. Introduction.

We study the class DisjNP of disjoint pairs (A, B), where A and B are nonempty, disjoint sets belonging to NP. Such disjoint NP-pairs are interesting for at least two reasons. First, Grollmann and Selman [GS88] showed that the question of whether DisjNP contains P-inseparable disjoint NP-pairs is related to the existence of public-key cryptosystems. Second, Razborov [Raz94] and Pudlák [Pud03] demonstrated that these pairs are closely related to the theory of proof systems for propositional calculus. Specifically, Razborov showed that existence of an optimal propositional proof system implies existence of a complete pair for DisjNP. Primarily in this paper we are interested in the question raised by Razborov [Raz94] of whether DisjNP contains a complete pair. We show connections between this question and earlier work on disjoint NP-pairs, and we exhibit an oracle relative to which DisjNP does not contain any complete pair.

From a technical point of view, disjoint pairs are simply an equivalent formulation of promise problems. There are natural notions of reducibilities between promise problems [ESY84, Sel88] that disjoint pairs inherit easily [GS88]. Hence, completeness and hardness notions follow naturally. We begin in the next section with these definitions, some easy observations, and a review of the known results.

In section 3 we observe that if DisjNP does not contain a Turing-complete disjoint NP-pair, then DisjNP does not contain a disjoint NP-pair all of whose separators are Turing-hard for NP. The latter is a conjecture formulated by Even, Selman, and Yacobi [ESY84] and has several known consequences: Public-key cryptosystems that are NP-hard to crack do not exist; NP $\neq$ UP, NP $\neq$ coNP, and NPMV $\not\subseteq_c$

*Received by the editors April 16, 2003; accepted for publication (in revised form) May 12, 2004; published electronically August 27, 2004.

http://www.siam.org/journals/sicomp/33-6/42584.html

†Lehrstuhl für Informatik IV, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany (glasser@informatik.uni-wuerzburg.de). The research of this author was performed at the University at Buffalo with support by a postdoctoral grant from the German Academic Exchange Service (Deutscher Akademischer Austauschdienst—DAAD).

‡Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260 (selman@cse.buffalo.edu, samik@cse.buffalo.edu, lzhang7@cse.buffalo.edu). The research of the second author was partially supported by NSF grant CCR-0307077.

NPSV. Our main result in this section is an oracle X relative to which DisjNP does not contain a Turing-complete disjoint NP-pair and relative to which P $\neq$ UP. Relative to X, by Razborov's result [Raz94], optimal propositional proof systems do not exist. P-inseparable disjoint NP-pairs exist relative to X, because P $\neq$ UP [GS88]. Most researchers believe that P-inseparable disjoint NP-pairs exist, and we believe that no disjoint NP-pair has only NP-hard separators. Both of these properties hold relative to X. This is the first oracle relative to which both of these conditions hold simultaneously. Homer and Selman [HS92] obtained an oracle relative to which all disjoint NP-pairs are P-separable, so the conjecture of Even, Selman, and Yacobi holds relative to their oracle only for this trivial reason. Now let us say a few things about the construction of oracle X. Previous researchers have obtained oracles relative to which certain (promise) complexity classes do not have complete sets. However, the technique of Gurevich [Gur83], who proved that NP$\cap$coNP has Turing-complete sets if and only if it has many-one-complete sets, does not apply. Neither does the technique of Hemaspaandra, Jain, and Vereshchagin [HJV93], who demonstrated, among other results, an oracle relative to which FewP does not have a Turing-complete set.

In section 4 we show that the question of whether DisjNP contains a Turing-complete disjoint NP-pair has an equivalent natural formulation as a hypothesis about classes of single-valued partial functions. Section 5 studies *symmetric* disjoint NP-pairs. Pudlák [Pud03] defined a disjoint pair (A, B) to be symmetric if (A, B) is many-one reducible to (B, A). P-separable easily implies symmetric. We give complexity-theoretic evidence of the existence of nonsymmetric disjoint NP-pairs. As a consequence, we obtain new ways to demonstrate existence of P-inseparable sets. Also, we use symmetry to show under reasonable hypotheses that many-one and Turing reducibilities differ for disjoint NP-pairs. (All reductions in this paper are polynomial-time-bounded.) Concrete candidates for P-inseparable disjoint NP-pairs come from problems in UP or in NP $\cap$ coNP. Nevertheless, Grollmann and Selman [GS88] proved that the existence of P-inseparable disjoint NP-pairs implies the existence of P-inseparable disjoint NP-pairs, where both sets are NP-complete. Here we prove two analogous results. Existence of nonsymmetric disjoint NP-pairs implies existence of nonsymmetric disjoint NP-pairs, where both sets are NP-complete. If there exists a many-one-complete disjoint NP-pair, then there exists such a pair where both sets are NP-complete. Natural candidates for nonsymmetric or $\leq_m^{pp}$-complete disjoint NP-pairs arise either from cryptography or from proof systems [Pud03]. Our theorems show that the existence of such pairs will imply that nonsymmetric (or $\leq_m^{pp}$-complete) disjoint NP-pairs exist where both sets of the pair are $\leq_m^p$-complete for NP.

Section 6 constructs two oracles O_1 and O_2 that possess several interesting properties. First, let us mention some properties that hold relative to both of these oracles. Relative to both oracles, many-one-complete disjoint NP-pairs exist. Therefore, while we expect that complete disjoint NP-pairs do not exist, this is not provable by relativizable techniques. P-inseparable disjoint NP-pairs exist relative to these oracles, which we obtain by proving that nonsymmetric disjoint NP-pairs exist. The conjecture of Even, Selman, and Yacobi holds. Therefore, while nonexistence of Turing-complete disjoint NP-pairs is a sufficient condition for this conjecture, the converse does not hold, even in worlds in which P-inseparable pairs exist. Also, relative to these oracles, there exist P-inseparable pairs that are symmetric. Whereas nonsymmetric implies P-inseparability, again, we see that the converse does not hold.

In section 6 we discuss the properties of these oracles in detail. Relative to O_1, optimal proof systems exist, while relative to O_2, optimal proof systems do not exist. In particular, relative to O_2, the converse of Razborov's result does not hold. (That

is, relative to O_2, many-one complete pairs exist.)

The construction of O_2 involves some aspects that are unusual in complexity theory. We introduce undecidable requirements, and as a consequence, the oracle is undecidable. In particular, we need to define sets A and B, such that relative to O_2, the pair (A, B) is many-one complete. Therefore, we need to show that for every two nondeterministic, polynomial-time-bounded oracle Turing machines NM_i and NM_j, either $L(NM_i^{O_2})$ and $L(NM_j^{O_2})$ are not disjoint or there is a reduction from the disjoint pair $(L(NM_i^{O_2}), L(NM_j^{O_2}))$ to (A, B). We accomplish this as follows: Given NM_i, NM_j, and a finite initial segment X of O_2, we prove that either there is a finite extension Y of X such that for all oracles Z that extend Y,

$$L(NM_i^Z) \cap L(NM_j^Z) \neq \emptyset$$

or there is a finite extension Y of X such that for all oracles Z that extend Y,

$$L(NM_i^Z) \cap L(NM_j^Z) = \emptyset.$$

Then we select the extension Y that exists. In this manner we *force* one of these two conditions to hold.

In the latter case, to obtain a reduction from the pair $(L(NM_i^{O_2}), L(NM_j^{O_2}))$ to (A, B) requires encoding information into the oracle O_2. The other conditions that we want O_2 to satisfy require diagonalizations. In order to prove that there is room to diagonalize, we need to carefully control the number of words that must be reserved for encoding. This is a typical concern in oracle constructions, but even more so here. We manage this part of the construction by inventing a unique data structure that stores words reserved for the encoding, and then prove that we do not store too many such words.

2. Preliminaries. We fix the alphabet $\Sigma = \{0, 1\}$, and we denote the length of a word w by $|w|$. The set of all (resp., nonempty) words is denoted by Σ^* (resp., Σ^+). Let $\Sigma^{<n} \stackrel{df}{=} \{w \in \Sigma^* \mid |w| < n\}$, and define $\Sigma^{\leq n}$, $\Sigma^{\geq n}$, and $\Sigma^{>n}$ analogously. For a set of words X let $X^{<n} \stackrel{df}{=} X \cap \Sigma^{<n}$, and define $X^{\leq n}$, $X^{=n}$, $X^{\geq n}$, and $X^{>n}$ analogously. For sets of words we take the complement with respect to Σ^*. For $A, B \subseteq \Sigma^*$ let $A \oplus B \stackrel{df}{=} \{0x \mid x \in A\} \cup \{1y \mid y \in B\}$.

The set of (nonzero) natural numbers is denoted by $\mathbb{N}$ (resp., $\mathbb{N}^+$). We use polynomial-time computable and polynomial-time invertible pairing functions $\langle \cdot, \cdot \rangle : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ and $\langle \cdot, \cdot, \cdot \rangle : \mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$. For a function f, $\mathrm{dom}(f)$ denotes the domain of f.

Cook and Reckhow [CR79] defined a *propositional proof system* (proof system, for short) to be a function $f : \Sigma^* \to \mathrm{TAUT}$ such that f is onto and $f \in \mathrm{PF}$. (TAUT denotes the set of tautologies.) Note that f is not necessarily honest; it is possible that a formula $\phi \in \mathrm{TAUT}$ has only exponentially long proofs w, i.e., $f(w) = \phi$ and $|w| = 2^{\Omega(|\phi|)}$.

Let f and f' be two proof systems. We say that f *simulates* f' if there is a polynomial p and a function $h : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$, $f(h(w)) = f'(w)$ and $|h(w)| \leq p(|w|)$. If, additionally, $h \in \mathrm{PF}$, then we say that f *p-simulates* f'.

A proof system is *optimal* (resp., *p-optimal*) if it simulates (resp., p-simulates) every other proof system. The notion of simulation between proof systems is analogous to the notion of reducibility between problems. Using that analogy, optimal proof systems correspond to complete problems.

2.1. Disjoint pairs, separators, and a conjecture. We begin with the following definition.

DEFINITION 2.1. *A* disjoint NP-pair *(*NP*-pair, for short) is a pair of nonempty sets A and B such that $A, B \in$ NP and $A \cap B = \emptyset$. Let* DisjNP *denote the class of all disjoint* NP*-pairs.*

Given a disjoint NP-pair (A, B), a *separator* is a set S such that $A \subseteq S$ and $B \subseteq \overline{S}$; we say that S *separates* (A, B). Let $Sep(A, B)$ denote the class of all separators of (A, B). For disjoint NP-pairs (A, B), the fundamental question is whether $Sep(A, B)$ contains a set belonging to P. In that case the pair is P*-separable*; otherwise, the pair is P*-inseparable*. The following proposition summarizes the known results about P-separability.

PROPOSITION 2.2.

1. P $\neq$ NP $\cap$ coNP *implies that* NP *contains* P*-inseparable sets.*
2. P $\neq$ UP *implies that* NP *contains* P*-inseparable sets* [GS88].
3. *If* NP *contains* P*-inseparable sets, then* NP *contains* NP*-complete* P*-inseparable sets* [GS88].

While it is probably the case that NP contains P-inseparable sets, there is an oracle relative to which P $\neq$ NP and P-inseparable sets in NP do not exist [HS92]. So P $\neq$ NP probably is not a sufficiently strong hypothesis to show existence of P-inseparable sets in NP.

DEFINITION 2.3. *Let (A, B) be a disjoint* NP*-pair.*

1. $X \leq_m^{pp} (A, B)$ *if, for every separator S of (A, B), $X \leq_m^p S$.*
2. $X \leq_T^{pp} (A, B)$ *if, for every separator S of (A, B), $X \leq_T^p S$.*
3. (A, B) *is* NP-hard *if* SAT $\leq_T^{pp} (A, B)$.
4. (A, B) *is* uniformly NP-hard *if there is a deterministic polynomial-time oracle Turing machine M such that for every $S \in Sep(A, B)$,* SAT $\leq_T^p S$ *via M.*

Grollmann and Selman [GS88] showed that NP-hard implies uniformly NP-hard; i.e., both statements of the definition are equivalent. Even, Selman, and Yacobi [ESY84] conjectured that there does not exist a disjoint NP-pair (A, B) such that all separators of (A, B) are $\leq_T^p$ hard for NP.

CONJECTURE 2.4 (see [ESY84]). *There do not exist disjoint* NP*-pairs that are* NP*-hard.*

If Conjecture 2.4 holds, then no public-key cryptosystem is NP-hard to crack [ESY84]. This conjecture is a strong hypothesis with the following known consequences. In section 3 we show a sufficient condition for Conjecture 2.4 to hold.

PROPOSITION 2.5 (see [ESY84, GS88, Sel94]). *If Conjecture* 2.4 *holds, then* NP $\neq$ coNP*,* NP $\neq$ UP*, and* NPMV $\not\subseteq_c$ NPSV*.*

2.2. Reductions for disjoint pairs. We review the natural notions of reducibilities between disjoint pairs [GS88].

DEFINITION 2.6 (nonuniform reductions for pairs). *Let (A, B) and (C, D) be disjoint pairs.*

1. (A, B) *is* many-one reducible in polynomial-time *to (C, D), $(A, B) \leq_m^{pp} (C, D)$, if for every separator $T \in Sep(C, D)$ there exists a separator $S \in Sep(A, B)$ such that $S \leq_m^p T$.*
2. (A, B) *is* Turing reducible in polynomial-time *to (C, D), $(A, B) \leq_T^{pp} (C, D)$, if for every separator $T \in Sep(C, D)$ there exists a separator $S \in Sep(A, B)$ such that $S \leq_T^p T$.*

DEFINITION 2.7 (uniform reductions for pairs). *Let (A, B) and (C, D) be disjoint pairs.*

1. (A, B) *is* uniformly many-one reducible in polynomial-time *to* (C, D), $(A, B) \leq^{pp}_{um} (C, D)$, *if there exists a polynomial-time computable function* f *such that for every separator* $T \in Sep(C, D)$, *there exists a separator* $S \in Sep(A, B)$ *such that* $S \leq^{p}_{m} T$ *via* f.
2. (A, B) *is* uniformly Turing reducible in polynomial-time *to* (C, D), $(A, B) \leq^{pp}_{uT} (C, D)$, *if there exists a polynomial-time oracle Turing machine* M *such that for every separator* $T \in Sep(C, D)$, *there exists a separator* $S \in Sep(A, B)$ *such that* $S \leq^{p}_{T} T$ *via* M.

If f and M are as above, then we also say that $(A, B) \leq^{pp}_{um} (C, D)$ via f and $(A, B) \leq^{pp}_{uT} (C, D)$ via M. Observe that if $(A, B) \leq^{pp}_{m} (C, D)$ and (C, D) is P-separable, then so is (A, B) (and the same holds for $\leq^{pp}_{T}$, $\leq^{pp}_{um}$, and $\leq^{pp}_{uT}$). We retain the promise problem notation in order to distinguish from reducibilities between sets. Grollmann and Selman proved that Turing reductions and uniform Turing reductions are equivalent.

PROPOSITION 2.8 (see [GS88]). $(A, B) \leq^{pp}_{T} (C, D) \Leftrightarrow (A, B) \leq^{pp}_{uT} (C, D)$ *for all disjoint pairs* (A, B) *and* (C, D).

In order to obtain the corresponding theorem for $\leq^{pp}_{um}$, we can adapt the proof of Proposition 2.8, but a separate argument is required.

LEMMA 2.9. *Let* S *and* T *be nonempty, disjoint sets. Let* X *and* Y *be nonempty, finite, disjoint sets such that* $X \cap T = \emptyset$ *and* $Y \cap S = \emptyset$. *Then the disjoint pairs* (S, T) *and* $(S \cup X, T \cup Y)$ *are equivalent by polynomial-time uniform reductions.*

Proof. First we show that $(S \cup X, T \cup Y) \leq^{pp}_{um} (S, T)$. Choose $a \in S$ and $b \in T$. Define the polynomial-time computable function f by

$$f(x) \stackrel{df}{=} \begin{cases} a & \text{if } x \in X, \\ b & \text{if } x \in Y, \\ x & \text{otherwise.} \end{cases}$$

Let $A \in Sep(S, T)$. We need to see that $f^{-1}(A) \in Sep(S \cup X, T \cup Y)$. So we show that

1. $S \cup X \subseteq \underline{f^{-1}(A)}$, and
2. $T \cup Y \subseteq \overline{f^{-1}(A)}$.

For item 1, if $x \in X$, then $f(x) = a \in S \subseteq A$. So $f(X) \subseteq A$. Hence, $X \subseteq f^{-1}(A)$. If $x \in S - X$, then $f(x) = x \in S \subseteq A$. So, $S - X \subseteq \underline{f^{-1}(A)}$. For item 2, if $x \in Y$, then $f(x) = b \in T \subseteq \overline{A}$. So $f(Y) \cap A = \emptyset$. That is, $Y \subseteq \overline{f^{-1}(A)}$. If $x \in T - Y$, then $f(x) = x \in T$. So $f(T - Y) \cap A = \emptyset$. That is, $T - Y \subseteq \overline{f^{-1}(A)}$.

Every separator of $(S \cup X, T \cup Y)$ is a separator of (S, T). Therefore, the identity function provides a uniform reduction from (S, T) to $(S \cup X, T \cup Y)$. □

THEOREM 2.10. $\leq^{pp}_{m} = \leq^{pp}_{um}$.

Proof. Assume that (Q, R) is not uniformly many-one reducible to (S, T). That is, for every polynomial-time computable function f, there exists a set $A \in Sep(S, T)$ such that $f^{-1}(A) \notin Sep(Q, R)$. Then for every polynomial-time computable function f, there exists $A \in Sep(S, T)$ and a string y that witnesses the fact that $f^{-1}(A) \notin Sep(Q, R)$. Namely, either

$$y \in Q \wedge y \notin f^{-1}(A) \text{ (i.e., } f(y) \notin A) \qquad \text{or} \qquad y \in R \wedge y \in f^{-1}(A) \text{ (i.e., } f(y) \in A).$$

We will show from this assumption that (Q, R) is not many-one reducible to (S, T). We will construct a decidable separator A of (S, T) such that for every polynomial-time computable function f, $f^{-1}(A)$ is not a separator of (Q, R). Let $\{f_i\}_{i \geq 1}$ be an

effective enumeration of the polynomial-time computable functions with associated polynomial-time bounds $\{p_i\}_{i\geq 1}$.

The separator A of (S,T) will be constructed inductively to be of the form $S \cup \bigcup\{Y_i \mid i \geq 1\}$, where $\bigcup\{Y_i \mid i \geq 1\}$ is a subset of $\overline{T}$ and $Y_0 \subseteq Y_1 \subseteq \cdots$. At stage i of the construction, we will choose a finite subset Y_i of $\overline{T}$ such that $f^{-1}(S \cup Y_i)$ is not a separator of (Q,R).

Stage 0. Define $Y_0 = \{0\}$ and $n_0 = 1$.

Stage i ($i \geq 1$). By induction hypothesis, Y_{i-1} is defined, $n_{i-1} \geq 0$ is defined, and $Y_{i-1} \subseteq \overline{T} \cap \Sigma^{\leq n_{i-1}}$.

Now we state a sequence of claims.

CLAIM 2.11. *There exists a set* X, $X \subseteq \overline{T \cup \Sigma^{\leq n_{i-1}}}$, *and a witness* y_i *demonstrating that* $f_i^{-1}(S \cup Y_{i-1} \cup X)$ *is not a separator of* (Q,R). *That is,*

$$y_i \in Q \wedge y_i \notin f_i^{-1}(S \cup Y_{i-1} \cup X) \text{ (i.e., } f_i(y_i) \notin S \cup Y_{i-1} \cup X)$$

or

$$y_i \in R \wedge y_i \in f_i^{-1}(S \cup Y_{i-1} \cup X) \text{ (i.e., } f_i(y_i) \in S \cup Y_{i-1} \cup X).$$

If the claim is false, then for every $X \subseteq \overline{T \cup \Sigma^{\leq n_{i-1}}}$, $Q \subseteq f_i^{-1}(S \cup Y_{i-1} \cup X)$ and $R \subseteq \overline{f_i^{-1}(S \cup Y_{i-1} \cup X)}$. The set of all languages $S \cup Y_{i-1} \cup X$, where $X \subseteq \overline{T \cup \Sigma^{\leq n_{i-1}}}$, is exactly the set of separators of the disjoint pair

$$(S \cup Y_{i-1}, T \cup (\Sigma^{\leq n_{i-1}} - (S \cup Y_{i-1}))).$$

Thus, if the claim is false, then (Q,R) is uniformly many-one reducible to $(S \cup Y_{i-1}, T \cup (\Sigma^{\leq n_{i-1}} - Y_{i-1}))$. However, by Lemma 2.9, this contradicts the assumption that (Q,R) is not uniformly reducible to (S,T). Hence the claim is true.

CLAIM 2.12. *There exists a finite set* X, $X \subseteq \overline{T \cup \Sigma^{\leq n_{i-1}}}$, *and a witness* y_i *that satisfy the condition of Claim* 2.11.

For X and witness y_i, whose existence Claim 2.11 guarantees, $|f_i(y_i)| \leq p_i(|y_i|)$. So $X' = X \cap \Sigma^{\leq p_i(|y_i|)}$ and y_i satisfy the condition as well.

CLAIM 2.13. *There is an effective procedure that on input* (i, Y_{i-1}, n_{i-1}) *finds a finite set* $X \subseteq \overline{T \cup \Sigma^{\leq n_{i-1}}}$ *and witness* y_i *to satisfy the condition of Claim* 2.11.

This is trivial. Effectively enumerate pairs of finite sets and strings until a pair with the desired property is found.

At Stage i, apply Claim 2.13; define $Y_i = Y_{i-1} \cup X$ and define $n_i = 1 + \max(2^{n_{i-1}}, p_i(|y_i|))$.

Define $A = S \cup \bigcup\{Y_i \mid i \geq 1\}$. Since $\bigcup\{Y_i \mid i \geq 1\} \subseteq \overline{T}$, A is a separator of (S,T). It is easy to see that A is decidable. Finally, for every f_i, $i \geq 1$, $f_i^{-1}(A)$ is not a separator of (Q,R): Clearly this holds for $f_i^{-1}(S \cup Y_i)$, and the construction preserves this property. □

We obtain the following useful characterization of many-one reductions. Observe that this is the way Razborov [Raz94] defined reductions between disjoint pairs.

THEOREM 2.14. $(Q,R) \leq_m^{pp} (S,T)$ *if and only if there exists a polynomial-time computable function* f *such that* $f(Q) \subseteq S$ *and* $f(R) \subseteq T$.

Proof. By Theorem 2.10 there is a polynomial-time computable function f such for every $A \in Sep(S,T)$, $f^{-1}(A) \in Sep(Q,R)$. That is, if $A \in Sep(S,T)$, then $Q \subseteq f^{-1}(A)$ and $R \subseteq \overline{f^{-1}(A)}$, which implies that $f(Q) \subseteq A$ and $f(R) \cap A = \emptyset$. Well, $S \in Sep(S,T)$. So $f(Q) \subseteq S$. Also, $\overline{T} \in Sep(S,T)$. So $f(R) \cap \overline{T} = \emptyset$. That is, $f(R) \subseteq T$. The converse is immediate. □

3. Complete disjoint NP-pairs. Keeping with common terminology, a disjoint pair (A, B) is $\leq_m^{pp}$-complete ($\leq_T^{pp}$-complete) for the class DisjNP if $(A, B) \in$ DisjNP and for every disjoint pair $(C, D) \in$ DisjNP, $(C, D) \leq_m^{pp} (A, B)$ (resp., $(C, D) \leq_T^{pp} (A, B)$).

Consider the following assertions:

1. DisjNP does not have a $\leq_T^{pp}$-complete disjoint pair.
2. DisjNP does not have a $\leq_m^{pp}$-complete disjoint pair.
3. DisjNP does not contain a disjoint pair all of whose separators are $\leq_T^p$-hard for NP (i.e., Conjecture 2.4 holds).
4. DisjNP does not contain a disjoint pair all of whose separators are $\leq_m^p$-hard for NP.

Assertions 1 and 2 are possible answers to the question raised by Razborov [Raz94] of whether DisjNP contains complete disjoint pairs. Assertion 3 is Conjecture 2.4. Assertion 4 is the analogue of this conjecture using many-one reducibility.

We can dispense with assertion 4 immediately, for it is equivalent to NP $\neq$ coNP.

PROPOSITION 3.1. NP $\neq$ coNP *if and only if* DisjNP *does not contain a disjoint pair all of whose separators are* $\leq_m^p$*-hard for* NP.

Proof. If NP = coNP, then $(\mathrm{SAT}, \overline{\mathrm{SAT}})$ is a disjoint pair in DisjNP all of whose separators are $\leq_m^p$-hard for NP.

To show the other direction, consider the disjoint pair $(A, B) \in$ DisjNP and assume that all of its separators are $\leq_m^p$-hard for NP. Since $\overline{B}$ is a separator of (A, B), SAT $\leq_m^p \overline{B}$. Therefore, $\overline{\mathrm{SAT}} \leq_m^p B$, implying that $\overline{\mathrm{SAT}} \in$ NP. Thus, NP = coNP. □

PROPOSITION 3.2. *Assertion* 1 *implies assertions* 2 *and* 3. *Assertion* 2 *implies assertion* 4. *Assertion* 3 *implies assertion* 4.

This proposition states, in part, that assertion 1 is so strong as to imply Conjecture 2.4.

Proof. It is trivial that assertion 1 implies assertion 2, and that assertion 3 implies assertion 4.

We prove that assertion 1 implies assertion 3. Assume assertion 3 is false and let $(A, B) \in$ DisjNP such that all separators are NP-hard. We claim that (A, B) is $\leq_T^{pp}$-complete for DisjNP. Let (C, D) belong to DisjNP. Let S be an arbitrary separator of (A, B). Note that S is NP-hard and $C \in$ NP. So $C \leq_T^p S$. Since C is a separator of (C, D), this demonstrates that $(C, D) \leq_T^{pp} (A, B)$.

Similarly, we prove that assertion 2 implies assertion 4. In this case, every separator S of (A, B) is $\leq_m^p$-hard for NP. So $C \leq_m^p S$. Therefore, $(C, D) \leq_m^{pp} (A, B)$. □

Homer and Selman [HS92] constructed an oracle relative to which P $\neq$ NP and every disjoint NP-pair is P-separable. Relative to this oracle, assertion 3 holds and assertions 1 and 2 are false. To see this, let (A, B) be an arbitrary disjoint NP-pair. We show that (A, B) is both $\leq_T^{pp}$-complete and $\leq_m^{pp}$-complete. For any other pair $(C, D) \in$ DisjNP, since (C, D) is P-separable, there is a separator S of (C, D) that is in P. Therefore, for any separator L of (A, B), S trivially $\leq_m^p$-reduces and $\leq_T^p$-reduces to L. So $(C, D) \leq_m^{pp} (A, B)$ and $(C, D) \leq_T^{pp} (A, B)$.

There exists an oracle relative to which UP = NP $\neq$ coNP [GW03]. So, relative to this oracle assertion 4 holds, but assertion 3 is false. In section 6 we will construct oracles relative to which assertion 4 holds while assertions 1 and 2 fail.

In Theorem 3.8 we construct an oracle X relative to which assertion 1 is true. In Corollary 3.11 we observe that P $\neq$ UP relative to X. Therefore, by Proposition 3.2, all of the following properties hold relative to X:

1. DisjNP does not have a $\leq_T^{pp}$-complete disjoint pair.
2. Conjecture 2.4 holds; so UP $\neq$ NP, NP $\neq$ coNP, NPMV $\not\subseteq_c$ NPSV, and NP-hard public-key cryptosystems do not exist [ESY84, Sel94].
3. P $\neq$ UP; therefore P-inseparable disjoint NP-pairs exist [GS88].
4. Optimal propositional proof systems do not exist [Raz94].
5. There is a tally set $T \in \text{coNP} - \text{NP}$ and a tally set $T' \in \text{coNE} - \text{E}$ [Pud86, KP89].

The following lemma is essential to the proofs of Theorems 3.8 and 6.1. Intuitively this lemma says that, given two nondeterministic machines and some oracle, either we can force the languages accepted by these machines to be not disjoint, or we can ensure that one of the machines rejects a given string q by reserving only polynomially many strings.

LEMMA 3.3. *Let M and N be nondeterministic polynomial-time oracle Turing machines with polynomial-time bounds p_M and p_N, respectively. Let Y be an oracle and $q \in \Sigma^*$, $|q| = n$. Then, for any set T, at least one of the following holds:*

- *$\exists S \subseteq T$, $\|S\| \leq p_M(n) + p_N(n)$, such that $q \in L(M^{Y \cup S}) \cap L(N^{Y \cup S})$.*
- *$\exists S' \subseteq T$, $\|S'\| \leq p_M(n) \cdot (p_N(n) + 1)$, such that either* (i) *for any $S \subseteq T$, if $S \cap S' = \emptyset$, then $M^{Y \cup S}(q)$ rejects, or* (ii) *for any $S \subseteq T$, if $S \cap S' = \emptyset$, then $N^{Y \cup S}(q)$ rejects.*

Proof. Let us define the following languages:

- $L_M = \{\langle P, Q_y, Q_n\rangle \mid$ for some set $S_M \subseteq T$, P is an accepting path of $M^{Y \cup S_M}(q)$ and Q_y (resp., Q_n) is the set of words in S_M (resp., $T - (Y \cup S_M)$) that are queried on $P\}$.
- $L_N = \{\langle P, Q_y, Q_n\rangle \mid$ for some set $S_N \subseteq T$, P is an accepting path of $N^{Y \cup S_N}(q)$ and Q_y (resp., Q_n) is the set of words in S_N (resp., $T - (Y \cup S_N)$) that are queried on $P\}$.

We say that $\langle P, Q_y, Q_n\rangle \in L_M$ *conflicts* with $\langle P', Q'_y, Q'_n\rangle \in L_N$ if $Q_y \cap Q'_n \neq \emptyset$ or $Q'_y \cap Q_n \neq \emptyset$. In other words, there is a conflict if there exists at least one query that is in T and that is answered differently on P and P'.

Case I. There exist $\langle P, Q_y, Q_n\rangle \in L_M$ and $\langle P', Q'_y, Q'_n\rangle \in L_N$ that do not conflict.

Let $S = Q_y \cup Q'_y$. We claim in this case that $q \in L(M^{Y \cup S}) \cap L(N^{Y \cup S})$. Let S_M and S_N be the subsets of T that witness $\langle P, Q_y, Q_n\rangle \in L_M$ and $\langle P', Q'_y, Q'_n\rangle \in L_N$. So P is an accepting path of $M^{Y \cup S_M}(q)$, and P' is an accepting path of $N^{Y \cup S_N}(q)$. Assume that on P there exists a query r that is answered differently with respect to the oracles $Y \cup S_M$ and $Y \cup S$. Hence $r \notin Y$. Moreover, either $r \in S_M - S$ or $r \in S - S_M$. However, r cannot belong to $S_M - S$, since otherwise $r \in Q_y$, and therefore $r \in S$. So $r \in S - S_M$. Hence $r \notin Q_y$, and therefore $r \in Q'_y$. On the other hand, $r \in S - S_M$ implies $r \in T - (Y \cup S_M)$. Therefore, $r \in Q_n \cap Q'_y$, which contradicts the assumption in Case I. This shows that P is an accepting path of $M^{Y \cup S}(q)$. Analogously we show that P' is an accepting path of $N^{Y \cup S}(q)$. Hence $q \in L(M^{Y \cup S}) \cap L(N^{Y \cup S})$. Note that $\|S\| = \|Q_y \cup Q'_y\| \leq p_M(n) + p_N(n)$.

Case II. Every triple $\langle P, Q_y, Q_n\rangle \in L_M$ conflicts with every triple $\langle P', Q'_y, Q'_n\rangle \in L_N$.

Note that in this case we cannot have both a triple $\langle P, \emptyset, Q_n\rangle$ in L_M and a triple $\langle P', \emptyset, Q'_n\rangle$ in L_N, simply because these two triples do not conflict with each other. We use the following algorithm to create the set S' as claimed in the statement of this lemma.

$\mathtt{S}' = \emptyset$
`while` ($\mathtt{L_M} \neq \emptyset$ `and` $\mathtt{L_N} \neq \emptyset$)

```
(1)        Choose some (P*, Q*_y, Q*_n) ∈ L_M
(2)        S' = S' ∪ Q*_y ∪ Q*_n
(3)        For every t = (P, Q_y, Q_n) ∈ L_M
(4)           if Q_y ∩ (Q*_y ∪ Q*_n) ≠ ∅ then remove t
(5)        For every t' = (P', Q'_y, Q'_n) ∈ L_N
(6)           if Q'_y ∩ (Q*_y ∪ Q*_n) ≠ ∅ then remove t'
      end while
```

We claim that after k iterations of the *while* loop, for every triple $(P', Q'_y, Q'_n) \in L_N$, $\|Q'_n\| \geq k$. If this claim is true, the while loop iterates at most $p_N(n)+1$ times, since for any triple in L_N, $\|Q'_n\|$ is bounded by the running time of N on q, i.e., $p_N(n)$. On the other hand, during each iteration, S' is increased by at most $p_M(n)$ strings, since for any triple in L_M, $\|Q_y \cup Q_n\|$ is bounded by the running time of M on q, i.e., $p_M(n)$. Therefore, $\|S'\| \leq p_M(n) \cdot (p_N(n)+1)$ when this algorithm terminates.

CLAIM 3.4. *After the kth iteration of the* while *loop of the above algorithm, for every $t' = \langle P', Q'_y, Q'_n\rangle$ that remains in L_N, $\|Q'_n\| \geq k$.*

Proof. For every k, t_k denotes the triple $\langle P^k, Q^k_y, Q^k_n\rangle \in L_M$ that is chosen during the kth iteration in step (1). For every $t' = \langle P', Q'_y, Q'_n\rangle$ that is in L_N at the beginning of this iteration, t_k conflicts with t' (assumption of Case II). Therefore, there is a query in $(Q^k_n \cap Q'_y) \cup (Q^k_y \cap Q'_n)$. If this query is in $Q^k_n \cap Q'_y$, then t' will be removed from L_N in step (6). Otherwise, i.e., if $Q^k_y \cap Q'_n \neq \emptyset$, then let q' be the lexicographically smallest query in $Q^k_y \cap Q'_n$. In this case, t' will not be removed from L_N; we say that t' *survives* the kth iteration *due to query* q'. Note that t' can survive only due to a query that is in Q'_n. We will use this fact to prove that $\|Q'_n\| \geq k$ after k iterations.

We show now that any triple that is left in L_N after k iterations survives each iteration due to a different query. This will complete the proof of the claim. Assume that t' survives iteration k by query $q' \in Q^k_y \cap Q'_n$. If t' had survived an earlier iteration $l < k$ by the same query q', then q' is also in $Q^l_y \cap Q'_n$. Therefore, $Q^l_y \cap Q^k_y \neq \emptyset$. So $t_k = \langle P^k, Q^k_y, Q^k_n\rangle$ should have been removed in step (4) during iteration l, and cannot be chosen at the beginning of iteration k, as claimed. Hence, q' cannot be the query by which t' had survived iteration l. This proves Claim 3.4. □

Therefore, now we have a set $S' \subseteq T$ of the required size such that either L_M or L_N is empty. Assume that L_M is empty, and for some set $S_M \subseteq T$ it holds that $S_M \cap S' = \emptyset$ and $M^{(Y \cup S_M)}(q)$ accepts. Let P be an accepting path of $M^{(Y \cup S_M)}(q)$ and let Q_y (resp., Q_n) be the set of words in S_M (resp., $T - (Y \cup S_M)$) that are queried on P. The triple $\langle P, Q_y, Q_n\rangle$ must have been in L_M and has been removed during some iteration. This implies that during that iteration, $Q_y \cap S' \neq \emptyset$ (step (4)). Since $Q_y \subseteq S_M$, this contradicts the assumption that $S_M \cap S' = \emptyset$.

A similar argument holds for L_N. Hence either $L_M = \emptyset$ and $M^{(Y \cup S)}(q)$ rejects for any $S \subseteq T$ such that $S \cap S' = \emptyset$, or $L_N = \emptyset$ and $N^{(Y \cup S)}(q)$ rejects for any $S \subseteq T$ such that $S \cap S' = \emptyset$. This ends the proof of Lemma 3.3. □

We define the following notions for reductions relative to oracles.

DEFINITION 3.5. *For any set X, a pair of disjoint sets (A, B) is* polynomial-time Turing reducible relative to X *($\leq^{pp,X}_T$) to a pair of disjoint sets (C, D) if for any separator S that separates (C, D), there exists a polynomial-time deterministic oracle Turing machine M such that $M^{S \oplus X}$ accepts a language that separates (A, B).*

DEFINITION 3.6. *For any set X, let*

$$\text{DisjNP}^X = \{(A, B) \mid A \in \text{NP}^X,\ B \in \text{NP}^X,\ A \neq \emptyset,\ B \neq \emptyset,\ \textit{and}\ A \cap B = \emptyset\}.$$

(C, D) is $\leq^{pp,X}_T$-complete for DisjNP^X *if $(C, D) \in \text{DisjNP}^X$ and for all $(A, B) \in$*

DisjNP^X, $(A,B) \leq_T^{pp,X} (C,D)$. *Similarly,* (C,D) *is* $\leq_T^{pp}$*-complete for* DisjNP^X *if* $(C,D) \in \mathrm{DisjNP}^X$ *and for all* $(A,B) \in \mathrm{DisjNP}^X, (A,B) \leq_T^{pp} (C,D)$.

However, the following proposition shows that if there exists a disjoint pair that is Turing-complete relative to X, then there is a pair that is Turing-complete such that the reduction between the separators does not access the oracle.

PROPOSITION 3.7. *For any set* X, DisjNP^X *has a* $\leq_T^{pp,X}$*-complete pair if and only if* DisjNP^X *has a* $\leq_T^{pp}$*-complete pair.*

Proof. The *if* direction is trivial. We only show the *only if* direction. Suppose (C,D) is $\leq_T^{pp,X}$-complete for DisjNP^X. We claim that $(C \oplus X, D \oplus \overline{X})$ is $\leq_T^{pp}$-complete for DisjNP^X. Observe that $(C \oplus X, D \oplus \overline{X}) \in \mathrm{DisjNP}^X$. Consider any $(A,B) \in \mathrm{DisjNP}^X$. Let S' separate $(C \oplus X, D \oplus \overline{X})$. Define $S = \{x \mid 0x \in S'\}$. Then S separates (C,D) and $S' = S \oplus X$. Since (C,D) is $\leq_T^{pp,X}$-complete for DisjNP^X, there exists a polynomial-time oracle Turing machine M so that $L(M^{S \oplus X})$ separates (A,B). That is, $L(M^{S'})$ separates (A,B), which is what we needed to prove. □

THEOREM 3.8. *There exists an oracle* X *such that* DisjNP^X *does not have a* $\leq_T^{pp,X}$*-complete pair.*

Proof. By Proposition 3.7, it suffices to show that DisjNP^X has no $\leq_T^{pp}$-complete pair. By Proposition 2.8, it suffices to construct X such that for every $(C,D) \in \mathrm{DisjNP}^X$ there exists a disjoint pair $(A,B) \in \mathrm{DisjNP}^X$ such that $(A,B) \not\leq_{uT}^{pp} (C,D)$.

Suppose $\{M_k\}_{k \geq 1}$ (resp., $\{N_i\}_{i \geq 1}$) is an enumeration of deterministic (resp., nondeterministic) polynomial-time oracle Turing machines. Let r_k and p_i be the corresponding polynomial time bounds for M_k and N_i. For any r, s, d, let $\Sigma_{rs}^d = 0^r 10^s 1 \Sigma^d$ and $l_{rs}^d = r + s + d + 2$ (i.e., l_{rs}^d is the length of strings in Σ_{rs}^d). For $Z \subseteq \Sigma^*$, $i \geq 1$, and $j \geq 1$, define

$$A_{ij}^Z = \{0^n \mid \exists x,\ |x| = n,\ 0^i 10^j 10x \in Z\}$$

and

$$B_{ij}^Z = \{0^n \mid \exists x,\ |x| = n,\ 0^i 10^j 11x \in Z\}.$$

We construct the oracle in stages. X_m denotes the oracle before stage m. We define $X = \bigcup_{m \geq 1} X_m$. Initially, let $X = \emptyset$. In stage $m = \langle i,j,k \rangle$, we choose some number $n = n_m$ and add strings from Σ_{ij}^{n+1} to the oracle such that either $L(N_i^{X_{m+1}}) \cap L(N_j^{X_{m+1}}) \neq \emptyset$ or $(A_{ij}^{X_{m+1}}, B_{ij}^{X_{m+1}})$ is not uniformly Turing reducible to $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$ via $M_k^{X_{m+1}}$. This construction ensures that for every i and j, $(L(N_i^X), L(N_j^X))$ is not $\leq_{uT}^{pp}$-complete for DisjNP^X.

We describe the construction of X_{m+1}. We choose some large enough $n = n_m$, and we will add words from Σ_{ij}^{n+1} to the oracle. We need a sufficient number of words in Σ_{ij}^{n+1} for diagonalization. Therefore, n has to be large enough such that

$$r_k(n) p_i(r_k(n))(p_j(r_k(n)) + 1) < 2^n.$$

On the other hand, if $m \geq 2$, then we have to make sure that adding words of length l_{ij}^{n+1} does not influence diagonalizations made in former steps. Therefore, if $m \geq 2$ and $m - 1 = \langle i', j', k' \rangle$, then $n > n_{m-1}$ and n has to be large enough such that l_{ij}^{n+1} is greater than $l_{i'j'}^{n_{m-1}+1}$, $\max(p_{i'}(n_{m-1}), p_{j'}(n_{m-1}))$, and $\max(p_{i'}(r_{k'}(n_{m-1})), p_{j'}(r_{k'}(n_{m-1})))$. Since $n_{m-1} > n_{m-2} > \cdots$, these conditions not only guard against interference with step $m-1$, but guard against interference with all steps $m' < m$.

Suppose there exists an $S \subseteq \Sigma^{n+1}_{ij}$ such that $L(N_i^{X_m \cup S}) \cap L(N_j^{X_m \cup S}) \cap \Sigma^{\leq r_k(n)} \neq \emptyset$. Let $X_{m+1} = X_m \cup S$ and go to the next stage $m+1$.

Otherwise,

$$\text{for all } S \subseteq \Sigma^{n+1}_{ij}, \quad L(N_i^{X_m \cup S}) \cap L(N_j^{X_m \cup S}) \cap \Sigma^{\leq r_k(n)} = \emptyset. \tag{1}$$

In this case, we consider the computation of M_k on 0^n. We determine some $w \in \Sigma^{n+1}_{ij}$ and let $X_{m+1} = X_m \cup \{w\}$. We construct a set $Q \subseteq \overline{L(N_j^{X_{m+1}})}$. Hence $L(N_i^{X_{m+1}}) \cup Q$ is a separator of $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$. The sets X_{m+1} and Q satisfy either

$$0^n \in A_{ij}^{X_{m+1}} \quad \text{and} \quad 0^n \notin L(M_k^{L(N_i^{X_{m+1}}) \cup Q}) \tag{2}$$

or

$$0^n \in B_{ij}^{X_{m+1}} \quad \text{and} \quad 0^n \in L(M_k^{L(N_i^{X_{m+1}}) \cup Q}). \tag{3}$$

This shows that $(A_{ij}^{X_{m+1}}, B_{ij}^{X_{m+1}})$ does not $\leq^{pp}_{uT}$-reduce to $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$ via M_k.

The difficulty of finding w and Q rises mainly from the following: If we want to preserve the computation of M_k on 0^n, then we have to ensure that all oracle queries are preserved. Since the oracle is a separator of two NP languages, we have to maintain the acceptance behaviors of N_i and N_j with respect to the queries made by $M_k(0^n)$. This results in reserving too many strings. In particular, this may leave no room for the diagonalization in Σ^{n+1}_{ij}. However, by Lemma 3.3, we can do better.

Now we construct the set Q, and at the same time we reserve strings for $\overline{X_{m+1}}$. The latter makes sure that either N_i or N_j rejects on certain queries.

Initially we set $Q = \emptyset$. We run M_k on 0^n using oracle $L(N_i^{X_m}) \cup Q$, until the first string q is queried. We apply Lemma 3.3 with $M = N_i$, $N = N_j$, $Y = X_m$, and $T = \Sigma^{n+1}_{ij}$. By equation (1), the first statement of Lemma 3.3 cannot hold. Hence, there is a set $S' \subseteq \Sigma^{n+1}_{ij}$, $\|S'\| \leq p_i(r_k(n)) \cdot (p_j(r_k(n)) + 1)$, such that either

$$(\forall S,\, S \subseteq \Sigma^{n+1}_{ij},\, S \cap S' = \emptyset)[q \notin L(N_i^{X_m \cup S})] \tag{4}$$

or

$$(\forall S,\, S \subseteq \Sigma^{n+1}_{ij},\, S \cap S' = \emptyset)[q \notin L(N_j^{X_m \cup S})]. \tag{5}$$

We reserve all strings in S' for $\overline{X_{m+1}}$. If equation (4) is true, then we continue running M_k without changing Q. (Hence, answer "no" to query q.) Otherwise, let $Q = Q \cup \{q\}$ and continue running M_k with oracle $X_m \cup Q$. (Hence, answer "yes" to query q.) By the choice of q, Q remains a separator of $(L(N_i^{X_m}), L(N_j^{X_m}))$. We continue running M_k until the next string is queried and then apply Lemma 3.3 again, obtain the set S' that satisfies equation (4) or (5) for the new query, and update Q accordingly. We do this repeatedly until the end of the computation of M_k on 0^n.

The number of strings in Σ^{n+1}_{ij} that are reserved for $\overline{X_{m+1}}$ is at most

$$r_k(n) \cdot p_i(r_k(n)) \cdot (p_j(r_k(n)) + 1) < 2^n.$$

So there exist a string $0^i10^j10x \in \Sigma^{n+1}_{ij}$ and a string $0^i10^j11y \in \Sigma^{n+1}_{ij}$ such that neither string is reserved for $\overline{X_{m+1}}$. If $M_k^{L(N_i^{X_m}) \cup Q}(0^n)$ accepts, then let $w = 0^i10^j11y$.

Otherwise, let $w = 0^i 10^j 10x$. We define $X_{m+1} = X_m \cup \{w\}$. This completes stage m and we can go to the next stage $m+1$.

The following two claims prove the correctness of the construction.

CLAIM 3.9. *After every stage $m = \langle i,j,k\rangle$, either $L(N_i^{X_{m+1}}) \cap L(N_j^{X_{m+1}}) \cap \Sigma^{\leq r_k(n_m)} \neq \emptyset$ or $(A_{ij}^{X_{m+1}}, B_{ij}^{X_{m+1}})$ does not $\leq_{uT}^{pp}$-reduce to $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$ via M_k.*

Proof. If $L(N_i^{X_{m+1}}) \cap L(N_j^{X_{m+1}}) \cap \Sigma^{\leq r_k(n_m)} \neq \emptyset$, then we are done. Otherwise, it follows that equation (1) holds. In this case we constructed Q. We know that every string that was added to Q is enforced to be rejected by $N_j^{X_m}$. Since w is not reserved and $X_{m+1} = X_m \cup \{w\}$, Q is also in the complement of $L(N_j^{X_{m+1}})$. Therefore, $L(N_i^{X_{m+1}}) \cup Q$ is a separator of $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$.

All queries of $M_k(0^{n_m})$ under oracle $L(N_i^{X_{m+1}}) \cup Q$ are answered the same way as in the construction of Q. The reason is as follows: For any query q, if we reserve strings from $\Sigma_{ij}^{n_m+1}$ for $\overline{X_{m+1}}$ such that N_i always rejects q (equation (4)), then q will not be put into Q. Hence q will get the answer "no" from oracle $L(N_i^{X_{m+1}}) \cup Q$, which is the same as in the construction of Q. If we reserve strings from $\Sigma_{ij}^{n_m+1}$ for $\overline{X_{m+1}}$ such that N_j always rejects q (equation (5)), then q will be put into Q. Hence q gets the answer "yes" under oracle $L(N_i^{X_{m+1}}) \cup Q$, which is the same answer as given in the construction of Q. Therefore, by the choice of w, we obtain the following:

- If $M_k^{L(N_i^{X_{m+1}}) \cup Q}(0^{n_m})$ accepts, then $0^{n_m+1} \in B_{ij}^{L(N_i^{X_{m+1}}) \cup Q}$.
- If $M_k^{L(N_i^{X_{m+1}}) \cup Q}(0^{n_m})$ rejects, then $0^{n_m+1} \in A_{ij}^{L(N_i^{X_{m+1}}) \cup Q}$.

Hence $L(M_k^{L(N_i^{X_{m+1}}) \cup Q})$ does not separate $(A_{ij}^{X_{m+1}}, B_{ij}^{X_{m+1}})$. □

CLAIM 3.10. *For all $(C,D) \in \mathrm{DisjNP}^X$, where $C = L(N_i^X)$ and $D = L(N_j^X)$, it holds that $(A_{ij}^X, B_{ij}^X) \in \mathrm{DisjNP}^X$ and $(A_{ij}^X, B_{ij}^X) \not\leq_{uT}^{pp} (C,D)$.*

Proof. First, we claim that there is no stage $m = \langle i,j,k\rangle$ such that $L(N_i^{X_{m+1}}) \cap L(N_j^{X_{m+1}}) \cap \Sigma^{\leq r_k(n_m)} \neq \emptyset$. Otherwise, since the number n_{m+1} is chosen large enough, all strings that are added to the oracle in later stages will not change the computations of N_i and N_j on inputs of lengths $\leq r_k(n_m)$. Therefore, $L(N_i^X) \cap L(N_j^X) \neq \emptyset$, which contradicts our assumption.

From Claim 3.9 it follows that for every stage $m = \langle i,j,k\rangle$, $(A_{ij}^{X_{m+1}}, B_{ij}^{X_{m+1}})$ does not $\leq_{uT}^{pp}$-reduce to $(L(N_i^{X_{m+1}}), L(N_j^{X_{m+1}}))$ via M_k. Again, since n_{m+1} is chosen large enough, all strings added to the oracle in later stages will not change the following:

1. The membership of 0^{n_m} in $A_{ij}^{X_{m+1}}$ and $B_{ij}^{X_{m+1}}$. Strings of length $l_{ij}^{n_m+1}$ are only added to the oracle at stage m and not in any other stage.
2. The computations of N_i and N_j on inputs of lengths $\leq r_k(n_m)$ (which is the maximal length of strings that can be queried by M_k on 0^{n_m}).

Hence, (A_{ij}^X, B_{ij}^X) does not $\leq_{uT}^{pp}$-reduce to (C,D) via M_k. Since this holds for all k, we obtain $(A_{ij}^X, B_{ij}^X) \not\leq_{uT}^{pp} (C,D)$.

It remains to observe that $(A_{ij}^X, B_{ij}^X) \in \mathrm{DisjNP}^X$: For each $m = \langle i,j,k\rangle$ we added exactly one string from $\Sigma_{ij}^{n_m+1}$ to the oracle. Moreover, for any other $m' = \langle i',j',k'\rangle$ we added only words from $\Sigma_{i'j'}^{n_{m'}+1}$ to the oracle; this does not influence A_{ij}^X and B_{ij}^X. □

This completes the proof of the theorem. □

COROLLARY 3.11. *For the oracle X from Theorem 3.8 it holds that $\mathrm{P}^X \neq \mathrm{UP}^X$.*

Proof. Choose i and j such that N_i^X (resp., N_j^X) accepts X (resp., $\overline{X}$). We show that $A_{ij}^X \in \mathrm{UP}^X - \mathrm{P}^X$.

Note that $L(N_i^X) \cap L(N_j^X) = \emptyset$. By the construction in Theorem 3.11, for every length n, we add at most one string of the form $0^i 10^j 10x$, $|x| = n$, to the oracle. So $A_{ij}^X \in \mathrm{UP}^X$.

Assume $A_{ij}^X = L(M_k^X)$ for some deterministic polynomial-time oracle Turing machine M_k. Note that X is the only separator of $(L(N_i^X), L(N_j^X))$. Therefore, it follows that $(A_{ij}^X, B_{ij}^X) \leq_{uT}^{pp} (L(N_i^X), L(N_j^X))$ via M_k. This contradicts Claim 3.10. □

4. Function classes and disjoint pairs. We show that there exists a Turing-complete disjoint NP-pair if and only if NPSV contains a Turing-complete partial function. We know already that there is a connection between disjoint NP-pairs and NPSV. Namely, Selman [Sel94] proved that Conjecture 2.4 holds if and only if NPSV does not contain an NP-hard partial function, and Köbler and Messner [KM00] proved that there exists a many-one-complete disjoint NP-pair if and only if NPSV contains a many-one-complete partial function. Recall [Sel94] that NPSV is the set of all partial, single-valued functions computed by nondeterministic polynomial-time-bounded transducers.

If g is a single-valued total function, then we define $M[g]$, the single-valued partial function computed by M with oracle g, as follows: $x \in \mathrm{dom}(M[g])$ if and only if M reaches an accepting state on input x. In this case, $M[g](x)$ is the final value of M's output tape. In the case that g is a total function and $f = M[g]$, we write $f \leq_T^p g$.

The literature contains two different definitions of reductions between partial functions, because one must decide what to do in case a query is made to the oracle function when the query is not in the domain of the oracle function. Fenner et al. [FHOS97] determined that in this case the value returned should be a special symbol, $\perp$. Selman [Sel94] permits the value returned in this case to be arbitrary, which is the standard paradigm for promise problems. Here we use the promise problem definition of Selman [Sel94]. Recall that for multivalued partial functions f and g, g is *an extension* of f if $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$, and for all $x \in \mathrm{dom}(f)$ and for every y, if $g(x) \mapsto y$, then $f(x) \mapsto y$.

Definition 4.1. *For polynomial-length-bounded, partial multivalued functions f and g, f is* Turing reducible to g *(as a promise problem, so we write $f \leq_T^{pp} g$) in polynomial time if for some deterministic polynomial-time-bounded oracle transducer M, for every single-valued total extension g' of g, $M[g']$ is an extension of f.*

Here, if the query q belongs to the domain of g, then the oracle returns a value of $g(q)$. We will use the result [Sel94] that $f \leq_T^{pp} g$ if and only if for every single-valued total extension g' of g, there is a single-valued total extension f' of f such that $f' \leq_T^p g'$.

A single-valued partial function g is *$\leq_T^{pp}$-complete* for NPSV if g belongs to NPSV and, for all $f \in$ NPSV, $f \leq_T^{pp} g$.

Theorem 4.2. NPSV *contains a $\leq_T^{pp}$-complete partial function* $\Leftrightarrow$ DisjNP *contains a $\leq_T^{pp}$-complete pair.*

Proof. For any $f \in$ NPSV, define the following sets:

$$R_f = \{\langle x, y\rangle \mid x \in \mathrm{dom}(f),\ y \leq f(x)\} \tag{6}$$

and

$$S_f = \{\langle x, y\rangle \mid x \in \mathrm{dom}(f),\ y > f(x)\}. \tag{7}$$

Note that (R_f, S_f) is a disjoint NP-pair.

CLAIM 4.3. *For every separator A of (R_f, S_f), there is a single-valued total extension f' of f such that $f' \leq_T^p A$.*

Consider the following oracle transducer T that computes f' with oracle A. On input x, if $x \in \mathrm{dom}(f)$, then T determines the value of $f(x)$, using a binary search algorithm, by making repeated queries to A. Note that for $x \in \mathrm{dom}(f)$ and for any y, if $y \leq f(x)$, then $\langle x, y\rangle \in R_f$, and if $y > f(x)$, then $\langle x, y\rangle \in S_f$. Clearly, T computes some single-valued total extension of f. This proves the claim.

Let f be a $\leq_T^{pp}$-complete function for NPSV and assume that A separates R_f and S_f. By Claim 4.3, there is a single-valued total extension f' of f such that $f' \leq_T^p A$.

Let $(U, V) \in$ DisjNP. We want to show that $(U, V) \leq_T^{pp} (R_f, S_f)$. Define

$$g(x) = \begin{cases} 0 & \text{if } x \in U, \\ 1 & \text{if } x \in V, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $g \in$ NPSV, so $g \leq_T^{pp} f$. Therefore, there is a single-valued total extension g' of g such that $g' \leq_T^p f'$.

Define $L = \{x \mid g'(x) = 0\}$. It is easy to see that $L \leq_T^p g'$. Also note that $U \subseteq L$ and $V \subseteq \overline{L}$, and, therefore, L separates U and V. Then the following sequence of reductions shows that $L \leq_T^p A$:

$$L \leq_T^p g' \leq_T^p f' \leq_T^p A.$$

Thus, for every separator A of (R_f, S_f), there is a separator L of (U, V) such that $L \leq_T^p A$. Therefore, (R_f, S_f) is $\leq_T^{pp}$-complete for DisjNP.

For the other direction, assume that (U, V) is $\leq_T^{pp}$-complete for DisjNP. Define the following function:

$$f(x) = \begin{cases} 0 & \text{if } x \in U, \\ 1 & \text{if } x \in V, \\ \uparrow & \text{otherwise.} \end{cases}$$

Clearly, $f \in$ NPSV.

Let f' be a single-valued total extension of f, and let $L = \{x \mid f'(x) = 0\}$. Clearly, $L \leq_T^p f'$. Also, since $U \subseteq L$ and $V \subseteq \overline{L}$, L is a separator of (U, V).

We want to show that for any $g \in$ NPSV, $g \leq_T^{pp} f$. Consider the disjoint NP-pair (R_g, S_g) for the function g as defined in equations (6) and (7). There is a separator A of (R_g, S_g) such that $A \leq_T^p L$, since L is a separator of the $\leq_T^{pp}$-complete disjoint NP-pair (U, V). As noted in Claim 4.3, there is a single-valued total extension g' of g such that $g' \leq_T^p A$. Therefore, the following sequence of reductions shows that $g \leq_T^{pp} f$:

$$g' \leq_T^p A \leq_T^p L \leq_T^p f'.$$

Hence, f is complete for NPSV. □

COROLLARY 4.4.

1. *Let $f \in$ NPSV be $\leq_T^{pp}$-complete for* NPSV. *Then (R_f, S_f) is $\leq_T^{pp}$-complete for* DisjNP.

2. *If* (U, V) *is* $\leq_T^{pp}$*-complete for* DisjNP*, then* $f_{U,V}$ *is complete for* NPSV*, where*

$$f_{U,V}(x) = \begin{cases} 0 & \text{if } x \in U, \\ 1 & \text{if } x \in V, \\ \uparrow & \text{otherwise.} \end{cases}$$

3. *Relative to the oracle in Theorem* 3.8*,* NPSV *does not have a* $\leq_T^{pp}$*-complete partial function.*

5. Nonsymmetric pairs and separation of reducibilities. Pudlák [Pud03] defined a disjoint pair (A, B) to be *symmetric* if $(B, A) \leq_m^{pp} (A, B)$. Otherwise, (A, B) is *nonsymmetric*. For example, the canonical disjoint NP-pair for the propositional proof system Resolution is symmetric [Pud03] (see section 6.3 for the definition of canonical pairs). In this section we give complexity-theoretic evidence of the existence of nonsymmetric disjoint NP-pairs. As a consequence, we obtain new ways to demonstrate existence of P-inseparable sets and show that $\leq_m^{pp}$ and $\leq_T^{pp}$ reducibilities differ for disjoint NP-pairs.

A set L is *P-printable* if there is $k \geq 1$ such that all elements of L up to length n can be printed by a deterministic Turing machine in time $n^k + k$ [HY84, HIS85]. Every P-printable set is sparse and belongs to P. An infinite set A is *P-printable-immune* if no infinite subset of A is P-printable.

A set L is *p-selective* if there is a polynomial-time-bounded function f such that for every $x, y \in \Sigma^*$, $f(x, y) \in \{x, y\}$, and $\{x, y\} \cap L \neq \emptyset \Rightarrow f(x, y) \in L$ [Sel79].

A partial function $f \in$ PF is *almost-always one-way* [FPS01] if no polynomial-time Turing machine inverts f correctly on more than a finite subset of range(f).

Proposition 5.1.

1. (A, B) *is symmetric if and only if* (B, A) *is symmetric.*
2. *If* (A, B) *is P-separable, then* (A, B) *is symmetric.*

Proof. The proof of the first assertion is trivial. For the proof of the second assertion, let (A, B) be a P-separable disjoint NP-pair. Fix $a \in A$ and $b \in B$, and let the separator be $S \in$ P. Consider the following polynomial-time computable function f. On input x, if $x \in S$, then f outputs b; otherwise, f outputs a. Therefore, $x \in A$ implies $x \in S$, which implies $f(x) = b \in B$, and $x \in B$ implies $x \notin S$, which implies $f(x) = a \in A$. Therefore, $(A, B) \leq_m^{pp} (B, A)$, i.e., (A, B) is symmetric. □

We will show the existence of a nonsymmetric disjoint NP-pair under certain hypotheses, due to the following proposition, that will separate $\leq_m^{pp}$ and $\leq_T^{pp}$ reducibilities.

Proposition 5.2.

1. *If* (A, B) *is a nonsymmetric disjoint* NP*-pair, then* $(B, A) \not\leq_m^{pp} (A, B)$.
2. *For any disjoint* NP*-pair* (A, B), $(B, A) \leq_T^{pp} (A, B)$.

Proof. The first assertion follows from the definition of symmetric pairs. For the second assertion, observe that for any S separating A and B, $\overline{S}$ separates B and A, while for any set S, $\overline{S} \leq_T^p S$. □

We will use the following proposition in a crucial way to provide some evidence for the existence of nonsymmetric disjoint NP-pairs. In other words, we will seek to obtain a disjoint NP-pair (A, B) such that either A or B is p-selective, but (A, B) is not P-separable.

Proposition 5.3. *For any disjoint* NP*-pair* (A, B)*, if either* A *or* B *is p-selective, then* (A, B) *is symmetric if and only if* (A, B) *is P-separable.*

Proof. We know from Proposition 5.1 that if (A, B) is P-separable, then it is symmetric. Now assume that (A, B) is symmetric via some function f and assume

(without loss of generality) that A is p-selective and the p-selector function is g. The following algorithm M separates A and B. On input x, M runs g on the strings $(x, f(x))$, and accepts x if and only if g outputs x. If $x \in A$, then $f(x) \in B$, and therefore g has to output x. On the other hand, if $x \in B$, then $f(x) \in A$. So g will output $f(x)$ and M will reject x. Therefore, $A \subseteq L(M) \subseteq \overline{B}$. □

Now we give evidence for the existence of nonsymmetric disjoint NP-pairs.

THEOREM 5.4. *If* $\mathrm{E} \neq \mathrm{NE} \cap \mathrm{coNE}$*, then there is a set* $A \in \mathrm{NP} \cap \mathrm{coNP}$ *such that* $(A, \overline{A})$ *is not symmetric.*

Proof. If $\mathrm{E} \neq \mathrm{NE} \cap \mathrm{coNE}$, then there is a tally set $T \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$. From Selman [Sel79, Theorem 5], the existence of such a tally set implies that there is a p-selective set $A \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$. Clearly, $(A, \overline{A})$ is not P-separable. Hence, by Proposition 5.3, $(A, \overline{A})$ is nonsymmetric. □

As a corollary, if $\mathrm{E} \neq \mathrm{NE} \cap \mathrm{coNE}$, then there is a set $A \in \mathrm{NP} \cap \mathrm{coNP}$ such that $(A, \overline{A}) \not\leq_m^{pp} (\overline{A}, A)$, yet clearly $(A, \overline{A}) \leq_T^{pp} (\overline{A}, A)$.

We will show that the hypotheses in Theorem 5.5 imply the existence of a nonsymmetric disjoint NP-pair. Note that the hypotheses in this theorem are similar to those studied by Fortnow, Pavan, and Selman [FPS01] and Pavan and Selman [PS02]. However, our hypotheses are stronger than the former and weaker than the latter.

THEOREM 5.5. *The following are equivalent.*

1. *There is a* UP*-machine* N *that accepts* 0^* *and, for every polynomial-time machine* M*,* $\{n \mid M$ *on input* 0^n *outputs the accepting computation of* N *on input* $0^n\}$ *is a finite set.*
2. *There is a set* S *in* UP *accepted by a* UP*-machine* N *such that* S *has exactly one string of every length and, for every polynomial-time machine* M*, the following set is finite:* $\{n \mid M$ *on input* 0^n *outputs the accepting computation of* N *on input* $x_n\}$*, where* x_n *denotes the word of length* n *that belongs to* S*.*
3. *There is an honest one-to-one, almost-always one-way function* f *such that* $\mathrm{range}(f) = 0^*$*.*
4. *There is a language* $L \in \mathrm{P}$ *that has exactly one string of every length and* L *is P-printable-immune.*
5. *There is a language* $L \in \mathrm{UP}$ *that has exactly one string of every length and* L *is P-printable-immune.*

Proof. We show the following cycles: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$ and $1 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$.

Trivially, item 1 implies item 2. To prove that item 2 implies item 3, let N be a UP-machine that satisfies the conditions of item 2 and let $S = L(N)$. For any y that encodes an accepting computation of N on some string x, define $f(y) = 0^{|x|}$. Since y also encodes x, f is polynomial-time computable. Since N runs in polynomial time, f is honest. On the other hand, if any polynomial-time computable machine can invert f on 0^n for infinitely many n, then that machine actually outputs infinitely many accepting computations of N.

We show that item 3 implies item 1. Given f as in item 3, we know that since f is honest, $\exists k > 0$ such that $|x| \leq |f(x)|^k$. We describe a UP-machine N that accepts 0^*. On input 0^n, N guesses x, $|x| \leq n^k$, and accepts 0^n if and only if $f(x) = 0^n$. Since f is one-to-one, N has exactly one accepting path for every input of the form 0^n, and since $\mathrm{range}(f) = 0^*$, $L(N) = 0^*$. If there is a polynomial-time machine M that outputs infinitely many accepting computations of N, then M also inverts f on infinitely many strings.

To prove that item 1 implies item 4, let N be the UP-machine in item 1. We can assume without loss of generality that for all but finitely many n, on input 0^n, N has

exactly one accepting computation of length n^k for some $k > 0$. Let us define the following language:

$$L' = \{x10^n10^l \mid n \geq 0,\ x \text{ is an accepting path of } N(0^n), \text{ and } 0 \leq l \leq (n+1)^k - n^k\}.$$

It is easy to see that L' is in P, and for all but finitely many n, L has exactly one string of length n. Therefore, there exists a finite variation $L \in \mathrm{P}$ such that L has exactly one string of every length. If L has an infinite P-printable subset, then so has L'. Let M' be a polynomial-time transducer that prints an infinite subset of L'. It follows that M' outputs infinitely many accepting computations of N.

Item 4 trivially implies item 5. We show that item 5 implies item 1. Let L be such a language in UP via a UP-machine N. Define a UP-machine N' to accept 0^* as follows. On input 0^n, N' guesses a string x of length n and a computation path w of N on x. N' accepts 0^n if and only if w is an accepting computation. If a polynomial-time machine can output infinitely many accepting computations of N', then essentially the same machine also outputs infinitely many strings in L, and hence L cannot be P-printable-immune. □

THEOREM 5.6. *Each of the hypotheses stated in Theorem* 5.5 *implies the existence of nonsymmetric disjoint* NP*-pairs.*

Proof. Let us define the following function:

$$dt(i) = \begin{cases} 1 & \text{if } i = 0, \\ 2^{2^{dt(i-1)}} & \text{otherwise.} \end{cases}$$

Let M be the UP-machine accepting 0^*, as in the first hypothesis in Theorem 5.5. Let a_n be the accepting computation of M on 0^n. We can assume that $|a_n| = p(n)$, where $p(\cdot)$ is some fixed polynomial. We define the following sets:

$$L_M = \{\langle 0^n, w\rangle \mid w \leq a_n,\ n = dt(i) \text{ for some } i > 0\}$$

and

$$R_M = \{\langle 0^n, w\rangle \mid w > a_n,\ n = dt(i) \text{ for some } i > 0\}.$$

Note that (L_M, R_M) is a disjoint NP-pair. We claim that L_M is p-selective. The description of a selector f for L_M follows. Assume that $\langle 0^k, w_1\rangle$ and $\langle 0^l, w_2\rangle$ are input to f. If $k = l$, then f outputs the lexicographically smaller one of w_1 and w_2. Otherwise, assume that $k < l$, and without loss of generality, both k and l are in range(dt). In that case, $l \geq 2^{2^k} > 2^{|a_k|}$, and therefore f can compute a_k, the accepting computation of M on 0^k, by checking all possible strings of length $|a_k|$. Therefore, in $O(l)$ time, f outputs $\langle 0^k, w_1\rangle$ if $w_1 \leq a_k$, and outputs $\langle 0^l, w_2\rangle$ otherwise. Similarly, we can show that R_M is p-selective.

We claim that (L_M, R_M) is a nonsymmetric disjoint NP-pair. Assume on the contrary that this pair is symmetric. Therefore, by Proposition 5.3 (L_M, R_M) is P-separable; i.e., there is $S \in \mathrm{P}$ that is a separator for (L_M, R_M). Using a standard binary search technique, a polynomial-time machine can compute the accepting computation of M on any 0^n, where $n = dt(i)$ for some $i > 0$. Since the length of the accepting computation of M on 0^n is $p(n)$, this binary search algorithm takes time $O(p(n))$ which is polynomial in n. This contradicts our hypothesis, since we assumed that no polynomial-time machine can compute infinitely many accepting computations of M. Therefore, (L_M, R_M) is a nonsymmetric disjoint NP-pair. □

If the hypotheses stated in Theorem 5.5 hold, then there exists a disjoint NP-pair (A, B) so that $(A, B) \not\leq_m^{pp} (B, A)$ while $(A, B) \leq_T^{pp} (B, A)$.

Grollmann and Selman [GS88] proved that the existence of P-inseparable disjoint NP-pairs implies the existence of P-inseparable pairs where both sets of the pair are NP-complete. The following results are in the same spirit. We note that natural candidates for nonsymmetric (or $\leq_m^{pp}$-complete) disjoint NP-pairs arise either from cryptography or from proof systems [Pud03]. However, the following theorems show that the existence of such pairs will imply that nonsymmetric (or $\leq_m^{pp}$-complete) disjoint NP-pairs exist where both sets of the pair are $\leq_m^p$-complete for NP.

THEOREM 5.7. *There exists a nonsymmetric disjoint* NP*-pair* (A, B) *if and only if there exists a nonsymmetric disjoint* NP*-pair* (C, D) *where both* C *and* D *are* $\leq_m^p$*-complete for* NP.

Proof. The *if* part is trivial. We prove the *only if* part. Let $\{NM_i\}_{i\geq 1}$ be a standard enumeration of polynomial-time-bounded nondeterministic Turing machines with associated polynomial-time bounds $\{p_i\}_{i\geq 1}$. It is known that the following set is NP-complete [BGS75]:

$$K = \{\langle i, x, 0^n\rangle \mid NM_i \text{ accepts } x \text{ within } n \text{ steps}\}.$$

Let (A, B) be a nonsymmetric disjoint NP-pair. There exists $i \geq 1$ such that $A = L(NM_i)$, and $A \leq_m^p K$ via $f(x) = \langle i, x, 0^{p_i(|x|)}\rangle$. Note that f is honest and one-to-one.

Our first goal is to show that $(K, f(B))$ is nonsymmetric. Since f is a reduction from A to K and $A \cap B = \emptyset$, $f(A) \subseteq K$ and $f(B) \subseteq \overline{K}$, and so $f(B)$ and K are disjoint sets. Observe that $f(B)$ is in NP because on any input y, we can guess x, and verify that $x \in B$ and $f(x) = y$. Therefore, $(K, f(B))$ is a disjoint NP-pair, and K is $\leq_m^p$-complete for NP.

In order to prove that this pair is nonsymmetric, assume otherwise. Then $(K, f(B)) \leq_m^{pp} (f(B), K)$ and, therefore, $\exists g \in$ PF such that $g(K) \subseteq f(B)$ and $g(f(B)) \subseteq K$. Consider the following polynomial-time computable function h. On input x, h first computes $y = g(f(x))$. If $y = \langle i, x', 0^{p_i(|x'|)}\rangle$ for some x', then h outputs x'; otherwise, it returns a fixed string $a \in A$. We claim that $h(A) \subseteq B$ and $h(B) \subseteq A$, thereby making (A, B) symmetric. For any $x \in A$, we know that $f(x) \in K$. Hence $g(f(x)) \in f(B)$, since $g(K) \subseteq f(B)$. So $g(f(x)) = \langle i, x', 0^{p_i(|x'|)}\rangle$ for some $x' \in B$, and so $h(x) = x' \in B$. For any $x \in B$, $y = g(f(x)) \in K$, since $g(f(B)) \subseteq K$. If $y = \langle i, x', 0^{p_i(|x'|)}\rangle$ for some x', then x' must be in A; else h will return $a \in A$, and so, in either case, $x \in B$ will imply that $h(x) \in A$. Therefore, $h(A) \subseteq B$ and $h(B) \subseteq A$. Thus $(A, B) \leq_m^{pp} (B, A)$, contradicting the fact that (A, B) is nonsymmetric. Hence $(K, f(B))$ is a nonsymmetric disjoint NP-pair.

To complete the proof of the theorem, apply the construction once again, this time with an honest reduction f' from $f(B)$ to K. Namely, $f'(f(B)) \subseteq K$ and $f'(K) \subseteq \overline{K}$. Similar to the above argument, it can be shown that $f'(K)$ and K are disjoint. Also, since f' is one-to-one, we claim that $f'(K)$ is $\leq_m^p$-complete for NP. Clearly, $x \in K$ implies $f'(x) \in f'(K)$. On the other hand, for some $x \notin K$, $f'(x)$ cannot be in $f'(K)$; otherwise, $f'(x) = f'(y)$ for some $y' \in K$, contradicting the fact that f' is one-to-one. Then K and $f'(K)$ are disjoint NP-complete sets, and the argument already given shows that $(f'(K), K)$ is nonsymmetric. □

THEOREM 5.8. *There exists a* $\leq_m^{pp}$*-complete disjoint* NP*-pair* (A, B) *if and only if there exists a* $\leq_m^{pp}$*-complete disjoint* NP*-pair* (C, D)*, where both* C *and* D *are* $\leq_m^p$*-complete sets for* NP.

Proof. The proof is similar to that of Theorem 5.7. Consider the one-to-one

function f defined by $f(x) = \langle i, x, 0^{p_i(|x|)} \rangle$ that many-one reduces A to the canonical NP-complete set K.

Obviously $(A, B) \leq_m^{pp} (K, f(B))$ via f, since $f(A) \subseteq K$, and $K \cap f(B) = \emptyset$, as shown in the proof of Theorem 5.7. Similar to that theorem, we apply the one-to-one function f' that many-one reduces $f(B)$ to K to obtain another disjoint pair $(f'(K), K)$ where $(K, f(B)) \leq_m^{pp} (f'(K), K)$ via f'. So $(A, B) \leq_m^{pp} (K, f(B)) \leq_m^{pp} (f'(K), K)$. Therefore $(f'(K), K)$ is also a $\leq_m^{pp}$-complete disjoint NP-pair, and both $f'(K)$ and K are $\leq_m^p$-complete sets for NP. □

6. Optimal proof systems relative to an oracle. The question of whether optimal propositional proof systems exist has been studied in detail. Pudlák [Pud86] and Krajíček and Pudlák [KP89] showed that NE = coNE implies the existence of optimal proof systems. Ben-David and Gringauze [BDG98] and Köbler, Messner, and Torán [KMT03] obtained the same conclusion under weaker assumptions. On the other hand, Messner and Torán [MT98] and Köbler, Messner, and Torán [KMT03] proved that existence of optimal proof systems results in the existence of $\leq_m^p$-complete sets for the promise class NP ∩ SPARSE. These results hold relative to all oracles. Therefore, optimal proof systems exist relative to any oracle in which NE = coNE holds. Krajíček and Pudlák [KP89], Ben-David and Gringauze [BDG98], and Buhrman et al. [BFFvM00] constructed oracles relative to which optimal proof systems do not exist. In addition, NP ∩ SPARSE does not have complete sets relative to the latter oracle.

The relationship between the existence of optimal proof systems and disjoint NP-pairs was first established by Razborov [Raz94], who showed that the existence of optimal proof systems implies the existence of many-one-complete disjoint NP-pairs. Köbler, Messner, and Torán [KMT03] proved that this holds even for a stronger form of many-one reductions. They defined *strong many-one reduction* (we denote this by $\leq_{sm}^{pp}$) between disjoint NP-pairs as follows: $(A, B) \leq_{sm}^{pp} (C, D)$ if there is $f \in$ PF such that $f(A) \subseteq C$, $f(B) \subseteq D$, and $f(\overline{A \cup B}) \subseteq \overline{C \cup D}$.[1]

In this section, we construct two oracles, O_1 and O_2. Relative to O_1, NE = coNE, and therefore [Pud86, KP89] optimal proof systems exist, implying the existence of $\leq_m^p$-complete sets for NP ∩ SPARSE [MT98] as well as the existence of $\leq_{sm}^{pp}$-complete disjoint NP-pairs [KMT03]. On the other hand, relative to this oracle, E ≠ NE∩coNE = NE, thus implying, by Theorem 5.4, that nonsymmetric (and therefore P-inseparable) pairs exist. Since nonexistence of $\leq_T^{pp}$-complete disjoint NP-pairs implies Conjecture 2.4, it is natural to ask whether the converse of this implication holds. Relative to O_1, Conjecture 2.4 holds, and so the converse is false.

Ben-David and Gringauze [BDG98] asked whether the converse to Razborov's result holds. Relative to O_2, NP ∩ SPARSE does not have a complete set, and so optimal proof systems do not exist. On the other hand, $\leq_{sm}^{pp}$-complete disjoint NP-pairs exist. This shows that the converse to Razborov's result does not hold (even for the stronger notion of many-one reduction) in a relativized setting. Relative to O_2, the existence of $\leq_{sm}^{pp}$-complete disjoint NP-pairs does not imply the existence of $\leq_m^p$-complete sets in NP∩SPARSE. In addition, relative to O_2, NE ≠ coNE [Pud86, KP89] and nonsymmetric disjoint NP-pairs exist.

Since relative to both O_1 and O_2, Conjecture 2.4 holds, $\leq_{sm}^{pp}$-complete disjoint NP-pairs exist, and nonsymmetric pairs exist, it follows that these are "independent"

[1] A forthcoming paper [GSS04] proves that there exist $\leq_{sm}^{pp}$-complete disjoint NP-pairs if and only if there exist $\leq_m^{pp}$-complete disjoint NP-pairs.

TABLE 1
Comparison of oracle properties.

	O_1	O_2
$\exists \leq^{pp}_{sm}$-complete disjoint NP-pairs	Yes	Yes
$\exists$ nonsymmetric disjoint NP-pairs	Yes	Yes
Conjecture 2.4 holds	Yes	Yes
E $\neq$ NE	Yes	Yes
NE = coNE	Yes	No
$\exists$ optimal propositional proof systems	Yes	No
NP $\cap$ SPARSE has $\leq^p_m$-complete sets	Yes	No

of the assertion that NE = coNE, the existence of optimal proof systems, and existence of $\leq^p_m$-complete sets in NP $\cap$ SPARSE. In Table 1, we summarize the properties of both oracles; "Yes" denotes that a particular property holds, while "No" means that the property does not hold.

6.1. Notation. We fix the following enumerations: $\{NM_i\}_i$ is an effective enumeration of nondeterministic, polynomial-time-bounded oracle Turing machines; $\{NE_i\}_i$ is an effective enumeration of nondeterministic, linear exponential-time-bounded oracle Turing machines; $\{M_i\}_i$ is an effective enumeration of deterministic, polynomial-time-bounded oracle Turing machines; $\{E_i\}_i$ is an effective enumeration of deterministic, linear exponential-time-bounded oracle Turing machines; and $\{T_i\}_i$ is an effective enumeration of deterministic, polynomial-time-bounded oracle Turing transducers. Moreover, NM_i, M_i, and T_i have running time $p_i = n^i$, and NE_i and E_i have running time 2^{in} independent of the choice of the oracle. For any oracle Z, let f_i^Z denote the function that T_i^Z computes.

We use the following model of nondeterministic oracle Turing machines. On some input the machine starts the first phase of its computation, during which it is allowed to make nondeterministic branches. In this phase the machine is not allowed to ask any queries. At the end of the first phase the machine has computed a list of queries $q_1, \ldots, q_n$, a list of guessed answers $g_1, \ldots, g_n$, and a character, which is either + or −. Now the machine asks in parallel all queries and gets the vector of answers $a_1, \ldots, a_n$. The machine accepts if the computed character is + and $(a_1, \ldots, a_n) = (g_1, \ldots, g_n)$; otherwise the machine rejects. An easy observation shows that for every nondeterministic polynomial-time oracle Turing machine M there exists a machine N that works in the described way such that for all oracles X, $L(M^X) = L(N^X)$.[2] The analogous statement holds for nondeterministic, linear exponential-time-bounded oracle Turing machines.

A computation path P of a nondeterministic polynomial-time oracle Turing machine N on an input x contains all nondeterministic choices, all queries, and all guessed answers. A computation path P that has the character + (resp., −) is called a positive (resp., negative) path. The set of queries that are guessed to be answered positively (resp., negatively) is denoted by P^{yes} (resp., P^{no}); the set of all queries is denoted by $P^{\text{all}} \stackrel{df}{=} P^{\text{yes}} \cup P^{\text{no}}$. The length of P (i.e., the number of computation steps) is denoted by $|P|$. Note that this description of paths makes it possible to talk about paths of computations without specifying the oracle; i.e., we can say that N on x has

[2]Note that for this property we need both: the character must be + and g_i must be guessed correctly. If the machine accepts just when the answers are guessed correctly, then we miss the machine that accepts $\emptyset$ for every oracle.

a positive path P such that P^{yes} and P^{no} satisfy certain conditions. However, when talking about accepting and rejecting paths we always have to specify the oracle. (A positive path can be accepting for certain oracles, and it can be rejecting for other oracles.)

For $X, Y \subseteq \Sigma^*$ we write $Y \supseteq_m X$ if $X \subseteq \Sigma^{\leq m}$ and $Y^{\leq m} = X$. We write $Y \subseteq_m X$ if and only if $X \supseteq_m Y$. We need to consider injective, partial functions $\mu : \mathbb{N}^+ \to \mathbb{N} \times \mathbb{N}^+$ that have a finite domain. We do not distinguish between the function and the set of all (n, i, j) such that $\mu(n) = (i, j)$. We denote both by μ. Let μ and μ' be injective, partial functions $\mathbb{N}^+ \to \mathbb{N} \times \mathbb{N}^+$ that have a finite domain. If $\mu \neq \emptyset$, then $\mu_{\max} \stackrel{df}{=} \max(\text{dom}(\mu))$. We write $\mu \preceq \mu'$ if either $\mu = \emptyset$, or $\mu \subseteq \mu'$ and $\mu_{\max} < n$ for all $n \in \text{dom}(\mu' - \mu)$. We write $\mu \prec \mu'$ if $\mu \preceq \mu'$ and $\mu \neq \mu'$.

For $j \geq 1$, SPARSE_j denotes the class of all languages L such that for all $k \geq 0$, $\|L \cap \Sigma^k\| \leq k^j + j$.

6.2. Existence of optimal proof systems. Now we develop the first of these oracles.

THEOREM 6.1. *There exists an oracle relative to which the following holds:*

(i) $\text{E} \neq \text{NE} = \text{coNE}$.

(ii) *Conjecture* 2.4 *holds.*

For a fixed set X, let us define the following set, which is complete for NE^X:

$$C^X \stackrel{df}{=} \{\langle i, x, l\rangle \mid NE_i^X \text{ accepts } x \text{ within } l \text{ steps}\}.$$

We also define the following property:

$$\text{P1:} \qquad \langle i, x, l\rangle \in C^X \Leftrightarrow (\forall y,\ |y| = 2^{2|\langle i,x,l\rangle|})[\langle i, x, l\rangle y \notin X].$$

We call a set $X \subseteq \Sigma^{\leq k}$ *k-valid* if property P1 holds for all strings $\langle i, x, l\rangle$ such that $|\langle i, x, l\rangle| + 2^{2|\langle i,x,l\rangle|} \leq k$. Note that $\emptyset$ is 0-valid and that the condition on the right-hand side of P1 only depends on words in X that have length $2^{2n} + n$ for some natural number n. We define the following sets:

$$A^X \stackrel{df}{=} \{0^n \mid (n \text{ is odd}) \wedge (\exists y,\ |y| = 2^n)[y \in X]\}$$

and

$$B^X \stackrel{df}{=} \{0^{2^n} z \mid (n \text{ is odd}) \wedge |z| = 2^n \wedge (\exists y,\ |y| = 2^n)[zy \in X]\}.$$

Clearly, $A^X \in \text{NE}^X$ and $B^X \in \text{NP}^X$. We require the following for O_1:

1. $C^{O_1} \in \text{coNE}^{O_1}$. (This implies $\text{NE}^{O_1} = \text{coNE}^{O_1}$, because C^{O_1} is complete for NE^{O_1} by a reduction that is computable in linear time.)
2. $A^{O_1} \notin \text{E}^{O_1}$ (which implies $\text{E}^{O_1} \neq \text{NE}^{O_1}$, since $A^{O_1} \in \text{NE}^{O_1}$).
3. For every i, j, and r, B^{O_1} does not $\leq_T^{pp}$-reduce to $(L(NM_i^{O_1}), L(NM_j^{O_1}))$ via M_r. This will ensure that Conjecture 2.4 holds relative to O_1.

Proof of Theorem 6.1. We will begin by stating two lemmas that will be used in this proof.

LEMMA 6.2. *For every i and every k-valid X, there exists an l-valid $Y \supseteq_k X$, where $l > k$, such that for every $Z \supseteq_l Y$, $A^Z \neq L(E_i^Z)$.*

LEMMA 6.3. *For every i, j, r and every k-valid X, there exists an l-valid $Y \supseteq_k X$, where $l > k$, such that for every $Z \supseteq_l Y$, B^Z does not $\leq_T^{pp}$-reduce to $(L(NM_i^Z), L(NM_j^Z))$ via M_r.*

We define the following list $\mathcal{T}$ of requirements. At the beginning of the construction, $\mathcal{T}$ contains $\{i\}_{i \geq 1}$ and $\{(i, j, r)\}_{i,j,r \geq 1}$. These have the following interpretations:

- $i \in \mathcal{T}$: ensure that $A^{O_1} \neq L(E_i^{O_1})$.
- $(i,j,r) \in \mathcal{T}$: ensure that B^{O_1} does not $\leq_T^{pp}$-reduce to $(L(NM_i^{O_1}), L(NM_j^{O_1}))$ via M_r.

The following algorithm is used to construct the oracle O_1.

```
1    O_1 := ∅; k := 0
2    while {true} {
3        Remove the next requirement t from T
4        if t = i then
5           apply Lemma 6.2 with X = O_1 to get Y and l
6        else  // t = (i, j, r)
7           apply Lemma 6.3 with X = O_1 to get Y and l
8        O_1 := Y; k := l
9    }
```

It is clear that the oracle constructed by this algorithm satisfies items (i) and (ii) of Theorem 6.1. It remains to prove Lemmas 6.2 and 6.3.

Proof of Lemma 6.2. Fix an i and let X be any k-valid oracle. Let n be the smallest odd length such that $k \leq 2^n - 1$, $n-1 < 2^{n-1}$, and $2^{in} < 2^{2^n}$. Note first that we can assume that $k = 2^n - 1$. Otherwise, we claim that X can be extended to some (2^n-1)-valid oracle $X' \supseteq_k X$. Assume that X is $(m-1)$-valid for $k < m \leq 2^n - 1$; we will show how X can be extended to an m-valid oracle. This can be iterated to extend X to be (2^n-1)-valid.

Assume $m = 2^{2r} + r$ and consider some $\langle j,x,l\rangle$ of length r. (If m is not of this form, then, by property P1, an $(m-1)$-valid oracle is automatically an m-valid oracle.)

Note that $|x| \leq r$ and $|l| \leq r$. Hence, $NE_j^X(x)$ can ask only queries of length $\leq 2^r < m-1$. The answers to these queries will not change during the later stages of the construction. So the result of $NE_j^X(x)$ is fixed. If $NE_j^X(x)$ rejects within l steps, then choose some y of length 2^{2r} and put $\langle j,x,l\rangle y$ in X. Otherwise, do not put any such string in X. After all strings $\langle j,x,l\rangle$ are treated, we obtain an oracle X that is m-valid. This shows that we can assume X to be (2^n-1)-valid.

Also note that any string $w = \langle j,x,l\rangle y$ cannot have length 2^n. If $|w| = 2^n$, then, since $|y| = 2^{2|\langle j,x,l\rangle|}$, $|\langle j,x,l\rangle| < n/2$. Hence, the highest length possible for $\langle j,x,l\rangle$ is $(n-1)/2$, in which case $|y| = 2^{n-1}$ and $|w| = (n-1)/2 + 2^{n-2} < 2^n$. If $|\langle j,x,l\rangle|$ is even smaller, then y is of smaller length as well, and so is $|w|$. This shows that $|w|$ can never be 2^n for any n. As a consequence, we know that at stage $k+1$ we do not have to put any strings of the form $\langle j,x,l\rangle y$ into X. Therefore, we can use this stage for diagonalization.

Now we want to show that there exists an l-valid Y, $l \geq 2^n$, such that for every $Z \supseteq_l Y$, $A^Z \neq L(E_i^Z)$. Consider the computation of E_i^X on 0^n. Since the running time of E_i is bounded above by 2^{in}, the queries made by $E_i^X(0^n)$ have length at most 2^{in}. Let N be the set of queries of length $\geq 2^n$ (these are answered "no" in this computation). Note that $\|N\| \leq 2^{in} < 2^{2^n}$. We put some $v \in \Sigma^{2^n} - N$ in X if and only if $E_i^X(0^n)$ rejects. By the above discussion, $k = 2^n \neq 2^{2r} + r$ for any r, and so v cannot be of the form $\langle j,x,l\rangle y$. Therefore, X is 2^n-valid.

Claim 6.4. *We can extend X to some 2^{in}-valid $Y \supseteq_{2^n} X$ such that $N \subseteq \overline{Y}$.*

Proof. Fix some $\langle j,x,l\rangle$ such that $2^n < |\langle j,x,l\rangle y| \leq 2^{in}$. First we show that there are at least 2^{2^n} different such y for this $\langle j,x,l\rangle$. We show this by proving that $|y| \geq 2^n$. If $|y| < 2^n$, then, since length of y can only be a power of 2, let us assume that $y = 2^{n-1}$. Then $|\langle j,x,l\rangle| = (n-1)/2$, and therefore $|\langle j,x,l\rangle y| = (n-1)/2 + 2^{n-1} < 2^n$, contradicting that $|\langle j,x,l\rangle y| > 2^n$.

Now, simulate $NE_j^X(x)$ for l steps. If the simulation $NE_j^X(x)$ accepts within l steps, then do not update X. Otherwise, i.e., if the simulation rejects, then choose y' such that $|y'| = 2^{2|\langle j,x,l\rangle|}$ and $\langle j,x,l\rangle y' \notin N$. Put $\langle j,x,l\rangle y'$ in X. Existence of such y' is ensured, since the possible number of these words is 2^{2^n}, whereas $\|N\| \le 2^{in} < 2^{2^n}$.

So, if NE_j^X accepts x within l steps, no extra string is put in X. On the other hand, if $NE_j^X(x)$ does not accept within l steps, then we put an appropriate $\langle j,x,l\rangle y' \notin N$ in X. Once this procedure is completed for all $\langle j,x,l\rangle$, the oracle we obtain is 2^{in}-valid. We call that oracle Y. This proves Claim 6.4. □

The proof of the lemma is completed by noting that $Y \supseteq_{2^n} X$ and $Y \subseteq \overline{N}$. Hence, $0^n \in A^Y \Leftrightarrow 0^n \notin L(E_i^Y)$. Let $l = 2^{in}$ (which is the l Lemma 6.2 refers to). Any $Z \supseteq_{2^{in}} Y$ differs from Y only by strings of lengths $> 2^{in}$. This does not affect the computation of $E_i(0^n)$, and therefore, by our construction, it follows that $0^n \in A^Z \Leftrightarrow 0^n \notin L(E_i^Z)$. This proves Lemma 6.2. □

Proof of Lemma 6.3. Similar to the proof of Lemma 6.2, we can assume that $k = 2^{n+1} - 1$, where n is odd. Let $c \stackrel{df}{=} (2^{n+1})^{r(i+j)}$. We choose n to be large enough so that the following hold:

- $p_r(2^{n+1})p_i(p_r(2^{n+1}))(p_j(p_r(2^{n+1})) + 1) < 2^{2^n}$;
- $2(2^{n+1})^{2r(i+j)} < 2^{2^n}$, i.e., $2c^2 < 2^{2^n}$.

CLAIM 6.5. *There exist $Y' \subseteq \Sigma^{\le c}$, $N' \subseteq \Sigma^{\le c}$ such that $\|Y'\| \le c^2$, $\|N'\| \le c^2$, and for all $X' \subseteq \Sigma^{2^{n+1}}$, if $N' \subseteq \overline{X'}$, then $X \cup Y' \cup X'$ is c-valid.*

We will prove this claim later.

Choose some z such that $|z| = 2^n$ and for all y, $|y| = 2^n$, $zy \notin Y'$, and $zy \notin N'$. (Such a z exists because both $\|Y'\|, \|N'\| \le c^2$, and $2c^2 < 2^{2^n}$.) We can assume that

$$(\forall X' \subseteq z\Sigma^{2^n})[L(NM_i^{X\cup Y'\cup X'}) \cap L(NM_j^{X\cup Y'\cup X'}) \cap \Sigma^{\le p_r(2^{n+1})} = \emptyset]. \tag{8}$$

Otherwise $Y = X \cup Y' \cup X'$ satisfies the requirement of Lemma 6.3.

We will consider the computation of M_r on $0^{2^n}z$ and construct sets Q and X' such that $L(NM_i^{X\cup Y'\cup X'}) \cup Q$ is a separator of $L(NM_i^{X\cup Y'\cup X'})$ and $L(NM_j^{X\cup Y'\cup X'})$, and either

$$0^{2^n}z \in B^{X\cup Y'\cup X'} \quad \text{and} \quad 0^{2^n}z \notin L(M_r^{L(NM_i^{X\cup Y'\cup X'})\cup Q})$$

or

$$0^{2^n}z \notin B^{X\cup Y'\cup X'} \quad \text{and} \quad 0^{2^n}z \in L(M_r^{L(NM_i^{X\cup Y'\cup X'})\cup Q}).$$

This will imply that $B^{X\cup Y'\cup X'}$ does not $\le_T^{pp}$-reduce to $(L(NM_i^{X\cup Y'\cup X'}), L(NM_j^{X\cup Y'\cup X'}))$ via M_r. The details follow.

Initially we set $Q = \emptyset$. We run M_r on $0^{2^n}z$ using oracle $L(NM_i^{X\cup Y'}) \cup Q$. Note that this oracle is a separator of $(L(NM_i^{X\cup Y'}), L(NM_j^{X\cup Y'}))$. The simulation of M_r on $0^{2^n}z$ is continued until it makes some query q. At this point, we apply Lemma 3.3 with $M = NM_i$, $N = NM_j$, $Y = X \cup Y'$, and $T = z\Sigma^{2^n}$. Note that on input $0^{2^n}z$, M_r can make queries up to length $p_r(2^{n+1})$, and we have $\|T\| = 2^{2^n} > p_i(p_r(2^{n+1}))(p_j(p_r(2^{n+1})) + 1)$. By Lemma 3.3 and equation (8), there is a set $S' \subseteq z\Sigma^{2^n}$ such that either

$$(\forall S \subseteq z\Sigma^{2^n},\ S \cap S' = \emptyset)[q \notin L(NM_i^{X\cup Y'\cup S})] \tag{9}$$

or

$$(\forall S \subseteq z\Sigma^{2^n},\ S \cap S' = \emptyset)[q \notin L(NM_j^{X\cup Y'\cup S})]. \tag{10}$$

We know that $\|S'\| \leq p_i(p_r(2^{n+1}))(p_j(p_r(2^{n+1})) + 1)$. We reserve all strings in S' for $\overline{X'}$. If equation (9) is true, then we continue simulating M_r without modifying the oracle (hence, answer "no" to query q). Otherwise, if equation (9) does not hold, we update $Q = Q \cup \{q\}$ (hence, answer "yes" to query q and add q to the oracle) and continue the simulation of M_r on $0^{2^n}z$. We continue running M_r until the next query, and then we apply Lemma 3.3 again, obtain the set S' that satisfies above equation (9) or equation (10) for the new query and update Q accordingly. We keep doing this until the end of the computation of M_r on $0^{2^n}z$. The number of strings in $z\Sigma^{2^n}$ we reserved for $\overline{X'}$ during the above process is at most $p_r(2^{n+1})p_i(p_r(2^{n+1}))(p_j(p_r(2^{n+1})) + 1) < 2^{2^n}$ since the running time of M_r on $0^{2^n}z$ is bounded by $p_r(2^{n+1})$.

Since the number of strings reserved for $\overline{X'}$ in the above process is strictly less than the number of strings of length 2^n, there exists a string zy in $z\Sigma^{2^n}$ that is not reserved for $\overline{X'}$. If M_r using oracle $L(NM_i^{X\cup Y'}) \cup Q$ accepts $0^{2^n}z$, we define $X' = \emptyset$. In this case, $0^{2^n}z \notin B^{X\cup Y'\cup X'}$. Otherwise, define $X' = \{zy\}$, in which case $0^{2^n}z \in B^{X\cup Y'\cup X'}$. Also observe that q is put in Q only when $q \notin L(NM_j^{X\cup Y'\cup X'})$. Therefore, $L(NM_i^{X\cup Y'\cup X'}) \cup Q$ remains a separator of $L(NM_i^{X\cup Y'\cup X'})$ and $L(NM_j^{X\cup Y'\cup X'})$.

Let $Y \stackrel{df}{=} X \cup Y' \cup X'$. It is clear from the discussion above that B^Y does not $\leq_T^{pp}$-reduce to $L(NM_i^Y, NM_j^Y)$ via M_r. Since $X' \subseteq \overline{N'}$, Y is $c = (2^{n+1})^{r(i+j)}$-valid. Furthermore, any string q that can be queried by M_r on $0^{2^n}z$ is of length $\leq (2^{n+1})^r$. Therefore, the strings that are queried by NM_i and NM_j on input q are of lengths at most $(2^{n+1})^{r(i+j)} = c$. This implies that for all $Z \supseteq_c Y$, B^Z does not $\leq_T^{pp}$-reduce to $(L(NM_i^Z), L(NM_j^Z))$ via M_r, since any string of length more than c will not affect the outcome of the computation. It remains to prove Claim 6.5.

Proof of Claim 6.5. We use the following algorithm to construct Y' and N'. Recall that $c = (2^{n+1})^{r(i+j)}$.

```
1.   Y' = ∅, N' = ∅
2.   Treated = ∅
3.   L = {⟨i,x,l⟩ | 2^{n+1} < |⟨i,x,l⟩y| ≤ c where |y| = 2^{2|⟨i,x,l⟩|}}
4.   while L ≠ ∅ {
5.      Remove the smallest ⟨i,x,l⟩ from L
6.      Treated = Treated ∪ {⟨i,x,l⟩}
7.      if (∃X' ⊆ Σ^{2^{n+1}} such that X' ⊆ N' and
                    NE_i^{X∪Y'∪X'}(x) accepts within l steps)
8.         Choose an accepting path P
9.         Y' = Y' ∪ P^yes and N' = N' ∪ P^no
        else
10.        Choose some y ∈ Σ^{2|⟨i,x,l⟩|} such that ⟨i,x,l⟩y ∉ N'
11.        Y' = Y' ∪ {⟨i,x,l⟩y}
12.  } //end while.
```

We claim that after each iteration of the *while* loop, the following invariance holds: For every $X' \subseteq \overline{N'} \cap \Sigma^{2^{n+1}}$, property P1 holds for each $\langle i, x, l\rangle$ in `Treated` with oracle $X \cup Y' \cup X'$. Initially, when `Treated` is empty, this holds trivially.

Let us assume that $\langle i, x, l\rangle$ is put in `Treated` during iteration $m \geq 1$ of the while loop. It is straightforward to see that after this iteration, the statements in the loop ensure that the invariance holds for $\langle i, x, l\rangle$, since $\langle i, x, l\rangle y$ is put into the oracle if and only if NE_i does not accept x within l steps. We have to show that the invariance also holds for every such triple that had been put into `Treated` in some iteration $m' < m$. Let $\langle j, u, t\rangle$ be such a triple. It suffices to show that for t steps, $NE_j(u)$ behaves the

same way after the mth iteration as it does after the m'th iteration. Assume that during the m'th iteration NE_j accepted u in t steps. All the queries that are made on that accepting path are already in Y' or N' accordingly. Therefore, that path remains accepting even during the mth iteration.

On the other hand, let us assume that for every X', NE_j rejected u in t steps during the m'th iteration. We will show that it will still reject u after the mth iteration. To see this, let us assume that a previously rejecting path has become an accepting path after the mth iteration. A query that was answered "yes" at that point cannot be answered "no" now, since Y' now contains strictly more strings. So assume that the queries $q_1, \ldots, q_d$ were answered "no" during the m'th iteration with $X \cup Y' \cup X'$ as the oracle and are now answered "yes." All strings that are added to Y' after iteration m' are either of lengths $\geq |\langle j, u, t\rangle y| > t$ or are from some $X' \subseteq \Sigma^{2^{n+1}}$. Hence $q_1, \ldots, q_d$ must be of length 2^{n+1}. Note that at least one of these queries must have been in N' during the m'th iteration; otherwise NE_j would accept u at that point with oracle $X \cup Y' \cup (X' \cup \{q_1, \ldots, q_d\})$. But any string that was in N' during an earlier iteration is not put in X' or Y' in later iterations. Therefore, our assumption is false, and NE_j will reject u during the mth iteration as well. This proves the invariance.

What remains to show are the bounds on the sizes of Y' and N' and the maximum length of strings in Y' and N'. For the size of Y' and N', note that if $|\langle i, x, l\rangle y| \leq c$, then, since $|y| = 2^{2|\langle i,x,l\rangle|}$, $|\langle i, x, l\rangle| \leq (\log c)/2$, and therefore $\|\mathcal{L}\| \leq 2^{(\log c)/2+1} < c$. On the other hand, during every iteration, at most l strings are added to Y' and N', and $|l| < |\langle i, x, l\rangle| \leq (\log c)/2$, and therefore $l < c$ as well. Since both Y' and N' are initially empty, they are at most c^2 in size. The maximum length of strings in Y' and N' is c since the longest string that is added to Y' or N' is $\max_{\langle i,x,l\rangle \in \mathcal{L}} |\langle i, x, l\rangle y| \leq c$.

This completes the proof of Claim 6.5. □

This finishes the proof of Lemma 6.3. □

This proves Theorem 6.1. □

COROLLARY 6.6. *The oracle O_1 of Theorem* 6.1 *has the following additional properties:*

(i) $\mathrm{UP}^{O_1} \neq \mathrm{NP}^{O_1} \neq \mathrm{coNP}^{O_1}$ *and* $\mathrm{NPMV}^{O_1} \not\subseteq_c \mathrm{NPSV}^{O_1}$.

(ii) *Relative to O_1, optimal propositional proof systems exist.*

(iii) *There exists a $\leq_{sm}^{pp,O_1}$-complete disjoint NP^{O_1}-pair (A, B) that is P^{O_1}-inseparable but symmetric.*

6.3. Nonexistence of optimal proof systems. In this section we construct an oracle relative to which there exist $\leq_{sm}^{pp}$-complete disjoint NP-pairs. For any oracle X, $(A, B) \leq_{sm}^{pp,X} (C, D)$ if there is a function $f \in \mathrm{PF}^X$ such that $f(A) \subseteq C$, $f(B) \subseteq D$, and $f(\overline{A \cup B}) \subseteq \overline{C \cup D}$.[3]

THEOREM 6.7. *There exists an oracle O_2 relative to which the following holds:*

(i) *There exist $\leq_{sm}^{pp}$-complete disjoint* NP*-pairs.*

(ii) *There exist nonsymmetric disjoint* NP*-pairs.*

(iii) $\mathrm{NP} \cap \mathrm{SPARSE}$ *does not have $\leq_m^p$-complete sets.*

(iv) *Conjecture* 2.4 *holds.*

[3] $(A, B) \leq_m^{pp,X} (C, D)$ if for every separator $T \in Sep(C, D)$, there exists a separator $S \in Sep(A, B)$ such that $S \leq_m^{p,X} T$. However, since Theorems 2.10 and 2.14 hold relative to all oracles, $(A, B) \leq_m^{pp,X} (C, D)$ if and only if there is a function $f \in \mathrm{PF}^X$ such that $f(A) \subseteq C$ and $f(B) \subseteq D$. It follows immediately that $(A, B) \leq_{sm}^{pp,X} (C, D)$ implies $(A, B) \leq_m^{pp,X} (C, D)$.

Proof. In our construction we use the following witness languages, which depend on an oracle Z:

$$A(Z) \stackrel{df}{=} \{w \mid w = 0^n 10^t 1x \text{ for } n,t \geq 1,\ x \in \Sigma^* \text{ and } (\exists y \in \Sigma^{3|w|+3})[0wy \in Z]\},$$
$$B(Z) \stackrel{df}{=} \{w \mid w = 0^n 10^t 1x \text{ for } n,t \geq 1,\ x \in \Sigma^* \text{ and } (\exists y \in \Sigma^{3|w|+3})[1wy \in Z]\},$$
$$C(Z) \stackrel{df}{=} \{0^k \mid k \equiv 1 \pmod 4,\ (\exists y \in \Sigma^{k-1})[0y \in Z]\},$$
$$D(Z) \stackrel{df}{=} \{0^k \mid k \equiv 1 \pmod 4,\ (\exists y \in \Sigma^{k-1})[1y \in Z]\},$$
$$E_i(Z) \stackrel{df}{=} \{0^i 1x \mid |0^i 1x| \equiv 1 \pmod 4 \text{ and } (\exists y \in \Sigma^*,\ |y| = |0^i 1x|)[0^i 1xy \in Z]\} \quad \text{for } i \geq 1,$$
$$F(Z) \stackrel{df}{=} \{0^k \mid k \equiv 3 \pmod 4,\ (\exists y \in \Sigma^k)[y \in Z]\}.$$

These languages are in NP^Z. By definition, $A(Z)$ and $B(Z)$ depend on oracle words of length $\equiv 0 \pmod 4$, $C(Z)$ and $D(Z)$ depend on oracle words of length $\equiv 1 \pmod 4$, all $E_i(Z)$ depend on oracle words of length $\equiv 2 \pmod 4$, and $F(Z)$ depends on oracle words of length $\equiv 3 \pmod 4$. We construct the oracle O_2 such that $A(O_2) \cap B(O_2) = C(O_2) \cap D(O_2) = \emptyset$ and the following holds:

- $(A(O_2), B(O_2))$ is $\leq_{sm}^{pp}$-complete. That is,

$$\begin{aligned}&(\forall (G,H) \in \mathrm{DisjNP}^{O_2})(\exists f \in \mathrm{PF})\\ &[f(G) \subseteq A(O_2) \wedge f(H) \subseteq B(O_2) \wedge f(\overline{G \cup H}) \subseteq \overline{A(O_2) \cup B(O_2)}].\end{aligned} \tag{11}$$

- $(C(O_2), D(O_2))$ is nonsymmetric. That is,

$$(\forall f \in \mathrm{PF}^{O_2})[f(C(O_2)) \not\subseteq D(O_2) \vee f(D(O_2)) \not\subseteq C(O_2)]. \tag{12}$$

- $\mathrm{NP}^{O_2} \cap \mathrm{SPARSE}$ does not have $\leq_m^{p,O_2}$-complete sets. That is,

$$\begin{aligned}&(\forall j,\ L(NM_j^{O_2}) \in \mathrm{SPARSE}_j)(\exists n,\ E_n(O_2) \text{ contains} \leq 2 \text{ words of every length})\\ &(\forall f \in \mathrm{PF}^{O_2})[E_n(O_2) \text{ does not} \leq_m^{p,O_2}\text{-reduce to } L(NM_j^{O_2}) \text{ via } f].\end{aligned} \tag{13}$$

- $F(O_2) \not\leq_T^{pp,O_2} (A(O_2), B(O_2))$. That is,

$$(\exists S,\ A(O_2) \subseteq S \subseteq \overline{B(O_2)})[F(O_2) \notin \mathrm{P}^S]. \tag{14}$$

In (11) and (14) we really mean $f \in \mathrm{PF}$ and $F(O_2) \notin \mathrm{P}^S$; we explain why this is equivalent to $f \in \mathrm{PF}^{O_2}$ and $F(O_2) \notin \mathrm{P}^{S,O_2}$. We have to see that expressions (11), (12), (13), and (14) imply statements (i), (ii), (iii), and (iv) of Theorem 6.7. For (11) and (12) this follows from the fact that $f \in \mathrm{PF}$ implies $f \in \mathrm{PF}^{O_2}$. Each language in NP is accepted by infinitely many machines NM_j. Therefore, if there exists a sparse language L such that L is many-one-complete for $\mathrm{NP}^{O_2} \cap \mathrm{SPARSE}$, then there exists a $j \geq 1$ such that $L = L(NM_j^{O_2})$ and $L \in \mathrm{SPARSE}_j$. This shows that expression (13) implies (iii). In (14) we actually should have $F(O_2) \notin \mathrm{P}^{S,O_2}$ since the reducing machine has access to the oracle O_2. However, since (i) holds and since $(O_2, \overline{O_2}) \in \mathrm{DisjNP}^{O_2}$, there exists an $f \in \mathrm{PF}$ with $f(O_2) \subseteq A(O_2) \subseteq S$ and $f(\overline{O_2}) \subseteq B(O_2) \subseteq \overline{S}$. Hence, $q \in O_2 \Leftrightarrow f(q) \in S$. So we can transform queries to O_2 into queries to S; i.e., it suffices to show $F(O_2) \notin \mathrm{P}^S$. By expression (14), the complete pair $(A(O_2), B(O_2))$ is not NP^{O_2}-hard; it follows that no disjoint NP^{O_2}-pair is NP^{O_2}-hard.

We define the following list $\mathcal{T}$ of requirements. At the beginning of the construction, $\mathcal{T}$ contains all pairs (i,n) with $i \in \{1,2,3,4\}$ and $n \in \mathbb{N}^+$. These pairs have the following interpretations, which correspond to statements (i)–(iv) of Theorem 6.7:

- $(1, \langle i, j\rangle)$: ensure

$$L(NM_i^{O_2}) \cap L(NM_j^{O_2}) \neq \emptyset$$

or

$$(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2)).$$

- $(2, i)$: ensure that there exists some n such that $[0^n \in C(O_2) \wedge T_i^{O_2}(0^n) \notin D(O_2)]$ or $[0^n \in D(O_2) \wedge T_i^{O_2}(0^n) \notin C(O_2)]$.
- $(3, \langle i, j\rangle)$: ensure either $L(NM_j^{O_2}) \notin \mathrm{SPARSE}_j$ or [for some n, $E_n(O_2)$ contains ≤ 2 words of every length, and $E_n(O_2)$ does not $\leq_m^{p,O_2}$-reduce to $L(NM_j^{O_2})$ via $f_i^{O_2}$] (in the construction, n does not depend on i; i.e., $(3, \langle i, j\rangle)$ and $(3, \langle i', j\rangle)$ use the same n).
- $(4, i)$: ensure that $(A(O_2), B(O_2))$ has a separator S such that $0^n \in F(O_2) \Leftrightarrow 0^n \notin L(M_i^S)$.

Once a requirement is satisfied, we delete it from the list. Conditions of the form $(2, \cdot)$ and $(4, \cdot)$ are reachable by the construction of one counterexample. In contrast, if we cannot reach $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) \neq \emptyset$ for a condition of the first type, then we have to ensure $(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$. Similarly, if we cannot reach $L(NM_j^{O_2}) \notin \mathrm{SPARSE}_j$ for a condition of the third type, then, for a suitable n, we have to ensure that $E_n(O_2)$ contains ≤ 2 words of every length. But these conditions cannot be reached by a finite segment of an oracle; instead they influence the whole remaining construction of the oracle. We have to encode answers to queries of the form "does x belong to $L(NM_i^{O_2})$ or to $L(NM_j^{O_2})$" into the oracle O_2, and we have to keep an eye on the number of elements of $E_n(O_2)$. For this reason we introduce the notion of (μ, k)-valid oracles. Here k is a natural number and μ is an injective, partial function $\mathbb{N}^+ \to \mathbb{N} \times \mathbb{N}^+$ that has a finite domain. Each (μ, k)-valid oracle is a subset of $\Sigma^{\leq k}$. If a pair $(0, j)$, $j \geq 1$, is in the range of μ, then this means that $L(NM_j^{O_2}) \in \mathrm{SPARSE}_j$ is forced, and therefore we must construct O_2 so that for a suitable n, $E_n(O_2)$ contains ≤ 2 words of every length. If a pair (i, j), $i, j \geq 1$, is in the range of μ, then $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) = \emptyset$ is forced, and therefore we must construct O_2 so that $(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$ holds. For the latter condition we have to encode certain information into O_2, and the number k says up to which level this encoding has been done. So (μ, k)-valid oracles should be considered as finite prefixes of oracles that contain these encodings. For the moment we postpone the formal definition of (μ, k)-valid oracles (Definition 6.9); instead we mention its essential properties, which we will prove later.

(a) The oracle $\emptyset$ is $(\emptyset, 0)$-valid.
(b) If X is a finite oracle that is (μ, k)-valid, then for all $\mu' \preceq \mu$, X is (μ', k)-valid.
(c) If O_2 is an oracle such that for some μ, ${O_2}^{\leq k}$ is (μ, k)-valid for infinitely many k, then the following hold:
   - $A(O_2) \cap B(O_2) = C(O_2) \cap D(O_2) = \emptyset$.
   - For all $(i, j) \in \mathrm{range}(\mu)$, if $i > 0$, then

$$(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$$

     via some $f \in \mathrm{PF}$.
   - For all $(n, 0, j) \in \mu$ it holds that $E_n(O_2)$ contains ≤ 2 words of every length and $L(NM_j^{O_2}) \in \mathrm{SPARSE}_j$.

Properties (a), (b), and (c) will be proved later in Propositions 6.10 and 6.11. Moreover, we will prove the following for all $i, j \geq 1$ and all (μ, k)-valid X. (Note that there is a correspondence between (i)–(iv) and P1–P4.)

P1: There exists an $l > k$ and a (μ', l)-valid $Y \supseteq_k X$, $\mu \preceq \mu'$ such that
- either for all $Z \supseteq_l Y$, $L(NM_i^Z) \cap L(NM_j^Z) \neq \emptyset$, or
- $(i, j) \in \text{range}(\mu')$.[4]

P2: There exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, if $C(Z) \cap D(Z) = \emptyset$, then $(C(Z), D(Z))$ does not $\leq_m^{pp,O_2}$-reduce to $(D(Z), C(Z))$ via T_i^Z.

P3: (a) There exists an $l > k$ and a (μ', l)-valid $Y \supseteq_k X$, $\mu \preceq \mu'$, such that
- either for all $Z \supseteq_l Y$, $L(NM_j^Z) \notin \text{SPARSE}_j$, or
- $(0, j) \in \text{range}(\mu')$.

(b) For every n, if $\mu(n) = (0, j)$, then there exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, $E_n(Z)$ does not $\leq_m^{p,Z}$-reduce to $L(NM_j^Z)$ via f_i^Z.

P4: There exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, if $A(Z) \cap B(Z) = \emptyset$, then there exists a separator S of $(A(Z), B(Z))$ such that $F(Z) \neq L(M_i^S)$.

We will prove properties P1, P2, P3(a), P3(b), and P4 in Propositions 6.21, 6.22, 6.23, 6.25, and 6.32, respectively.

We construct an ascending sequence of finite oracles $X_0 \subseteq_{k_0} X_1 \subseteq_{k_1} X_2 \subseteq_{k_2} \cdots$ such that each X_r is (μ_r, k_r)-valid, $k_0 < k_1 < k_2 < \cdots$, and $\mu_0 \preceq \mu_1 \preceq \mu_2 \preceq \cdots$. By definition, $O_2 = \bigcup_{r \geq 0} X_r$. By items (b) and (c), $A(O_2) \cap B(O_2) = C(O_2) \cap D(O_2) = \emptyset$ follows immediately. Note that for each $r \geq 0$ and $i \geq 1$ it holds that $X_{r+i} \supseteq_{k_r} X_r$ and $\mu_r \preceq \mu_{r+i}$.

1. $r := 0$, $k_r := 0$, $\mu_r := \emptyset$, and $X_r := \emptyset$. Then by (a), X_r is (μ_r, k_r)-valid.
2. Let e be the next requirement on $\mathcal{T}$.
   (a) If $e = (1, \langle i, j \rangle)$, then we apply property P1 to X_r. Define $k_{r+1} = l$, $\mu_{r+1} = \mu'$, and $X_{r+1} = Y$. Then $k_r < k_{r+1}$, $\mu_r \preceq \mu_{r+1}$, and $X_{r+1} \supseteq_{k_r} X_r$ is (μ_{r+1}, k_{r+1})-valid such that
   - either for all $Z \supseteq_{k_{r+1}} X_{r+1}$, $L(NM_i^Z) \cap L(NM_j^Z) \neq \emptyset$, or
   - $(i, j) \in \text{range}(\mu_{r+1})$.

   Remove e from $\mathcal{T}$ and go to step 3.

   Comment: If the former holds, then, since $O_2 \supseteq_{k_{r+1}} X_{r+1}$, it holds that $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) \neq \emptyset$, and therefore $(L(NM_i^{O_2}), L(NM_j^{O_2})) \notin \text{DisjNP}^{O_2}$. Otherwise, $(i, j) \in \text{range}(\mu_{r+1})$. By (b), for all $i \geq 1$, X_{r+i} is (μ_{r+1}, k_{r+i})-valid. Therefore, by (c), $(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$ via some $f \in \text{PF}$.

   (b) If $e = (2, i)$, then $\mu_{r+1} \stackrel{df}{=} \mu_r$ and apply property P2 to X_r. We define $k_{r+1} = l$ and $X_{r+1} = Y$. Then $k_{r+1} > k_r$ and $X_{r+1} \supseteq_{k_r} X_r$ is (μ_{r+1}, k_{r+1})-valid so that for all $Z \supseteq_{k_{r+1}} X_{r+1}$, if $C(Z) \cap D(Z) = \emptyset$, then $(C(Z), D(Z))$ does not $\leq_m^{pp,O_2}$-reduce to $(D(Z), C(Z))$ via T_i^Z. Remove e from $\mathcal{T}$ and go to step 3.

   Comment: Since $O_2 \supseteq_{k_{r+1}} X_{r+1}$ and $C(O_2) \cap D(O_2) = \emptyset$, this ensures that $(C(O_2), D(O_2))$ does not $\leq_m^{pp,O_2}$-reduce to $(D(O_2), C(O_2))$ via $T_i^{O_2}$.

   (c) If $e = (3, \langle i, j \rangle)$ and $(0, j) \notin \text{range}(\mu_r)$, then we apply property P3(a) to X_r. Define $k_{r+1} = l$, $\mu_{r+1} = \mu'$, and $X_{r+1} = Y$. Then $k_r < k_{r+1}$, $\mu_r \preceq \mu_{r+1}$, and $X_{r+1} \supseteq_{k_r} X_r$ is (μ_{r+1}, k_{r+1})-valid such that
   - either for all $Z \supseteq_{k_{r+1}} X_{r+1}$, $L(NM_j^Z) \notin \text{SPARSE}_j$, or

[4] Proposition 6.21 says $L(NM_i^Z) \cap L(NM_j^Z) \cap \Sigma^{\leq l} \neq \emptyset$, which is a stronger statement.

- $(0, j) \in \text{range}(\mu_{r+1})$.

If the former holds, then remove e from $\mathcal{T}$ and go to step 3. Otherwise, do not remove e from $\mathcal{T}$ (it will be removed in the next iteration) and go to step 3.

Comment: If the former of the two alternatives holds, then, since $O_2 \supseteq_{k_{r+1}} X_{r+1}$, it holds that $L(NM_j^{O_2}) \notin \text{SPARSE}_j$. Otherwise, for a suitable n, $(n, 0, j) \in \mu_{r+1}$. By (b), for all $i \geq 1$, X_{r+i} is (μ_{r+1}, k_{r+i})-valid. Therefore, by (c), it is enforced that $E_n(O_2)$ contains ≤ 2 words of every length and $L(NM_j^{O_2}) \in \text{SPARSE}_j$. From now on, all requirements of the form $(3, \langle \cdot, j \rangle)$ are treated in step 2(d). These steps will make sure that $E_n(O_2) \not\leq_m^{p,O_2} L(NM_j^{O_2})$.

(d) If $e = (3, \langle i, j \rangle)$ and $(0, j) \in \text{range}(\mu_r)$, then choose n such that $(n, 0, j) \in \mu_r$ and apply property P3(b) to X_r. Define $k_{r+1} = l$, $\mu_{r+1} = \mu_r$, and $X_{r+1} = Y$. Then $k_r < k_{r+1}$, $\mu_r \preceq \mu_{r+1}$, and $X_{r+1} \supseteq_{k_r} X_r$ is (μ_{r+1}, k_{r+1})-valid such that for all $Z \supseteq_{k_{r+1}} X_{r+1}$, $E_n(Z)$ does not $\leq_m^{p,Z}$-reduce to $L(NM_j^Z)$ via f_i^Z. Remove e from $\mathcal{T}$ and go to step 3.

Comment: In the comment of the previous step we have seen that $(0, j) \in \text{range}(\mu_r)$ implies that $E_n(O_2) \in \text{SPARSE}_{j+1}$. Since $O_2 \supseteq_{k_{r+1}} X_{r+1}$ this step ensures that $E_n(O_2)$ does not $\leq_m^{p,O_2}$-reduce to $L(NM_j^{O_2})$ via $f_i^{O_2}$.

(e) If $e = (4, i)$, then $\mu_{r+1} \stackrel{df}{=} \mu_r$ and apply property P4 to X_r. We define $k_{r+1} = l$ and $X_{r+1} = Y$. Then $k_{r+1} > k_r$ and $X_{r+1} \supseteq_{k_r} X_r$ is (μ_{r+1}, k_{r+1})-valid such that for all $Z \supseteq_{k_{r+1}} X_{r+1}$, if $A(Z) \cap B(Z) = \emptyset$, then there exists a separator S of $(A(Z), B(Z))$ such that $F(Z) \neq L(M_i^S)$. Remove e from $\mathcal{T}$ and go to step 3.

Comment: Since $O_2 \supseteq_{k_{r+1}} X_{r+1}$ and $A(O_2) \cap B(O_2) = \emptyset$, this ensures that there exists a separator S of $(A(O_2), B(O_2))$ such that $F(O_2) \neq L(M_i^S)$.

3. $r := r + 1$, go to step 2.

We see that this construction ensures (i), (ii), (iii), and (iv). This proves Theorem 6.7 except to show that we can define an appropriate notion of a (μ, k)-valid oracle that has properties (a), (b), (c) and P1, P2, P3, P4.

We want to construct our oracle such that $(A(O_2), B(O_2))$ is a $\leq_{sm}^{pp}$-complete disjoint NP^{O_2}-pair. So we have to make sure that pairs $(L(NM_i), L(NM_j))$ that are enforced to be disjoint (which means that $(i, j) \in \text{range}(\mu)$) can be $\leq_{sm}^{pp}$-reduced to $(A(O_2), B(O_2))$. Therefore, we put certain codewords into O_2 if and only if the computation $NM_i^{O_2}(x)$ (resp., $NM_j^{O_2}(x)$) accepts within t steps.

DEFINITION 6.8 (μ-codeword). *Let $\mu : \mathbb{N}^+ \to \mathbb{N} \times \mathbb{N}^+$ be an injective, partial function with a finite domain. A word w is called a* μ-codeword *if $w = 00^n10^t1xy$ or $w = 10^n10^t1xy$ such that $n, t \geq 1$, $|y| = 3|00^n10^t1x|$, and $\mu(n) = (i, j)$ such that $i, j \geq 1$. If $w = 00^n10^t1xy$, then we say that w is a μ-codeword for (i, t, x); if $w = 10^n10^t1xy$, then we say it is a μ-codeword for (j, t, x).*

Condition (i) of Theorem 6.7 opposes conditions (ii), (iii), and (iv), because for (i) we have to encode information about NP^{O_2} computations into O_2, and (ii), (iii), and (iv) say that we cannot encode too much information (e.g., enough information for $\text{UP}^{O_2} = \text{NP}^{O_2}$). For this reason we have to look at certain finite oracles that contain the needed information for (i) and that allow all diagonalization needed to reach (ii), (iii), and (iv). We call such oracles (μ, k)-valid.

DEFINITION 6.9 ((μ, k)-valid oracle). *Let $k \geq 0$ and let $\mu : \mathbb{N}^+ \to \mathbb{N} \times \mathbb{N}^+$ be an injective, partial function with a finite domain. We define a finite oracle X to be (μ, k)-valid by induction over the size of the domain of μ.*

(IB) *If $\|\mu\| = 0$, then X is (μ, k)-valid $\stackrel{df}{\iff}$ $X \subseteq \Sigma^{\leq k}$ and $A(X) \cap B(X) = C(X) \cap D(X) = \emptyset$.*

(IS) *If* $\|\mu\| > 0$, *then* $\mu = \mu_0 \cup \{(n_0, i_0, j_0)\}$, *where* $n_0 = \mu_{\max}$ *and* $\mu_0 \prec \mu$. X *is* (μ, k)*-valid* $\overset{df}{\iff}$ $k \geq n_0$, X *is* (μ_0, k)*-valid, and the following holds:*

1. *If* $i_0 > 0$, *then we demand the following:*
   (a) *For all* $t \geq 1$ *and all* $x \in \Sigma^*$, *if* $4 \cdot |00^{n_0}10^t1x| \leq k$, *then*
      (i) $(\exists y,\ |y| = 3|00^{n_0}10^t1x|)[00^{n_0}10^t1xy \in X] \Leftrightarrow NM_{i_0}^X(x)$ *accepts within* t *steps, and*
      (ii) $(\exists y,\ |y| = 3|10^{n_0}10^t1x|)[10^{n_0}10^t1xy \in X] \Leftrightarrow NM_{j_0}^X(x)$ *accepts within* t *steps.*
   (b) *For all* $l \geq n_0$ *and all* (μ_0, l)*-valid* Y, *if* $Y^{\leq n_0} = X^{\leq n_0}$, *then* $L(NM_{i_0}^Y) \cap L(NM_{j_0}^Y) \cap \Sigma^{\leq l} = \emptyset$.
2. *If* $i_0 = 0$, *then*
   (a) *for every* $r \geq 0$, $\|E_{n_0}(X) \cap \Sigma^r\| \leq 2$, *and*
   (b) *for all* $l \geq n_0$ *and all* (μ_0, l)*-valid* Y, *if* $Y^{\leq n_0} = X^{\leq n_0}$, *then* $L(NM_{j_0}^Y) \cap \Sigma^{\leq l} \in \mathrm{SPARSE}_{j_0}$.

Due to conditions 1(b) and 2(b), (μ, k)-valid oracles can be extended to (μ, k')-valid oracles with $k' > k$ (Lemma 6.17). There we really need the intersection with $\Sigma^{\leq l}$. Otherwise—for example, in 1(b)—it could be possible that for a small oracle $Y \subseteq \Sigma^{\leq l}$ both machines accept the same word w that is much longer than l, but there is no way to extend Y in a valid way to the level $|w|$ such that both machines still accept w (the reason is that the reservations (Definition 6.12) become too large).

PROPOSITION 6.10 (basic properties of validity).

1. *The oracle* $\emptyset$ *is* $(\emptyset, 0)$*-valid (property* (a)*).*
2. *For every* (μ, k)*-valid* X *and every* $\mu' \preceq \mu$, X *is* (μ', k)*-valid (property* (b)*).*
3. *For every* (μ, k)*-valid* X *and every* $(n, 0, j) \in \mu$, *it holds that*
   (a) *for every* $r \geq 0$, $\|E_n(X) \cap \Sigma^r\| \leq 2$, *and*
   (b) $L(NM_j^X) \cap \Sigma^{\leq k} \in \mathrm{SPARSE}_j$.
4. *Let* X *be* (μ, k)*-valid and* $S \subseteq \Sigma^{k+1}$ *such that* $k + 1 \not\equiv 0 \pmod 4$, $C(S) \cap D(S) = \emptyset$, *and for all* $(n, 0, j) \in \mu$ *it holds that* $\|E_n(S)\| \leq 2$. *Then* $X \cup S$ *is* $(\mu, k+1)$*-valid.*
5. *For every* (μ, k)*-valid* X *and every* $(i, j) \in \mathrm{range}(\mu)$, $i > 0$, *it holds that* $L(NM_i^X) \cap L(NM_j^X) \cap \Sigma^{\leq k} = \emptyset$.
6. *If* X *is* (μ, k)*-valid, then for every* k', $\mu_{\max} \leq k' \leq k$ *(resp.,* $0 \leq k' \leq k$ *if* $\mu = \emptyset$*), it holds that* $X^{\leq k'}$ *is* (μ, k')*-valid.*

Proof. Statements 6.10.1 and 6.10.2 follow immediately from Definition 6.9.

Let X be (μ, k)-valid and let $(n, 0, j) \in \mu$. Let $n_0 \overset{df}{=} n$, $i_0 \overset{df}{=} 0$, $j_0 \overset{df}{=} j$, and $\mu_0 \overset{df}{=} \{(n', i', j') \in \mu \mid n' < n\}$. By 6.10.2, X is $(\mu_0 \cup \{(n_0, i_0, j_0)\}, k)$-valid and also (μ_0, k)-valid. From 6.9.2(a) it follows that 6.10.3(a) holds. From 6.9.2(b) (for $l = k$ and $Y = X$) we obtain $L(NM_{j_0}^X) \cap \Sigma^{\leq k} \in \mathrm{SPARSE}_{j_0}$. This shows 6.10.3(b).

We prove statement 6.10.4 by induction on $\|\mu\|$. First of all we see that $A(S) = B(S) = \emptyset$, since S contains no words of length $\equiv 0 \pmod 4$. If $\|\mu\| = 0$, then, by Definition 6.9, $X \cup S$ is $(\mu, k+1)$-valid. So assume $\|\mu\| > 0$ and choose μ_0, n_0, i_0, j_0 as in Definition 6.9. We assume as an induction hypothesis that if X is (μ_0, k)-valid, then $X \cup S$ is $(\mu_0, k+1)$-valid. We verify Definition 6.9 for $X \cup S$ and $k+1$. Clearly, $k + 1 > k \geq n_0$. Since X is (μ, k)-valid it is also (μ_0, k)-valid. By the induction hypothesis we obtain that $X \cup S$ is $(\mu_0, k+1)$-valid.

Assume that $i_0 > 0$; we verify item 1 of Definition 6.9. Since $k + 1 \not\equiv 0 \pmod 4$, the condition $4 \cdot |00^{n_0}10^t1x| \leq k + 1$ is equivalent to $4 \cdot |00^{n_0}10^t1x| \leq k$. Since $t < k$, the computations mentioned in 6.9.1(a) cannot ask queries longer than k. So nothing changes when these machines use oracle X instead of $X \cup S$. Moreover, at the left-

hand sides in 6.9.1(a), we can also use X instead of $X \cup S$ since we only test the membership for words of length $\equiv 0 \pmod 4$. This shows that in 6.9.1(a) we can replace every occurrence of $X \cup S$ with X and obtain an equivalent condition. This condition holds since X is (μ, k)-valid. Therefore, 6.9.1(a) holds for $X \cup S$ and $k+1$. Condition 6.9.1(b) holds for $X \cup S$ and $k+1$, since this condition does not depend on k and since $(X \cup S) \cap \Sigma^{\leq k} = X^{\leq k}$.

Assume that $i_0 = 0$; we verify item 2 of Definition 6.9. By assumption, $\|E_{n_0}(S)\| \leq 2$ and (since X is (μ, k)-valid) for all $r \geq 0$, it holds that $\|E_{n_0}(X) \cap \Sigma^r\| \leq 2$. Words in $E_{n_0}(X)$ are of length $\leq \lfloor k/2 \rfloor$. In contrast, words in $E_{n_0}(S)$ are of length $\lceil (k+1)/2 \rceil$. Hence, words in $E_{n_0}(X)$ are shorter than words in $E_{n_0}(S)$. So for all $r \geq 0$,

$$\begin{aligned}\|E_{n_0}(X \cup S) \cap \Sigma^r\| &= \|(E_{n_0}(X) \cap \Sigma^r) \cup (E_{n_0}(S) \cap \Sigma^r)\| \\ &= \|(E_{n_0}(X) \cap \Sigma^r)\| + \|(E_{n_0}(S) \cap \Sigma^r)\| \leq 2.\end{aligned}$$

This shows 6.9.2(a). Condition 6.9.2(b) holds for $X \cup S$ and $k+1$, since this condition does not depend on k, and since $(X \cup S) \cap \Sigma^{\leq k} = X^{\leq k}$. This proves statement 6.10.4.

We prove statement 5 of Proposition 6.10 as follows. Let X be (μ, k)-valid and $(i_0, j_0) \in \text{range}(\mu)$ such that $i_0 > 0$. Choose n_0 such that $(n_0, i_0, j_0) \in \mu$. Let $\mu_0 \stackrel{df}{=} \{(n', i', j') \in \mu \mid n' < n_0\}$. By 6.10.2, X is $(\mu_0 \cup \{(n_0, i_0, j_0)\}, k)$-valid and also (μ_0, k)-valid. Together with 6.9.1(b) (for $l = k$ and $Y = X$) this implies that $L(NM_{i_0}^X) \cap L(NM_{j_0}^X) \cap \Sigma^{\leq k} = \emptyset$.

We prove statement 6 of Proposition 6.10 by induction on $\|\mu\|$. If $\|\mu\| = 0$, then, by Definition 6.9, $X^{\leq k'}$ is (μ, k')-valid for $0 \leq k' \leq k$. So assume $\|\mu\| > 0$ and choose μ_0, n_0, i_0, j_0 as in Definition 6.9. We assume as an induction hypothesis that if X is (μ_0, k)-valid, then, for every k', $n_0 \leq k' \leq k$, it holds that $X^{\leq k'}$ is (μ_0, k')-valid. Choose k' such that $n_0 \leq k' \leq k$; we show that $X^{\leq k'}$ is (μ, k')-valid. Since X is (μ, k)-valid it is also (μ_0, k)-valid. By the induction hypothesis we obtain that $X^{\leq k'}$ is (μ_0, k')-valid.

Assume that $i_0 > 0$; we verify Definition 6.9.1. Note that in 6.9.1(a) we have the condition $4 \cdot |00^{n_0}10^t1x| \leq k'$. Hence, $t < k'$, and therefore the computations mentioned in 6.9.1(a) cannot ask queries longer than k'. So nothing changes when these machines use oracle X instead of $X^{\leq k'}$. Moreover, at the left-hand sides in 6.9.1(a), we can also use X instead of $X^{\leq k'}$ since we only test the membership for words of length $\leq k'$. This shows that in 6.9.1(a) we can replace every occurrence of $X^{\leq k'}$ with X and obtain an equivalent condition. This condition holds since X is (μ, k)-valid. Therefore, 6.9.1(a) holds. Condition 6.9.1(b) holds, since $X^{\leq k'} \cap \Sigma^{\leq n_0} = X^{\leq n_0}$.

Assume that $i_0 = 0$; we verify Definition 6.9.2. Condition 6.9.2(a) follows immediately, since X is (μ, k)-valid. Condition 6.9.2(b) holds, since $X^{\leq k'} \cap \Sigma^{\leq n_0} = X^{\leq n_0}$. This proves statement 6 of Proposition 6.10. □

PROPOSITION 6.11. *Let O_2 be an oracle such that for some μ there exist infinitely many k such that ${O_2}^{\leq k}$ is (μ, k)-valid (property* (c)*).*

1. $A(O_2) \cap B(O_2) = C(O_2) \cap D(O_2) = \emptyset$.
2. *For all $(i, j) \in \text{range}(\mu)$, $i > 0$, it holds that $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) = \emptyset$ and there exists some $f \in \text{PF}$ such that $(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$ via f.*
3. *For all $(n, 0, j) \in \mu$ it holds that $E_n(O_2)$ contains ≤ 2 words of every length, and $L(NM_j^{O_2}) \in \text{SPARSE}_j$.*

Proof. Assume that $A(O_2) \cap B(O_2) \neq \emptyset$ and let $w \in A(O_2) \cap B(O_2)$. Then, for $k = 4 \cdot (|w| + 1)$, w is already in $A({O_2}^{\leq k}) \cap B({O_2}^{\leq k})$. This contradicts the

assumption that there exists a $k' \geq k$ such that ${O_2}^{\leq k'}$ is (μ, k')-valid. Therefore, $A(O_2) \cap B(O_2) = \emptyset$. Analogously we see that $C(O_2) \cap D(O_2) = \emptyset$. This shows item 1 of Proposition 6.11.

Let $(i,j) \in \text{range}(\mu)$, $i > 0$, and choose n such that $(n,i,j) \in \mu$. Assume $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) \neq \emptyset$, and let $w \in L(NM_i^{O_2}) \cap L(NM_j^{O_2})$. Then, for $k = |w|^{i+j}$, w is already in $L(NM_i^{{O_2}'}) \cap L(NM_j^{{O_2}'}) \cap \Sigma^{\leq k}$, where ${O_2}' \stackrel{df}{=} {O_2}^{\leq k}$. By our assumption there exists a $k' \geq k$ such that ${O_2}'' \stackrel{df}{=} {O_2}^{\leq k'}$ is (μ, k')-valid. It follows that $w \in L(NM_i^{{O_2}''}) \cap L(NM_j^{{O_2}''}) \cap \Sigma^{\leq k'}$. This contradicts Proposition 6.10.5, and therefore $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) = \emptyset$.

Let $\mu_0 \stackrel{df}{=} \{(n',i',j') \in \mu \mid n' < n\}$. From our assumption and Proposition 6.10.2 it follows that for infinitely many k, ${O_2}^{\leq k}$ is $(\mu_0 \cup \{(n,i,j)\}, k)$-valid. So by Definition 6.9, for infinitely many k the following holds: For all $t \geq 1$ and all $x \in \Sigma^*$, if $4 \cdot |00^n10^t1x| \leq k$, then

- $(\exists y,\ |y| = 3|00^n10^t1x|)[00^n10^t1xy \in {O_2}^{\leq k}] \Leftrightarrow NM_i^{{O_2}^{\leq k}}(x)$ accepts within t steps, and
- $(\exists y,\ |y| = 3|10^n10^t1x|)[10^n10^t1xy \in {O_2}^{\leq k}] \Leftrightarrow NM_j^{{O_2}^{\leq k}}(x)$ accepts within t steps.

During the first t steps a machine can ask queries of length $\leq t < k$ only. Therefore, above we can replace $NM_i^{{O_2}^{\leq k}}(x)$ and $NM_j^{{O_2}^{\leq k}}(x)$ by $NM_i^{O_2}(x)$ and $NM_j^{O_2}(x)$, respectively. Moreover, since we have the condition $4 \cdot |00^n10^t1x| \leq k$, we can replace ${O_2}^{\leq k}$ with O_2 on the left-hand sides. Since the resulting condition holds for infinitely many k, the following holds for all $t \geq 1$ and $x \in \Sigma^*$:

- $(\exists y,\ |y| = 3|00^n10^t1x|)[00^n10^t1xy \in O_2] \Leftrightarrow NM_i^{O_2}(x)$ accepts within t steps.
- $(\exists y,\ |y| = 3|10^n10^t1x|)[10^n10^t1xy \in O_2] \Leftrightarrow NM_j^{O_2}(x)$ accepts within t steps.

The left-hand sides of these equivalences say $0^n10^t1x \in A(O_2)$ and $0^n10^t1x \in B(O_2)$, respectively. This shows that $(L(NM_i^{O_2}), L(NM_j^{O_2})) \leq_{sm}^{pp} (A(O_2), B(O_2))$ via some $f \in \text{PF}$.[5] Hence statement 2 of Proposition 6.11 holds.

Let $(n,0,j) \in \mu$. Assume that there exists an $r \geq 0$ such that $\|E_n(O_2) \cap \Sigma^r\| \geq 3$. Then there exists some k such that $\|E_n({O_2}') \cap \Sigma^r\| \geq 3$, where ${O_2}' \stackrel{df}{=} {O_2}^{\leq k}$. By our assumption there exists some $k' \geq k$ such that ${O_2}'' \stackrel{df}{=} {O_2}^{\leq k'}$ is (μ, k')-valid. It follows that $\|E_n({O_2}'') \cap \Sigma^r\| \geq 3$. This contradicts Proposition 6.10.3(a), and therefore $E_n(O_2)$ contains at most two words of every length.

Assume that $L(NM_j^{O_2}) \notin \text{SPARSE}_j$. Then there exists some m such that $L(NM_j^{O_2}) \cap \Sigma^m$ contains more than $m^j + j$ words. Therefore, with $k \stackrel{df}{=} m^j$ and ${O_2}' \stackrel{df}{=} {O_2}^{\leq k}$ we obtain $L(NM_j^{{O_2}'}) \cap \Sigma^{\leq k} \notin \text{SPARSE}_j$. By our assumption there exists some $k' \geq k$ such that ${O_2}'' \stackrel{df}{=} {O_2}^{\leq k'}$ is (μ, k')-valid. It follows that $L(NM_j^{{O_2}''}) \cap \Sigma^{\leq k'} \notin \text{SPARSE}_j$. This contradicts Proposition 6.10.3(b), and therefore $L(NM_j^{O_2}) \in \text{SPARSE}_j$. □

Remember that our construction consists of a coding part to obtain condition (i) of Theorem 6.7 and of separating parts to obtain conditions (ii), (iii), and (iv). In order to diagonalize, we will fix certain words that are needed for the coding part, and we will change our oracle on nonfixed positions to obtain the separation. For this we introduce the notion of a reservation for an oracle. A reservation consists of two sets Y and N, where Y contains words that are reserved for the oracle while N

[5] We can use $f(x) \stackrel{df}{=} 0^n10^{|x|^{i+j}}1x$, since $NM_i(x)$ and $NM_j(x)$ have computation times $|x|^i$ and $|x|^j$, respectively.

contains words that are reserved for the complement of the oracle. This notion has two important properties:

- Whenever an oracle X agrees with a reservation that is not too large, we can find an extension of X that agrees with the reservation (Lemma 6.14).
- If we want to fix certain words to be in the oracle, then this is possible using a reservation of small size. For this reason we can fix certain words to be in the oracle and still be able to diagonalize (Lemma 6.18).

DEFINITION 6.12 ((μ,k)-reservation). *A pair (Y,N) of finite sets is a (μ,k)-reservation for X if X is (μ,k)-valid, $Y\cap N=\emptyset$, $Y^{\leq k}\subseteq X$, $N^{\leq k}\subseteq \overline{X}$, $A(Y)\cap B(Y)=\emptyset$, all words in $Y^{>k}$ are of length $\equiv 0 \pmod 4$, and if $w\in Y^{>k}$ is a μ-codeword for (i,t,x), then $NM_i(x)$ has a positive path P such that $|P|\leq t$, $P^{\text{yes}}\subseteq Y$, and $P^{\text{no}}\subseteq N$.*

PROPOSITION 6.13 (basic properties of reservations). *The following holds for every (μ,k)-valid X:*

1. *$(\emptyset,\emptyset)$ is a (μ,k)-reservation for X.*
2. *If (Y,N) is a (μ,k)-reservation for X, then also $(Y,N\cup N')$ for every $N'\subseteq \overline{Y\cup X}$.*
3. *For every $N\subseteq\overline{X}$, $(\emptyset,N)$ is a (μ,k)-reservation for X.*
4. *Let (Y,N) be a (μ,k)-reservation for X. For each $(\mu,k+1)$-valid $Z\supseteq_k X$ such that $Y^{=k+1}\subseteq Z^{=k+1}\subseteq \overline{N}^{=k+1}$, it holds that (Y,N) is a $(\mu,k+1)$-reservation for Z.*
5. *Let (Y,N) be a (μ,k)-reservation for X. For every $m\geq 0$, $(Y\cap\Sigma^{\leq m}, N\cap\Sigma^{\leq m})$ is a (μ,k)-reservation for X.*

Proof. This follows immediately from Definition 6.12. □

Whenever a (μ,k)-reservation of some oracle X is not too large, then X has a (μ,m)-valid extension Z that agrees with the reservation.

LEMMA 6.14. *Let (Y,N) be a (μ,k)-reservation for X and let $m\stackrel{df}{=}\max(\{|w|\mid w\in Y\cup N\}\cup\{k\})$. If $\|N\|\leq 2^{k/2}$, then there exists a (μ,m)-valid $Z\supseteq_k X$ such that $Y\subseteq Z$, $N\subseteq\overline{Z}$, and $(Z-Y)\cap\Sigma^{>k}$ contains only μ-codewords.*

Proof. Assume $\|N\|\leq 2^{k/2}$. We show the lemma by induction on $n\stackrel{df}{=}m-k$. If $n=0$, then let $Z=X$ and we are done.

Now assume $n>0$. First of all we show that it suffices to find a $(\mu,k+1)$-valid $Z'\supseteq_k X$ such that $Y^{=k+1}\subseteq Z'^{=k+1}\subseteq\overline{N}^{=k+1}$ and $(Z'-Y)\cap\Sigma^{k+1}$ contains only μ-codewords. In this case, Proposition 6.13.4 implies that (Y,N) is a $(\mu,k+1)$-reservation for Z'. So we can apply the induction hypothesis to (Y,N) considered as a $(\mu,k+1)$-reservation for Z'. We obtain a (μ,m)-valid $Z\supseteq_{k+1}Z'$ such that $Y\subseteq Z$, $N\subseteq\overline{Z}$, and $(Z-Y)\cap\Sigma^{>k+1}$ contains only μ-codewords. Together this yields $Z\supseteq_k X$ and $(Z-Y)\cap\Sigma^{>k}$ contains only μ-codewords. It remains to find the mentioned Z'.

If $k+1\not\equiv 0 \pmod 4$, then $Y^{=k+1}=\emptyset$, since $Y^{=k+1}$ contains only words of length $\equiv 0 \pmod 4$. We apply Proposition 6.10.4 to $S\stackrel{df}{=}\emptyset$, and obtain that X is $(\mu,k+1)$-valid. Therefore, with $Z'\stackrel{df}{=}X$ we found the desired Z'.

If $k+1\equiv 0 \pmod 4$, then, starting with the empty set, we construct a set $S\subseteq\Sigma^{k+1}$ by doing the following for each $(n,i,j)\in\mu$, each $t\geq 1$, and each $x\in\Sigma^*$ such that $i>0$ and $4\cdot|00^n10^t1x|=k+1$:

- If $NM_i^X(x)$ accepts within t steps, then choose some $y\in\Sigma^{3|00^n10^t1x|}$ such that $00^n10^t1xy\notin N$. Add 00^n10^t1xy to S.
- If $NM_j^X(x)$ accepts within t steps, then choose some $y\in\Sigma^{3|10^n10^t1x|}$ such that $10^n10^t1xy\notin N$. Add 10^n10^t1xy to S.

Observe that the choices of words y are possible since $\|N\| \leq 2^{k/2} < 2^{3(k+1)/4} = \|\Sigma^{3|00^n10^t1x|}\|$. Moreover, S contains only μ-codewords. For $Z' \stackrel{df}{=} X \cup S \cup Y^{=k+1}$ we have $Z' \supseteq_k X$ and $Y^{=k+1} \subseteq Z'^{=k+1} \subseteq \overline{N}^{=k+1}$, since $S \subseteq \overline{N}^{=k+1}$. In addition, $(Z' - Y) \cap \Sigma^{k+1}$ contains only μ-codewords, since this set is a subset of S. It remains to show that Z' is $(\mu, k+1)$-valid.

CLAIM 6.15. $A(Z') \cap B(Z') = C(Z') \cap D(Z') = \emptyset$.

Proof. Since X is (μ, k)-valid we have $A(X) \cap B(X) = C(X) \cap D(X) = \emptyset$. When we look at the definitions of $A(X)$, $B(X)$, $C(X)$, and $D(X)$, we see that in order to show Claim 6.15, it suffices to show

$$A(Z') \cap B(Z') \cap \Sigma^{\frac{(k+1)}{4}-1} = C(Z') \cap D(Z') \cap \Sigma^{k+1} = \emptyset.$$

We immediately obtain $C(Z') \cap D(Z') \cap \Sigma^{k+1} = \emptyset$, since by definition, $C(Z')$ and $D(Z')$ contain only words of lengths $\equiv 1 \pmod 4$. Assume that $A(Z') \cap B(Z') \cap \Sigma^{(k+1)/4-1} \neq \emptyset$, and choose some $w \in A(Z') \cap B(Z') \cap \Sigma^{(k+1)/4-1}$. So there exist $n, t \geq 1$, $x \in \Sigma^*$, and $y_0, y_1 \in \Sigma^{3|w|+3}$ such that $w = 0^n10^t1x$ and $0wy_0, 1wy_1 \in Z'$. Note that $0wy_0, 1wy_1 \in S \cup Y^{=k+1}$, but both words cannot be in $Y^{=k+1}$, since otherwise we have $A(Y) \cap B(Y) \neq \emptyset$, which contradicts our assumption that (Y, N) is a (μ, k)-reservation. Therefore, either $0wy_0$ or $1wy_1$ belongs to S. Since all words in S are μ-codewords, there exist $i, j \geq 1$ such that $(n, i, j) \in \mu$. Hence $0wy_0$ and $1wy_1$ are μ-codewords. We claim that $NM_i^X(x)$ accepts within t steps, regardless of whether $0wy_0$ belongs to S or to $Y^{=k+1}$. This can be seen as follows:

- If $0wy_0 \in S$, then from the construction of S it follows that $NM_i^X(x)$ accepts within t steps.
- If $0wy_0 \in Y^{=k+1}$, then, since $0wy_0$ is a μ-codeword of length $> k$, $NM_i(x)$ has a positive path P with $|P| \leq t$, $P^{\text{yes}} \subseteq Y$, and $P^{\text{no}} \subseteq N$. Since $t \leq k$ it follows that $P^{\text{yes}} \cup P^{\text{no}} \subseteq \Sigma^{\leq k}$, and therefore $P^{\text{yes}} \subseteq X$ and $P^{\text{no}} \subseteq \Sigma^{\leq k} - X$. It follows that $NM_i^X(x)$ accepts within t steps.

Analogously we obtain that $NM_j^X(x)$ accepts within t steps. Since $|x| \leq k$ we have seen that $L(NM_i^X) \cap L(NM_j^X) \cap \Sigma^{\leq k} \neq \emptyset$ and $(i, j) \in \text{range}(\mu)$ such that $i > 0$. This contradicts Proposition 6.10.5 and finishes the proof of Claim 6.15. □

CLAIM 6.16. *Z' is $(\mu', k+1)$-valid for every $\mu' \preceq \mu$.*

Proof. We prove the claim by induction on $\|\mu'\|$. If $\|\mu'\| = 0$, then Z' is $(\mu', k+1)$-valid by Claim 6.15.

Assume now that $\|\mu'\| > 0$, and choose suitable μ_0, n_0, i_0, j_0 such that $n_0 = \mu'_{\max}$, $\mu' = \mu_0 \cup \{(n_0, i_0, j_0)\}$, and $\mu_0 \prec \mu'$. Clearly, $n_0 \leq \mu_{\max} \leq k < k+1$. As an induction hypothesis we assume that Z' is $(\mu_0, k+1)$-valid. We show that Z' is $(\mu', k+1)$-valid.

Assume $i_0 > 0$. We claim that for all $t \geq 1$ and all $x \in \Sigma^*$, if $4 \cdot |00^{n_0}10^t1x| \leq k+1$, then the equivalences in 6.9.1(a) hold for Z' instead of X. This is seen as follows:

- If $4 \cdot |00^{n_0}10^t1x| \leq k$, then they hold since X is (μ', k)-valid and $Z' \supseteq_k X$.
- If $4 \cdot |00^{n_0}10^t1x| = k+1$, then the implications "$\Leftarrow$" in statement 6.9.1(a) hold, since $NM_{i_0}^{Z'}(x)$ and $NM_{j_0}^{Z'}(x)$ run at most $t \leq k$ steps and can therefore use oracle X instead of Z', and because $S \subseteq Z'$. For the other direction, let $w = 0^{n_0}10^t1x$ and assume that there exists some $y \in \Sigma^{3|w|+3}$ such that $0wy \in Z'$. If $0wy \in S$, then we have put this word to S, because $NM_i^X(x)$ accepts within t steps. Since $t < k$, also $NM_i^{Z'}(x)$ accepts within t steps. So assume $0wy \in Y^{=k+1}$ and note that $0wy$ is a μ-codeword. Since (Y, N) is a (μ, k)-reservation for X, $NM_i(x)$ has a positive path P with $|P| \leq t$, $P^{\text{yes}} \subseteq Y$, and $P^{\text{no}} \subseteq N$. Since $t < k$, we have $P^{\text{yes}} \subseteq X$ and $P^{\text{no}} \subseteq \Sigma^{\leq k} - X$.

Hence, $NM_i^X(x)$ accepts within t steps, and therefore $NM_i^{Z'}(x)$ accepts within t steps. This shows the implication "$\Rightarrow$" in 6.9.1(a)(i). Analogously we see the implication "$\Rightarrow$" in 6.9.1(a)(ii).

Condition 6.9.1(b) holds for Z' instead of X, since X is (μ', k)-valid, $n_0 \leq k$ and therefore $Z'^{\leq n_0} = X^{\leq n_0}$.

Assume $i_0 = 0$. Since X is (μ', k)-valid, for all $r \geq 0$ it holds that $\|E_{n_0}(X) \cap \Sigma^r\| \leq 2$. Moreover, we have $E_{n_0}(Z' \cap \Sigma^{k+1}) = \emptyset$, since by definition, E_{n_0} depends only on oracle words of lengths $\equiv 2 \pmod 4$. Therefore, for all $r \geq 0$, $\|E_{n_0}(Z') \cap \Sigma^r\| \leq 2$. This shows 6.9.2(a). Condition 6.9.2(b) holds for Z' instead of X, since X is (μ', k)-valid, $n_0 \leq k$, and therefore $Z'^{\leq n_0} = X^{\leq n_0}$. This proves Claim 6.16. □

Claim 6.16 implies in particular that Z' is $(\mu, k+1)$-valid. This completes the proof of Lemma 6.14. □

One of the main consequences of Lemma 6.14 is that (μ, k)-valid oracles can be extended to (μ, k')-valid oracles for larger k'. We needed to include conditions 1(b) and 2(b) in Definition 6.9 in order to obtain this property. Otherwise it is possible that a certain way of extending the finite oracle X to some oracle X' has no extension to an infinite oracle O_2 so that $L(NM_i^{O_2}) \cap L(NM_j^{O_2}) = \emptyset$. If this happens, then by statement 6.9.1(a), for all extensions to an infinite oracle O_2, $A(O_2)$ and $B(O_2)$ would not be disjoint.

LEMMA 6.17. *If X is (μ, k)-valid, then for every $m > k$ there exists a (μ, m)-valid $Z \supseteq_k X$ such that $Z^{>k}$ contains only μ-codewords.*

Proof. It suffices to show the lemma for $m = k+1$. Let $Y = \emptyset$ and $N = \{0^{k+1}\}$. By Proposition 6.13.3, (Y, N) is a (μ, k)-reservation for X. Since $\|N\| = 1 \leq 2^{k/2}$ we can apply Lemma 6.14, and we obtain a $(\mu, k+1)$-valid $Z \supseteq_k X$ such that $Z^{>k}$ contains only μ-codewords. □

For a finite $X \subseteq \Sigma^*$, let $\ell(X) \stackrel{df}{=} \sum_{w \in X} |w|$.

LEMMA 6.18. *Let X be (μ, k)-valid and let $Z \supseteq_k X$ be (μ, m)-valid such that $m \geq k$ and $Z^{>k}$ contains only words of length $\equiv 0 \pmod 4$. For every $Y \subseteq Z$ and every $N \subseteq \overline{Z}$ there exists a (μ, k)-reservation (Y', N') for X such that $Y \subseteq Y'$, $N \subseteq N'$, $\ell(Y' \cup N') \leq 2 \cdot \ell(Y \cup N)$, $Y' \subseteq Z$, and $N' \subseteq \overline{Z}$.*

Proof. For every $Y \subseteq Z$ let

$$\mathcal{D}(Y) \stackrel{df}{=} \{q \mid Y^{>k} \text{ contains a } \mu\text{-codeword for } (i,t,x) \text{ and } q \in P^{\text{all}}_{i,t,x}\},$$

where $P_{i,t,x}$ is the lexicographically smallest path among all paths of $NM_i^Z(x)$ that are accepting and that are of length $\leq t$. Note that $\mathcal{D}(Y)$ is well-defined: If $Y^{>k} \subseteq Z$ contains a μ-codeword, then this has the form $00^{n_0}10^t1xy$ (resp., $10^{n_0}10^t1xy$), and there exist $i_0, j_0 \geq 1$ such that $(n_0, i_0, j_0) \in \mu$. Let $\mu_0 \stackrel{df}{=} \{(n', i', j') \in \mu \mid n' < n_0\}$. By statement 2 of Proposition 6.10, Z is $(\mu_0 \cup \{(n_0, i_0, j_0)\}, m)$-valid. From statement 6.9.1(a) it follows that the path $P_{i_0,t,x}$ (resp., $P_{j_0,t,x}$) exists.

If w is a μ-codeword for (i,t,x), then $|P_{i,t,x}| \leq t < |w|/4$. Therefore, when looking at the definition of $\mathcal{D}(Y)$, we see that the sum of lengths of q's that are induced by some μ-codeword w is at most $|w|/4$ (remember that we use nondeterministic machines that ask all queries in parallel). This shows the following.

CLAIM 6.19. *For all $Y \subseteq Z$, $\ell(\mathcal{D}(Y)) \leq \ell(Y)/4$, and words in $\mathcal{D}(Y)$ are not longer than the longest word in Y.*

Given Y and N, the procedure below computes the (μ, k)-reservation (Y', N').

```
1    Y_0 := Y
2    N_0 := N
3    c := 0
```

```
4    do
5        c := c + 1
6        Y_c := D(Y_{c-1}) ∩ Z
7        N_c := D(Y_{c-1}) ∩ Z̄
8    repeat until Y_c = N_c = ∅
9    Y' := Y_0 ∪ Y_1 ∪ ··· ∪ Y_c
10   N' := N_0 ∪ N_1 ∪ ··· ∪ N_c
```

Note that since all Y_c are subsets of Z, the expressions $\mathcal{D}(Y_{c-1})$ in lines 6 and 7 are defined. It is immediately clear that $Y \subseteq Y' \subseteq Z$ and $N \subseteq N' \subseteq \overline{Z}$. Therefore $Y' \cap N' = \emptyset$. From Claim 6.19 we obtain $\ell(Y_i \cup N_i) = \ell(\mathcal{D}(Y_{i-1})) \leq \ell(Y_{i-1})/4$ for $1 \leq i \leq c$. Therefore, the procedure terminates and $\ell(Y' \cup N') \leq 2 \cdot \ell(Y \cup N)$. It remains to show the following.

CLAIM 6.20. *(Y', N') is a (μ, k)-reservation for X.*

Clearly, $Y'^{\leq k} \subseteq X$ and $N'^{\leq k} \subseteq \overline{X}$. Moreover, $A(Y') \cap B(Y') = \emptyset$, since otherwise $A(Z) \cap B(Z) \neq \emptyset$, which is not possible, since Z is (μ, m)-valid. All words in $Y'^{>k}$ are of length $\equiv 0 \pmod 4$, since $Y' \subseteq Z$. Let $v \in Y'^{>k}$ be a μ-codeword for (i, t, x). More precisely, $v \in Y_{i'} \subseteq Z$ for a suitable $i' < c$. Z is (μ, m)-valid and v is a μ-codeword that belongs to Z. Therefore, as seen at the beginning of this proof, it follows that $NM_i^Z(x)$ accepts within t steps. Thus the path $P_{i,t,x}$ exists and we obtain $P^{\text{all}}_{i,t,x} \subseteq \mathcal{D}(Y_{i'})$. It follows that $P^{\text{yes}}_{i,t,x} \subseteq Y_{i'+1} \subseteq Y'$ and $P^{\text{no}}_{i,t,x} \subseteq N_{i'+1} \subseteq N'$. Therefore, $NM_i(x)$ has a positive path P with $|P| \leq t$, $P^{\text{yes}} \subseteq Y'$, and $P^{\text{no}} \subseteq N'$. This proves Claim 6.20 and finishes the proof of Lemma 6.18. □

For any (μ, k)-valid oracle either we can find a finite extension that makes the languages accepted by NM_i and NM_j not disjoint, or we can force these languages to be disjoint for all valid extensions.

PROPOSITION 6.21 (property P1). *Let $i, j \geq 1$ and let X be (μ, k)-valid. There exists an $l > k$ and a (μ', l)-valid $Y \supseteq_k X$, $\mu \preceq \mu'$ such that*

- *either for all $Z \supseteq_l Y$, $L(NM_i^Z) \cap L(NM_j^Z) \cap \Sigma^{\leq l} \neq \emptyset$, or*
- *$(i, j) \in \text{range}(\mu')$.*

This proposition tells us that if the first property does not hold, then by Definition 6.9, since Y is (μ', l)-valid, $L(NM_i^Z) \cap L(NM_j^Z) \cap \Sigma^{\leq m} = \emptyset$ for all (μ', m)-valid extensions Z of Y, where $m \geq l$.

Proof. By Lemma 6.17, we can assume that k is large enough so that $2 \cdot k^{i+j} < 2^{k/2}$. If $(i, j) \in \text{range}(\mu)$, then by Lemma 6.17, for $\mu' = \mu$ and $l = k + 1$ there exists a (μ', l)-valid $Y \supseteq_k X$. Otherwise we distinguish two cases.

Case 1. There exists an $l' > k$ and a (μ, l')-valid $Y' \supseteq_k X$ such that $L(NM_i^{Y'}) \cap L(NM_j^{Y'}) \cap \Sigma^{\leq l'} \neq \emptyset$. Choose some $x \in L(NM_i^{Y'}) \cap L(NM_j^{Y'}) \cap \Sigma^{\leq l'}$ and let P_i, P_j be accepting paths of the computations $NM_i^{Y'}(x)$, $NM_j^{Y'}(x)$, respectively. Note that $(P_i^{\text{yes}} \cup P_j^{\text{yes}}) \cap \Sigma^{>l'} = \emptyset$ and let $N \stackrel{\text{df}}{=} (P_i^{\text{no}} \cup P_j^{\text{no}}) \cap \Sigma^{>l'}$. By Proposition 6.13.3, $(\emptyset, N)$ is a (μ, l')-reservation for Y'. Since $\|N\| \leq 2 \cdot |x|^{i+j} \leq 2 \cdot l'^{i+j} < 2^{l'/2}$ we can apply Lemma 6.14. We obtain some $l \geq l' > k$ and some (μ, l)-valid $Y \supseteq_{l'} Y' \supseteq_k X$ such that $N \subseteq \Sigma^{\leq l}$ and $N \subseteq \overline{Y}$. Therefore, for every $Z \supseteq_l Y$ the computations $NM_i^Z(x)$ and $NM_j^Z(x)$ will accept at the paths P_i and P_j, respectively. Hence $L(NM_i^Z) \cap L(NM_j^Z) \cap \Sigma^{\leq l} \neq \emptyset$ for every $Z \supseteq_l Y$.

Case 2. For every $l' > k$ and every (μ, l')-valid $Y' \supseteq_k X$ it holds that $L(NM_i^{Y'}) \cap L(NM_j^{Y'}) \cap \Sigma^{\leq l'} = \emptyset$. By Lemma 6.17, there exists a (μ, l)-valid $Y \supseteq_k X$ where $l \stackrel{\text{df}}{=} k + 1$. Let $n_0 \stackrel{\text{df}}{=} l$, $i_0 \stackrel{\text{df}}{=} i$, $j_0 \stackrel{\text{df}}{=} j$, $\mu_0 \stackrel{\text{df}}{=} \mu$, and $\mu' \stackrel{\text{df}}{=} \mu_0 \cup \{(n_0, i_0, j_0)\}$. Observe that $n_0 > k \geq \mu_{\max}$, and therefore $\mu \preceq \mu'$. We show that Y is (μ', l)-valid.

We already know that $l \geq n_0$ and that Y is (μ_0, l)-valid. Since $i_0 > 0$ we only have to verify Definition 6.9.1. When looking at condition 6.9.1(a), we see that $4 \cdot |00^{n_0}10^t1x| \leq l$ is not possible, since $n_0 = l$. Therefore, condition 6.9.1(a) holds. Condition 6.9.1(b) follows from our assumption in Case 2. Therefore, Y is (μ', l)-valid. □

In order to show that $(C(O_2), D(O_2))$ is not symmetric we have to diagonalize against every possible reducing function, i.e., against every deterministic polynomial-time oracle transducer. The following proposition makes sure that this diagonalization is compatible with the notion of valid oracles.

PROPOSITION 6.22 (property P2). *Let $i \geq 1$ and let X be (μ, k)-valid. There exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, if $C(Z) \cap D(Z) = \emptyset$, then $(C(Z), D(Z))$ does not $\leq_m^{pp,O_2}$-reduce to $(D(Z), C(Z))$ via T_i^Z.*

Proof. By Lemma 6.17 we can assume that $k \equiv 0 \pmod 4$ and $(k+1)^i + 1 < 2^{(k+1)/2}$. Consider the computation $T_i^X(0^{k+1})$, let x be the output of this computation, and let N be the set of queries that are of length greater than k. If $|x| > k$, then additionally we add the word $0^{|x|}$ to N. Note that this yields an N such that $X \cap N = \emptyset$ and $\|N\| \leq (k+1)^i + 1 < 2^{(k+1)/2}$.

If $x \in C(X)$ (note that this implies $x = 0^{k'}$ for some $k' \leq k$), then choose some $y \in 0\Sigma^k - N$ and let $S \stackrel{df}{=} \{y\}$. In this case it holds that $0^{k+1} \in C(X \cup S) \wedge x \notin D(X \cup S)$. The right part of the conjunction holds, since X is (μ, k)-valid, and therefore $C(X) \cap D(X) = \emptyset$. Otherwise, if $x \notin C(X)$, then choose some $y \in 1\Sigma^k - N$ and let $S \stackrel{df}{=} \{y\}$. Here we obtain $0^{k+1} \in D(X \cup S) \wedge x \notin C(X \cup S)$. Together this means that we find some $y \in \Sigma^{k+1} - N$ such that with $S \stackrel{df}{=} \{y\}$ it holds that

$$(15) \quad [0^{k+1} \in C(X \cup S) \wedge x \notin D(X \cup S)] \ \vee \ [0^{k+1} \in D(X \cup S) \wedge x \notin C(X \cup S)].$$

Note that $S \subseteq \Sigma^{k+1}$ and $k+1 \not\equiv 0 \pmod 4$. Moreover, $C(S) \cap D(S) = \emptyset$ and for every n, $E_n(S) = \emptyset$, since by definition E_n depends only on oracle words of length $\equiv 2 \pmod 4$. From Proposition 6.10.4 it follows that $X \cup S$ is $(\mu, k+1)$-valid. So by Proposition 6.13.3, $(\emptyset, N)$ is a $(\mu, k+1)$-reservation for $X \cup S$. Since $\|N\| < 2^{(k+1)/2}$ we can apply Lemma 6.14. For $l \stackrel{df}{=} \max(\{|w| \mid w \in N\} \cup \{k+1\})$ we obtain a (μ, l)-valid $Y \supseteq_{k+1} X \cup S$ such that $N \subseteq \overline{Y}$ and $Y^{>k+1}$ contains only words of length $\equiv 0 \pmod 4$. Therefore, $T_i^Y(0^{k+1})$ computes x. Since all queries asked at this computation are of length $\leq l$, we obtain that $T_i^Z(0^{k+1})$ computes x for every $Z \supseteq_l Y$. Since $Y^{>k+1}$ does not contain words of length $\equiv 1 \pmod 4$ we have $C(Z) \cap \Sigma^{\leq l} = C(X \cup S)$ and $D(Z) \cap \Sigma^{\leq l} = D(X \cup S)$ for each $Z \supseteq_l Y$. Note that $k+1 \leq l$ and $|x| \leq l$. Therefore, by equation (15), the following holds for every $Z \supseteq_l Y$:

$$(16) \quad [0^{k+1} \in C(Z) \wedge T_i^Z(0^{k+1}) \notin D(Z)] \ \vee \ [0^{k+1} \in D(Z) \wedge T_i^Z(0^{k+1}) \notin C(Z)].$$

Hence, for every $Z \supseteq_l Y$, if $C(Z) \cap D(Z) = \emptyset$, then $(C(Z), D(Z))$ does not $\leq_m^{pp,O_2}$-reduce to $(D(Z), C(Z))$ via T_i^Z. □

For any (μ, k)-valid oracle, either we can find a finite extension that destroys NM_j's promise to be sparse, or we can force NM_j to be sparse for all valid extensions.

PROPOSITION 6.23 (property P3(a)). *Let $j \geq 1$ and let X be (μ, k)-valid. There exists an $l > k$ and a (μ', l)-valid $Y \supseteq_k X$, $\mu \preceq \mu'$, such that*

- *either for all $Z \supseteq_l Y$, $L(NM_j^Z) \notin \mathrm{SPARSE}_j$, or*
- *$(0, j) \in \mathrm{range}(\mu')$.*

This proposition tells us that if the first property does not hold, then there exists some n such that $(n, 0, j) \in \mu'$. In this case, from Definition 6.9 we obtain that for all

(μ', m)-valid extensions Z of Y it holds that $L(NM_j^Z) \cap \Sigma^{\leq m} \in \mathrm{SPARSE}_j$ and $E_n(Z)$ contains at most 2 words of every length.

Proof. By Lemma 6.17, we can assume that k is large enough so that $(k^j + j + 1) \cdot k^j < 2^{k/2}$. If $(0, j) \in \mathrm{range}(\mu)$, then by Lemma 6.17, for $\mu' = \mu$ and $l = k + 1$ there exists a (μ', l)-valid $Y \supseteq_k X$. Otherwise we distinguish two cases.

Case 1. There exists an $l' > k$ and a (μ, l')-valid $Y' \supseteq_k X$ such that $L(NM_j^{Y'}) \cap \Sigma^{\leq l'} \notin \mathrm{SPARSE}_j$. More precisely, there exists an $m \leq l'$ such that $\|L(NM_j^{Y'}) \cap \Sigma^m\| > m^j + j$. We choose $m^j + j + 1$ different words $x_0, \ldots, x_{m^j+j}$ from $L(NM_j^{Y'}) \cap \Sigma^m$. For $0 \leq i \leq m^j + j$, let P_i be an accepting path of the computation $NM_j^{Y'}(x_i)$. For all i, note that $P_i^{\mathrm{yes}} \cap \Sigma^{>l'} = \emptyset$ and let N be the union of all $P_i^{\mathrm{no}} \cap \Sigma^{>l'}$. By Proposition 6.13.3, $(\emptyset, N)$ is a (μ, l')-reservation for Y'. Since $\|N\| \leq (m^j + j + 1) \cdot m^j \leq (l'^j + j + 1) \cdot l'^j < 2^{l'/2}$ we can apply Lemma 6.14. We obtain some $l \geq l' > k$ and some (μ, l)-valid $Y \supseteq_{l'} Y' \supseteq_k X$ such that $N \subseteq \Sigma^{\leq l}$ and $N \subseteq \overline{Y}$. Therefore, for every $Z \supseteq_l Y$ and every i, the computation $NM_j^Z(x_i)$ will accept at path P_i. Hence, for every $Z \supseteq_l Y$, $L(NM_j^Z) \notin \mathrm{SPARSE}_j$.

Case 2. For every $l' > k$ and every (μ, l')-valid $Y' \supseteq_k X$, it holds that $L(NM_j^{Y'}) \cap \Sigma^{\leq l'} \in \mathrm{SPARSE}_j$. By Lemma 6.17, there exists a (μ, l)-valid $Y \supseteq_k X$ with $l \stackrel{df}{=} k + 1$. Let $n_0 \stackrel{df}{=} l$, $i_0 \stackrel{df}{=} 0$, $j_0 \stackrel{df}{=} j$, $\mu_0 \stackrel{df}{=} \mu$, and $\mu' \stackrel{df}{=} \mu_0 \cup \{(n_0, i_0, j_0)\}$. Observe that $n_0 > k \geq \mu_{\max}$, and therefore $\mu_0 \preceq \mu'$. We will show that Y is (μ', l)-valid.

Since $l = \mu'_{\max}$ we have $l \geq \mu'_{\max}$. We already know $l \geq n_0$ and that Y is (μ_0, l)-valid. Since $i_0 = 0$, we only have to verify Definition 6.9.2. Since $l = n_0$ and $Y \subseteq \Sigma^{\leq l}$, we have $E_{n_0}(Y) = \emptyset$, which shows 6.9.2(a). Condition 6.9.2(b) follows from our assumption in Case 2. Therefore, Y is (μ', l)-valid. □

If NM_j is forced to be sparse for all valid extensions (Proposition 6.23), then we have to make sure that $L(NM_j)$ is not many-one-complete for NP ∩ SPARSE. We show that a certain E_n is sparse but is not many-one reducible to $L(NM_j)$. For this we have to diagonalize against every possible reducing function, i.e., against every deterministic polynomial-time oracle transducer. Proposition 6.25 makes sure that this diagonalization is possible. Before we give this proposition, we prove the following argument, which is used in the proofs for Proposition 6.25 and Lemma 6.29.

PROPOSITION 6.24. *Let X be (μ, k)-valid. Let (Y_1, N_1) be a $(\mu, k+1)$-reservation of some $(\mu, k+1)$-valid $Z_1 \supseteq_k X$, and let (Y_2, N_2) be a $(\mu, k+1)$-reservation of some $(\mu, k+1)$-valid $Z_2 \supseteq_k X$ such that ${Y_1}^{>k+1} \cup {Y_2}^{>k+1}$ contains only μ-codewords. If $\|N_1 \cup N_2\| \leq 2^{(k+1)/2}$, $Y_1 \cap N_2 = Y_2 \cap N_1 = \emptyset$, and $X' \stackrel{df}{=} X \cup {Y_1}^{=k+1} \cup {Y_2}^{=k+1}$ is $(\mu, k+1)$-valid, then $A(Y_1 \cup Y_2) \cap B(Y_1 \cup Y_2) = \emptyset$.*

Proof. In order to see that (Y_1, N_1) is a $(\mu, k+1)$-reservation for X', it suffices to show that ${Y_1}^{=k+1} \subseteq X'$ and ${N_1}^{=k+1} \subseteq \overline{X'}$. The first inclusion holds by the definition of X'. The second one holds, since otherwise either $Y_1 \cap N_1 \neq \emptyset$ (not possible since (Y_1, N_1) is a $(\mu, k+1)$-reservation) or $Y_2 \cap N_1 \neq \emptyset$ (not possible by assumption). It follows that (Y_1, N_1) is a $(\mu, k+1)$-reservation for X', and, analogously, (Y_2, N_2) is a $(\mu, k+1)$-reservation for X'.

Assume that $A(Y_1 \cup Y_2) \cap B(Y_1 \cup Y_2) \neq \emptyset$. Choose a shortest $w \in A(Y_1 \cup Y_2) \cap B(Y_1 \cup Y_2)$. Hence, there exist $y_0, y_1 \in \Sigma^{3|w|+3}$ such that $0wy_0, 1wy_1 \in Y_1 \cup Y_2$. Let $m \stackrel{df}{=} |0wy_0| - 1$. We show $m \geq k + 1$. Otherwise, if $m \leq k$, then $|0wy_0| = |1wy_1| \leq k + 1$. It follows that $0wy_0, 1wy_1 \in X'$, since (Y_1, N_1) and (Y_2, N_2) are $(\mu, k+1)$-reservations for X'. This implies $w \in A(X') \cap B(X')$, which is not possible. Therefore, $m \geq k + 1$.

By Proposition 6.13.5, $(Y_1^{\leq m}, N_1^{\leq m})$ and $(Y_2^{\leq m}, N_2^{\leq m})$ are $(\mu, k+1)$-reservations for X'. Let $Y \stackrel{df}{=} Y_1^{\leq m} \cup Y_2^{\leq m}$ and $N \stackrel{df}{=} N_1^{\leq m} \cup N_2^{\leq m}$. We show that (Y, N) is a $(\mu, k+1)$-reservation for X'. For this it suffices to verify $Y \cap N = \emptyset$ and $A(Y) \cap B(Y) = \emptyset$. The first equality holds, since otherwise either $Y_1 \cap N_2 \neq \emptyset$ or $Y_2 \cap N_1 \neq \emptyset$, which is not possible by assumption. If $A(Y) \cap B(Y) \neq \emptyset$, then there exists some $w' \in A(Y) \cap B(Y)$ such that $|w'| < |w|$. This is not possible, since $A(Y) \cap B(Y) \subseteq A(Y_1 \cup Y_2) \cap B(Y_1 \cup Y_2)$ and since w was chosen as short as possible. Therefore, (Y, N) is a $(\mu, k+1)$-reservation for X'.

Note that $\|N\| \leq 2^{(k+1)/2}$. By Lemmas 6.14 and 6.17, there exists a (μ, m)-valid $Z \supseteq_{k+1} X'$ such that $Y \subseteq Z$ and $N \subseteq \overline{Z}$. We know that $|0wy_0| > k+1$ and $0wy_0 \in Y_1 \cup Y_2$. Without loss of generality we assume $0wy_0 \in Y_1$. So by assumption, $0wy_0$ is a μ-codeword. Hence, $w = 0^n 10^t 1x$ for suitable n, t, x such that n is in the domain of μ. Let $\mu(n) = (i, j)$, where $i, j \geq 1$. From $0wy_0 \in Y_1$ it follows that $NM_i(x)$ has a positive path P such that $|P| \leq t$, $P^{\text{yes}} \subseteq Y_1$, and $P^{\text{no}} \subseteq N_1$. Since elements from P^{yes} and P^{no} are of length $\leq t \leq m$, we obtain $P^{\text{yes}} \subseteq Y \subseteq Z$, and $P^{\text{no}} \subseteq N \subseteq \overline{Z}$. It follows that $NM_i^Z(x)$ accepts. Analogously (i.e., with the help of $1wy_1$) we obtain that $NM_j^Z(x)$ accepts. This shows $x \in L(NM_i^Z) \cap L(NM_j^Z) \cap \Sigma^{\leq m}$, which contradicts Proposition 6.10.5. □

PROPOSITION 6.25 (property P3(b)). *Let $i, j \geq 1$ and let X be (μ, k)-valid such that for a suitable n, $\mu(n) = (0, j)$. There exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, $E_n(Z)$ does not $\leq_m^{p,Z}$-reduce to $L(NM_j^Z)$ via f_i^Z.*

Proof. Let $\alpha \stackrel{df}{=} (k+1)^i$, $\beta \stackrel{df}{=} (\alpha+1) \cdot (\alpha^j + j) + 1$, and $\gamma \stackrel{df}{=} \beta \cdot (2 \cdot \alpha^j + 2)$. Note that if i and j are considered as constants, then the values of α, β, and γ are polynomial in $k+1$. By Lemma 6.17, we can assume that $k \equiv 1 \pmod 4$, and that k is large enough such that $n + 2 + \log \gamma \leq (k+1)/2$ and $(2 \cdot \alpha^j + 2) \cdot \gamma < 2^{(k+1)/2}$.

Let $x_1, \ldots, x_\gamma$ be the binary representations (possibly with leading zeros) of $1, \ldots, \gamma$, respectively, such that for all r, $|0^n 1 x_r| = (k+1)/2$. For $1 \leq r \leq \gamma$, let $z_r \stackrel{df}{=} f_i^X(0^n 1 x_r)$ and note that the lengths of these words are bounded by α. We consider two cases.

Case 1. There exist a, b such that $1 \leq a < b \leq \gamma$ and $z_a = z_b$. Let N be the set of queries of length $> k$ that are asked during the computations $f_i^X(0^n 1 x_a)$ and $f_i^X(0^n 1 x_b)$. Note that these are negative queries. Observe that $\|N\| \leq 2 \cdot \alpha < 2^{(k+1)/2}$ and choose a word y_a of length $(k+1)/2$ such that $0^n 1 x_a y_a \notin N$. Let $S \stackrel{df}{=} \{0^n 1 x_a y_a\}$. It follows that $C(S) \cap D(S) = \emptyset$. Moreover, for all $n' \geq 1$, $\|E_{n'}(S)\| \leq 1$. From Proposition 6.10.4 it follows that $X' \stackrel{df}{=} X \cup S$ is $(\mu, k+1)$-valid. By Proposition 6.13.3, $(\emptyset, N)$ is a $(\mu, k+1)$-reservation for X'. By Lemma 6.14, there exists a (μ, l)-valid $Y \supseteq_{k+1} X'$ such that $N \subseteq \Sigma^{\leq l}$ and $N \subseteq \overline{Y}$. Therefore, for all $Z \supseteq_l Y$ it holds that $f_i^Z(0^n 1 x_a) = f_i^Z(0^n 1 x_b) = z_a$. Moreover, $0^n 1 x_a \in E_n(Z)$ and $0^n 1 x_b \notin E_n(Z)$. This shows that for all $Z \supseteq_l Y$, $E_n(Z)$ does not $\leq_m^{p,Z}$-reduce to $L(NM_j^Z)$ via f_i^Z.

Case 2. For $1 \leq r \leq \gamma$, all z_r are pairwise different. The remaining part of the proof deals with this case. Until the end of the proof, r will always be such that $1 \leq r \leq \gamma$. For every r, define the following set:

$$L_r \stackrel{df}{=} \{(Y_r, N_r) \mid (Y_r, N_r) \text{ is a } (\mu, k+1)\text{-reservation for some } (\mu, k+1)\text{-valid } Z \supseteq_k X \text{ such that } Z^{=k+1} \subseteq 0^n 1 \Sigma^*,\ \|Z^{=k+1}\| \leq 1,\ Y_r{}^{>k+1} \text{ contains only } \mu\text{-codewords},\ \ell(Y_r \cup N_r) \leq 2 \cdot \alpha^j,\ \text{and } NM_j(z_r) \text{ has a positive path } P_r \text{ such that } P_r^{\text{yes}} \subseteq Y_r \text{ and } P_r^{\text{no}} \subseteq N_r\}.$$

In the following we consider vectors $v = ((Y_{r_1}, N_{r_1}), (Y_{r_2}, N_{r_2}), \ldots, (Y_{r_s}, N_{r_s}))$ such that $1 \leq s \leq \beta$, all r_a are from $[1, \gamma]$ and are pairwise different, and $(Y_{r_a}, N_{r_a}) \in$

L_{r_a}. Such vectors v are called *vectors of reservations from* $L_1, \ldots, L_\gamma$. We say that v *has a conflict* if there exist a, b such that $1 \le a < b \le s$, and either $Y_{r_a} \cap N_{r_b} \neq \emptyset$ or $N_{r_a} \cap Y_{r_b} \neq \emptyset$. In this case we also say that the reservations (Y_{r_a}, N_{r_a}) and (Y_{r_b}, N_{r_b}) *conflict*. Now we are going to prove three claims. After this, with Claim 6.28 at hand, we are able to finish Case 2.

CLAIM 6.26. *Let* $(Y_a, N_a) \in L_a$ *and* $(Y_b, N_b) \in L_b$. *If* (Y_a, N_a) *and* (Y_b, N_b) *do not conflict, then* $A(Y_a \cup Y_b) \cap B(Y_a \cup Y_b) = \emptyset$.

Assume that (Y_a, N_a) and (Y_b, N_b) do not conflict. Let $S \stackrel{df}{=} {Y_a}^{=k+1} \cup {Y_b}^{=k+1}$ and $X' \stackrel{df}{=} X \cup S$. From the definition of L_a and L_b it follows that $\|S\| \le 2$. Therefore, for all $n' \ge 1$, $\|E_{n'}(S)\| \le 2$. Moreover, $C(S) = D(S) = \emptyset$, since C and D depend only on oracle words of length $\equiv 1 \pmod 4$. From Proposition 6.10.4, we obtain that X' is $(\mu, k{+}1)$-valid. Note that $\|N_a \cup N_b\| \le 2^{(k+1)/2}$, since $\|N_a \cup N_b\| \le \ell(N_a) + \ell(N_b) + 2 \le 2(2\alpha^j + 1) \le \gamma(2\alpha^j + 1)$. By assumption, $Y_a \cap N_b = Y_b \cap N_a = \emptyset$. Therefore, from Proposition 6.24 it follows that $A(Y_a \cup Y_b) \cap B(Y_a \cup Y_b) = \emptyset$. This shows Claim 6.26.

CLAIM 6.27. *Every* β*-dimensional vector of reservations has a conflict.*

Proof. Assume that there exists a vector of reservations

$$v = ((Y_{r_1}, N_{r_1}), (Y_{r_2}, N_{r_2}), \ldots, (Y_{r_\beta}, N_{r_\beta}))$$

such that v has no conflict. Let $\mu' \stackrel{df}{=} \{(n', i', j') \in \mu \mid n' < n\}$. Note that X is (μ', k)-valid and also $(\mu' \cup \{n, 0, j\}, k)$-valid (Proposition 6.10.2). Let $Y \stackrel{df}{=} \bigcup_{1 \le a \le \beta} Y_{r_a}$, $N \stackrel{df}{=} \bigcup_{1 \le a \le \beta} N_{r_a}$, and $X' \stackrel{df}{=} X \cup Y^{=k+1}$. We show that X' is $(\mu', k+1)$-valid. Since C and D depend only on oracle words of length $\equiv 1 \pmod 4$, we have $C(Y^{=k+1}) = D(Y^{=k+1}) = \emptyset$. Moreover, since n is not in the domain of μ' and since all words in $Y^{=k+1}$ have the prefix $0^n 1$, for all $(n', 0, j') \in \mu'$ it holds that $E_{n'}(Y^{=k+1}) = \emptyset$. Therefore, from Proposition 6.10.4 it follows that X' is $(\mu', k+1)$-valid.

Let us show that for $1 \le a \le \beta$, (Y_{r_a}, N_{r_a}) is a $(\mu', k+1)$-reservation for X'. By definition, (Y_{r_a}, N_{r_a}) is a $(\mu, k+1)$-reservation for some $(\mu, k+1)$-valid $Z \supseteq_k X$. Since every μ'-codeword is a μ-codeword, it suffices to verify ${Y_{r_a}}^{=k+1} \subseteq X'$ and ${N_{r_a}}^{=k+1} \subseteq \overline{X'}$. The first inclusion holds by the definition of X'. If the latter inclusion does not hold, then ${N_{r_a}}^{=k+1} \cap Y^{=k+1} \neq \emptyset$. Since $N_{r_a} \cap Y_{r_a} = \emptyset$, it follows that $N_{r_a} \cap Y_{r_b} \neq \emptyset$ for some $b \neq a$. This implies that v has a conflict, which is not possible by our assumption. This shows that for all a, if $1 \le a \le \beta$, then (Y_{r_a}, N_{r_a}) is a $(\mu', k+1)$-reservation for X'.

We show that (Y, N) is a $(\mu', k{+}1)$-reservation for X'. All (Y_{r_a}, N_{r_a}) are $(\mu', k{+}1)$-reservations that do not conflict with each other. From this we immediately obtain that $Y \cap N = \emptyset$, $Y^{\le k+1} \subseteq X'$, $N^{\le k+1} \subseteq \overline{X'}$, and all words in $Y^{>k+1}$ are of length $\equiv 0 \pmod 4$. If $A(Y) \cap B(Y) \neq \emptyset$, then there exist a, b such that $A(Y_{r_a} \cup Y_{r_b}) \cap B(Y_{r_a} \cup Y_{r_b}) \neq \emptyset$. This contradicts Claim 6.26. Therefore, $A(Y) \cap B(Y) = \emptyset$. Finally, if $w \in Y^{>k+1}$ is a μ'-codeword for (i', t', x'), then there exists some a such that $w \in {Y_{r_a}}^{>k+1}$. Since (Y_{r_a}, N_{r_a}) is a $(\mu', k+1)$-reservation, $NM_{i'}(x')$ has a positive path P such that $|P| \le t'$, $P^{\text{yes}} \subseteq Y_{r_a} \subseteq Y$, and $P^{\text{no}} \subseteq N_{r_a} \subseteq N$. This shows that (Y, N) is a $(\mu', k+1)$-reservation for X'.

By definition, for all r and all $(Y_r, N_r) \in L_r$ it holds that $\ell(Y_r \cup N_r) \le 2 \cdot \alpha^j$. Therefore, $\|N_r\| \le 2 \cdot \alpha^j + 1$ and it follows that $\|N\| \le \beta \cdot (2 \cdot \alpha^j + 1) \le 2^{(k+1)/2}$. By Lemmas 6.14 and 6.17 there exists some (μ', m)-valid $Z \supseteq_{k+1} X'$ such that $Y \cup N \subseteq \Sigma^{\le m}$, $Y \subseteq Z$, $N \subseteq \overline{Z}$, and $m \ge \alpha$. From the definition of the sets L_r it follows that for all a, if $1 \le a \le \beta$, then $NM_j^Z(z_{r_a})$ accepts. The length of all z_{r_a} is bounded by α. So there exists a length l such that $0 \le l \le \alpha$ and at least $\beta/(\alpha + 1) > (\alpha^j + j) \ge l^j + j$

of the words z_{r_a} are of length l. Hence $\|L(NM_j^Z) \cap \Sigma^l\| > l^j + j$, and therefore $L(NM_j^Z) \cap \Sigma^{\leq m} \notin \mathrm{SPARSE}_j$.

We know that X is $(\mu' \cup \{n, 0, j\}, k)$-valid. Moreover, $m \geq k \geq n$ and Z is (μ', m)-valid such that $Z^{\leq k} = X^{\leq k}$, and therefore $Z^{\leq n} = X^{\leq n}$. From Definition 6.9.2(b) it follows that $L(NM_j^Z) \cap \Sigma^{\leq m} \in \mathrm{SPARSE}_j$. This contradicts our observation in the last paragraph and finishes the proof of Claim 6.27. □

CLAIM 6.28. *There exist some r and an $N \subseteq \Sigma^{>k}$ such that $\|N\| \leq (2 \cdot \alpha^j + 2) \cdot \gamma$ and, for every (μ, m)-valid $Z \supseteq_k X$, if $m > k$, $N \subseteq \overline{Z} \cap \Sigma^{\leq m}$, $Z^{=k+1} \subseteq 0^n 1 \Sigma^*$, $\|Z \cap \Sigma^{k+1}\| \leq 1$, and $Z^{>k+1}$ contains only μ-codewords, then $NM_j^Z(z_r)$ rejects.*

Proof. We use the following algorithm to create the set N. Note that this algorithm modifies the sets L_r. This will decrease the number of possible vectors of reservations from $L_1, \ldots, L_\gamma$.

```
1  N(0) := ∅, R(0) := ∅, i := 0
2  while (all L_r ≠ ∅)
3     i := i + 1
4     choose the largest d such that there exists a
         d-dimensional vector v = ((Y_{r_1}, N_{r_1}), ..., (Y_{r_d}, N_{r_d})) of
         reservations from L_1, ..., L_γ such that
         v has no conflict
5     R(i) := R(i − 1) ∪ {r_1, r_2, ..., r_d}
6     N(i) := N(i − 1) ∪ Y^{>k}_{r_1} ∪ N^{>k}_{r_1} ∪ ··· ∪ Y^{>k}_{r_d} ∪ N^{>k}_{r_d}
7     for every r and every (Y_r, N_r) ∈ L_r:
         remove (Y_r, N_r) if Y_r ∩ N(i) ≠ ∅
8  end while
9  N := N(i)
```

Let $i \geq 1$ and consider the algorithm after the ith iteration of the *while* loop. We claim that for every $r \notin R(i)$ and every (Y_r, N_r) that remains in L_r, it holds that $N_r \cap (N(i) - N(i-1)) \neq \emptyset$. Otherwise, there exist r and (Y_r, N_r) such that $r \notin R(i)$, $(Y_r, N_r) \in L_r$, $N_r \cap (N(i) - N(i-1)) = \emptyset$, and (Y_r, N_r) has not been removed in step 7. Therefore, $Y_r \cap N(i) = \emptyset$, which implies $Y_r \cap (N(i) - N(i-1)) = \emptyset$. Together with our assumption, this gives us $(Y_r \cup N_r) \cap (N(i) - N(i-1)) = \emptyset$. By step 6 this means that (Y_r, N_r) does not conflict with any reservation in v. Therefore, with $((Y_r, N_r), (Y_{r_1}, N_{r_1}), \ldots, (Y_{r_d}, N_{r_d}))$ we found a $(d+1)$-dimensional vector of reservations that has no conflict. This contradicts the choice of v in step 4. Therefore, for every $r \notin R(i)$ and every (Y_r, N_r) that remains in L_r, it holds that $N_r \cap (N(i) - N(i-1)) \neq \emptyset$. It follows that after l iterations of the *while* loop, for every $r \notin R(l)$ and every (Y_r, N_r) that remains in L_r, it holds that $\|N_r\| \geq l$.

By Claim 6.27 and the choice of d in step 4 we have $d < \beta$. Therefore, after $(2 \cdot \alpha^j + 2)$ iterations, $\|R(i)\| < (2 \cdot \alpha^j + 2) \cdot \beta = \gamma$. So during the first $(2 \cdot \alpha^j + 2)$ iterations i there always exists an $r \notin R(i)$. Moreover, for every r and every $(Y_r, N_r) \in L_r$, it holds that $\ell(Y_r \cup N_r) \leq 2 \cdot \alpha^j$, and therefore $\|N_r\| \leq 2 \cdot \alpha^j + 1$. From the conclusion of the previous paragraph it follows that the *while* loop iterates at most $2 \cdot \alpha^j + 2$ times. This shows that the algorithm terminates. Since $d < \beta$, for all $i \geq 1$ it holds that $\|N(i) - N(i-1)\| < \beta \cdot (2 \cdot \alpha^j + 1) \leq \gamma$. Therefore, $\|N\| \leq (2 \cdot \alpha^j + 2) \cdot \gamma$ and $N \subseteq \Sigma^{>k}$ when the algorithm terminates.

So we have a set N of the required size and an r such that $L_r = \emptyset$. We show that N and r satisfy Claim 6.28. Assume that for some $m \geq k+1$ there exists a (μ, m)-valid $Z \supseteq_k X$ such that $N \subseteq \overline{Z} \cap \Sigma^{\leq m}$, $Z^{=k+1} \subseteq 0^n 1 \Sigma^*$, $\|Z \cap \Sigma^{k+1}\| \leq 1$, $Z^{>k+1}$ contains only μ-codewords, and $NM_j^Z(z_r)$ accepts. Let P_r be an accepting path of $NM_j^Z(z_r)$.

Let $Z' \stackrel{df}{=} Z^{\leq k+1}$. From Proposition 6.10.6 it follows that Z' is $(\mu, k+1)$-valid (since $k+1 > k \geq \mu_{\max}$). $Z^{>k+1}$ contains only words of length $\equiv 0 \pmod 4$, since it contains only μ-codewords. So we can apply Lemma 6.18 (for $X = Z'$, $Y = P_r^{\text{yes}}$, and $N = P_r^{\text{no}}$). We obtain a $(\mu, k+1)$-reservation (Y', N') for Z' such that $P_r^{\text{yes}} \subseteq Y'$, $P_r^{\text{no}} \subseteq N'$, $\ell(Y' \cup N') \leq 2 \cdot \ell(P_r^{\text{yes}} \cup P_r^{\text{no}}) \leq 2 \cdot \alpha^j$, $Y' \subseteq Z$, and $N' \subseteq \overline{Z}$. Together with $N \subseteq \overline{Z}$, this implies

$$Y' \cap N = \emptyset. \tag{17}$$

We show that at the beginning of the algorithm, (Y', N') must have been in L_r . Since $Z^{>k+1}$ contains only μ-codewords and since $Y' \subseteq Z$, then $Y'^{>k+1}$ also contains only μ-codewords. Moreover, $Z'^{=k+1} = Z^{=k+1} \subseteq 0^n 1 \Sigma^*$ and $\|Z' \cap \Sigma^{k+1}\| = \|Z \cap \Sigma^{k+1}\| \leq 1$. By our assumption, P_r is a positive path of $NM_j(z_r)$, and it holds that $P_r^{\text{yes}} \subseteq Y'$ and $P_r^{\text{no}} \subseteq N'$. It follows that (Y', N') must have been in L_r.

Since $L_r = \emptyset$ when the algorithm terminates, (Y', N') has been removed during some iteration i. This implies that during that iteration, $Y' \cap N(i) \neq \emptyset$ (by line 7). Moreover, by line 9, $N(i) \subseteq N$. This implies $Y' \cap N \neq \emptyset$, which contradicts (17). This proves Claim 6.28. □

Now we finish Case 2. Let r and N be as in Claim 6.28. Choose a word y_r of length $(k+1)/2$ such that $0^n 1 x_r y_r \notin N$. Let $S \stackrel{df}{=} \{0^n 1 x_r y_r\}$. It follows that $C(S) = D(S) = \emptyset$. Moreover, for all $n' \geq 1$, $\|E_{n'}(S)\| \leq 1$. From Proposition 6.10.4 it follows that $X' \stackrel{df}{=} X \cup S$ is $(\mu, k+1)$-valid. By Proposition 6.13.3, $(\emptyset, N)$ is a $(\mu, k+1)$-reservation for X'. Note that $\|N\| \leq (2 \cdot \alpha^j + 2) \cdot \gamma < 2^{(k+1)/2}$. Therefore, by Lemmas 6.14 and 6.17 there exists an $l \geq \alpha^j$ and a (μ, l)-valid $Y \supseteq_{k+1} X'$ such that $N \subseteq \overline{Y} \cap \Sigma^{\leq l}$ and $Y^{>k+1}$ contains only μ-codewords. From Claim 6.28 it follows that $NM_j^Y(z_r)$ rejects. The computation times of $f_i^Y(0^n 1 x_r)$ and $NM_j^Y(z_r)$ are bounded by $\alpha^j \leq l$. Therefore, for all $Z \supseteq_l Y$ it holds that $f_i^Z(0^n 1 x_r) = z_r$, $0^n 1 x_r \in E_n(Z)$, and $NM_j^Z(z_r)$ rejects. This shows that $E_n(Z)$ does not $\leq_m^{p,Z}$-reduce to $L(NM_j^Z)$ via f_i^Z. This finishes the proof of Proposition 6.25. □

Recall that we want to construct the oracle in a way such that $(A(O_2), B(O_2))$ is not $\leq_T^{pp,O_2}$-hard for NP^{O_2}. We have seen that it suffices to construct $F(O_2)$ such that it does not $\leq_T^{pp}$-reduce to $(A(O_2), B(O_2))$. We prevent $F(O_2) \leq_T^{pp} (A(O_2), B(O_2))$ via M_i as follows: We consider the computation $M_i(0^n)$, where the machine can ask queries to the pair $(A(X), B(X))$. In Lemma 6.29 we show that each query to this pair can be forced to be either in the complement of $A(X)$ or in the complement of $B(X)$. For this forcing it is enough to reserve polynomially many words for the complement of X. If we forced the query to be in the complement of $A(X)$, then the oracle can safely answer that the query belongs to $B(X)$. Otherwise it can safely answer that the query belongs to $A(X)$. After forcing all queries of the computation, we add an unreserved word to $F(X)$ if and only if the computation rejects. This will show that $F(X)$ does not $\leq_T^{pp}$-reduce to $(A(X), B(X))$ via M_i (Proposition 6.32).

LEMMA 6.29. *Let $k \equiv 2 \pmod 4$ and let X be (μ, k)-valid. For every $q \in \Sigma^*$, $|q| \leq 2^{k/2-4} - 2$, there exists an $N \subseteq \Sigma^{>k}$ such that $\|N\| \leq (8 \cdot |q| + 10)^2$ and one of the following properties holds:*

1. *For all (μ, m)-valid $Z \supseteq_k X$, if $m > k$, $N \subseteq \overline{Z}$, and $Z^{>k+1}$ contains only μ-codewords, then $q \notin A(Z)$.*
2. *For all (μ, m)-valid $Z \supseteq_k X$, if $m > k$, $N \subseteq \overline{Z}$, and $Z^{>k+1}$ contains only μ-codewords, then $q \notin B(Z)$.*

Proof. We can assume that $q = 0^n 10^t 1x$ for suitable n, t, x. Otherwise, q cannot belong to $A(Z) \cup B(Z)$ for all oracles Z, and we are done. Define the following sets:

$L_A \stackrel{df}{=} \{(Y_A, N_A) \mid (Y_A, N_A)$ is a $(\mu, k+1)$-reservation for some $(\mu, k+1)$-valid $Z \supseteq_k X$, ${Y_A}^{>k+1}$ contains only μ-codewords, $\ell(Y_A \cup N_A) \leq 8(|q|+1)$, and $(\exists y \in \Sigma^{3|q|+3})[0qy \in Y_A]\}$,

$L_B \stackrel{df}{=} \{(Y_B, N_B) \mid (Y_B, N_B)$ is a $(\mu, k+1)$-reservation for some $(\mu, k+1)$-valid $Z \supseteq_k X$, ${Y_B}^{>k+1}$ contains only μ-codewords, $\ell(Y_B \cup N_B) \leq 8(|q|+1)$, and $(\exists y \in \Sigma^{3|q|+3})[1qy \in Y_B]\}$.

We say that $(Y_A, N_A) \in L_A$ and $(Y_B, N_B) \in L_B$ *conflict* if and only if $Y_A \cap N_B \neq \emptyset$ or $N_A \cap Y_B \neq \emptyset$. Note that if (Y_A, N_A) and (Y_B, N_B) conflict, then even $Y_A \cap N_B \cap \Sigma^{>k} \neq \emptyset$ or $N_A \cap Y_B \cap \Sigma^{>k} \neq \emptyset$.

CLAIM 6.30. *Every $(Y_A, N_A) \in L_A$ conflicts with every $(Y_B, N_B) \in L_B$.*

Proof. Assume that there exist $(Y_A, N_A) \in L_A$ and $(Y_B, N_B) \in L_B$ that do not conflict. Let $Y' \stackrel{df}{=} Y_A \cup Y_B$, $N' \stackrel{df}{=} N_A \cup N_B$ and $S \stackrel{df}{=} {Y_A}^{=k+1} \cup {Y_B}^{=k+1}$.

We show that (Y', N') is a $(\mu, k+1)$-reservation for $X' \stackrel{df}{=} X \cup S$. Since $k \equiv 2 \pmod 4$ and $S \subseteq \Sigma^{k+1}$, it holds that $C(S) = D(S) = \emptyset$ and, for all $n' \geq 1$, $E_{n'}(S) = \emptyset$. From Proposition 6.10.4, it follows that X' is $(\mu, k+1)$-valid. Note that $\|N_A \cup N_B\| \leq 2^{(k+1)/2}$, since $\|N_A \cup N_B\| \leq \ell(N_A) + \ell(N_B) + 2 \leq 16|q| + 18 \leq 2^{k/2}$. By assumption, $Y_A \cap N_B = Y_B \cap N_A = \emptyset$. From Proposition 6.24 it follows that $A(Y_A \cup Y_B) \cap B(Y_A \cup Y_B) = \emptyset$. Therefore, it remains to verify $Y' \cap N' = \emptyset$, ${Y'}^{=k+1} \subseteq X'$, and ${N'}^{=k+1} \subseteq \overline{X'}$. The first condition holds, since (Y_A, N_A) and (Y_B, N_B) do not conflict. The second one holds by the definition of X'. Finally, ${N'}^{=k+1} \subseteq \overline{X'}$ holds, since otherwise ${N'}^{=k+1} \cap S \neq \emptyset$, and therefore $Y' \cap N' \neq \emptyset$. This shows that (Y', N') is a $(\mu, k+1)$-reservation for X'.

From the definition of L_A and L_B it follows that $\|N'\| \leq 16 \cdot |q| + 18 \leq 2^{k/2}$. By Lemma 6.14, there exist an $m \geq k+1$ and a (μ, m)-valid $Z \supseteq_{k+1} X'$ such that $Y' \subseteq Z$. Since $(Y_A, N_A) \in L_A$ and $(Y_B, N_B) \in L_B$, there exist $y_0, y_1 \in \Sigma^{3|q|+3}$ such that $0qy_0 \in Y_A \subseteq Y' \subseteq Z$ and $1qy_1 \in Y_B \subseteq Y' \subseteq Z$. Therefore, $q \in A(Z) \cap B(Z)$, which contradicts the fact that Z is (μ, m)-valid. This proves Claim 6.30. □

We use the following algorithm to create the set N as claimed in the statement of this lemma.

```
1    N := ∅
2    while (L_A ≠ ∅ and L_B ≠ ∅)
3        choose some (Y'_A, N'_A) ∈ L_A
4        N := N ∪ Y'_A^{>k} ∪ N'_A^{>k}
5        for every (Y_A, N_A) ∈ L_A
6            remove (Y_A, N_A) if Y_A ∩ (Y'_A^{>k} ∪ N'_A^{>k}) ≠ ∅
7        for every (Y_B, N_B) ∈ L_B
8            remove (Y_B, N_B) if Y_B ∩ (Y'_A^{>k} ∪ N'_A^{>k}) ≠ ∅
9    end while
```

We claim that after l iterations of the *while* loop, for every $(Y_B, N_B) \in L_B$, $\|N_B\| \geq l$. If this claim is true, the while loop iterates at most $8 \cdot |q| + 10$ times, since for any $(Y_B, N_B) \in L_B$, $\ell(N_B) \leq 8 \cdot |q| + 8$, and therefore $\|N_B\| \leq 8 \cdot |q| + 9$. On the other hand, during each iteration, N is increased by at most $8 \cdot |q| + 9$ strings. Therefore, $\|N\| \leq (8 \cdot |q| + 10)^2$ and $N \subseteq \Sigma^{>k}$ when this algorithm terminates.

CLAIM 6.31. *After l iterations of the* while *loop, for every (Y_B, N_B) that remains in L_B, $\|N_B\| \geq l$.*

Proof. For every l, let us denote the pair that is chosen during the lth iteration in step 3 by (Y_A^l, N_A^l). By Claim 6.30, every (Y_B, N_B) that belongs to L_B at the beginning of this iteration conflicts with (Y_A^l, N_A^l), i.e., $N_A^l \cap Y_B \cap \Sigma^{>k} \neq \emptyset$ or $Y_A^l \cap N_B \cap \Sigma^{>k} \neq \emptyset$. If $N_A^l \cap Y_B \cap \Sigma^{>k} \neq \emptyset$, then (Y_B, N_B) will be removed from L_B in step 8. Otherwise, $Y_A^l \cap N_B \cap \Sigma^{>k}$ is not empty, and therefore there exists a lexicographically smallest word w_l in this set. In this case, (Y_B, N_B) will not be removed from L_B; we say that (Y_B, N_B) *survives* the lth iteration *due to the word* w_l. Note that (Y_B, N_B) can survive only due to a word that belongs to N_B. We will use this fact to prove that $\|N_B\| \geq l$ after l iterations.

We show now that any pair (Y_B, N_B) that is left in L_B after l iterations survives each of these iterations due to a different word. Since these words all belong to N_B, this will complete the proof of the claim. Assume that there exist iterations l and l' with $l < l'$ such that $w_l = w_{l'}$. Then $w_l \in Y_A^l \cap N_B \cap \Sigma^{>k}$ and $w_{l'} \in Y_A^{l'} \cap N_B \cap \Sigma^{>k}$. Therefore, $Y_A^l \cap Y_A^{l'} \cap \Sigma^{>k} \neq \emptyset$. So the pair $(Y_A^{l'}, N_A^{l'})$ should have been removed in iteration l (step 6) and cannot be chosen at the beginning of iteration l', as claimed. Hence, $w_l \neq w_{l'}$. This proves Claim 6.31. □

Therefore, we now have a set N of the required size such that either L_A or L_B will be empty. Assume that L_A is empty; we will show that Lemma 6.29.1 holds. Analogously we show that if L_B is empty, then Lemma 6.29.2 holds. Assume that for some $m \geq k+1$ there exists a (μ, m)-valid $Z \supseteq_k X$ such that $q \in A(Z)$, $N \subseteq \overline{Z}$, and $Z^{>k+1}$ contains only μ-codewords. Hence, there exists some $y \in \Sigma^{3|q|+3}$ such that $0qy \in Z$.[6]

Let $Z' \stackrel{df}{=} Z^{\leq k+1}$. From Proposition 6.10.6 it follows that Z' is $(\mu, k+1)$-valid. Since $Z^{>k+1}$ contains only μ-codewords, we can apply Lemma 6.18 for $(\{0qy\}, \emptyset)$. We obtain a $(\mu, k+1)$-reservation (Y', N') for Z' such that $0qy \in Y'$, $\ell(Y' \cup N') \leq 2 \cdot |0qy| = 8 \cdot (|q|+1)$, and $Y' \subseteq Z \subseteq \overline{N'}$. Together with $N \subseteq \overline{Z}$, this implies

$$Y' \cap N = \emptyset. \tag{18}$$

Moreover, since $Y' \subseteq Z$, it holds that ${Y'}^{>k+1}$ contains only μ-codewords. It follows that (Y', N') must have been in L_A and has been removed during some iteration. This implies that during that iteration, $Y' \cap ({Y'_A}^{>k} \cup {N'_A}^{>k}) \neq \emptyset$ (by line 6). Moreover, by line 4, ${Y'_A}^{>k} \cup {N'_A}^{>k}$ is a subset of N when the algorithm stops. This implies $Y' \cap N \neq \emptyset$, which contradicts equation (18). This proves Lemma 6.29. □

PROPOSITION 6.32 (property P4). *Let $i \geq 1$ and let X be (μ, k)-valid. There exists an $l > k$ and a (μ, l)-valid $Y \supseteq_k X$ such that for all $Z \supseteq_l Y$, if $A(Z) \cap B(Z) = \emptyset$, then there exists a separator S of $(A(Z), B(Z))$ such that $F(Z) \neq L(M_i^S)$.*

Proof. By Lemma 6.17, we can assume that $k \equiv 2 \pmod 4$ and $64(k+10)^{3i} < 2^{k/2}$.

We describe the construction of S_A and S_B, which are sets of queries we reserve for $\overline{B(Y)}$ and $\overline{A(Y)}$, respectively. Let $S_A := A(X)$ and $S_B := B(X)$. We simulate the computation $M_i^{S_A}(0^{k+1})$ until we reach a query q_1 that belongs to neither S_A nor S_B. Note that $|q_1| \leq (k+1)^i \leq 2^{k/2-4} - 2$. From Lemma 6.29 we obtain some $N_1 \subseteq \Sigma^{>k}$ such that $\|N_1\| \leq (8 \cdot |q_1| + 10)^2$ and either property 6.29.1 or property 6.29.2 holds. If property 6.29.1 holds, then add q_1 to S_B; otherwise add q_1 to S_A. Now return the

[6] Actually, it even holds that $0qy \in Z - X$, but we do not need this explicitly in our argumentation. In order to see this, we assume that $0qy$ is in X. Then q is in $A(X)$ and $(\{0qy\}, \emptyset)$ is a (μ, k)-reservation for X. Therefore, $(\{0qy\}, \emptyset)$ is a $(\mu, k+1)$-reservation for every $(\mu, k+1)$-valid $Z \supseteq_k X$. Hence, $(\{0qy\}, \emptyset)$ is in L_A at the beginning of the algorithm. So it has been removed during the algorithm. But this is not possible since elements in L_A can only be removed in step 6, and there we remove only (Y_A, N_A) with $Y_A \cap \Sigma^{>k} \neq \emptyset$. This shows $0qy \in Z - X$.

answer of "$q_1 \in S_A$?" to the computation. We continue the simulation until we reach a query q_2 that belongs to neither S_A nor S_B. Again we apply Lemma 6.29, obtain the set N_2, and add q_2 either to S_A or to S_B. We continue the simulation until the computation stops. Let n be the number of queries that were added to S_A or S_B. Observe that $S_A \cap S_B = \emptyset$ at the end of our simulation.

Let $N \stackrel{df}{=} N_1 \cup \cdots \cup N_n \cup \{0^{4(k+1)^i+4}\}$. Then $\|N\| \leq (k+1)^i \cdot (8 \cdot (k+1)^i + 10)^2 + 1 \leq 2^{k/2}$. Hence there exists some $w \in \Sigma^{k+1} - N$. If the simulation accepts, then let $S' = \emptyset$; otherwise let $S' \stackrel{df}{=} \{w\}$. Since $S \subseteq \Sigma^{k+1}$ and $k+1 \equiv 3 \pmod 4$, we have $C(S') = D(S') = \emptyset$ and for all $n \geq 1$, $E_n(S') = \emptyset$. From Proposition 6.10.4, it follows that $Y' \stackrel{df}{=} X \cup S'$ is $(\mu, k+1)$-valid. Since $N \subseteq \Sigma^{>k}$ and $N \cap S' = \emptyset$, we have $N \subseteq \overline{Y'}$. Therefore, by Proposition 6.13.3, $(\emptyset, N)$ is a $(\mu, k+1)$-reservation for Y'. By Lemma 6.14, there exist an $l \geq 4(k+1)^i + 4$ and a (μ, l)-valid $Y \supseteq_{k+1} Y'$ such that $N \subseteq \overline{Y}$ and $Y^{>k+1}$ contains only μ-codewords. In particular, it holds that $l > k$ and $Y \supseteq_k X$.

CLAIM 6.33. *For every $Z \supseteq_l Y$ it holds that $S_A \subseteq \overline{B(Z)}$ and $S_B \subseteq \overline{A(Z)}$.*

Assume that $S_A \cap B(Z) \neq \emptyset$ for some $Z \supseteq_l Y$, and choose a $v \in S_A \cap B(Z)$. Since S_A contains only words of length $\leq (k+1)^i$, we obtain $v \in S_A \cap B(Z^{\leq 4(k+1)^i+4}) \subseteq S_A \cap B(Y)$. So v cannot belong to $A(Y)$ since $A(Y) \cap B(Y) = \emptyset$. In particular this means $v \in S_A - A(X)$; i.e., $v = q_j$ for a suitable j with $1 \leq j \leq n$. By our construction q_j was only added to S_A when property 2 of Lemma 6.29 holds. Remember that Y is (μ, l)-valid with $l > k$, $Y \supseteq_k X$, $N_j \subseteq N \subseteq \overline{Y}$, and $Y^{>k+1}$ contains only μ-codewords. Therefore, from property 6.29.2 it follows that $v = q_j \notin B(Y)$, which contradicts $v \in S_A \cap B(Y)$. This shows $S_A \subseteq \overline{B(Z)}$. By the symmetric argument we obtain $S_B \subseteq \overline{A(Z)}$. This proves Claim 6.33.

Consider any $Z \supseteq_l Y$ with $A(Z) \cap B(Z) = \emptyset$. Let $S \stackrel{df}{=} A(Z) \cup S_A$. Assume that S is not a separator of $(A(Z), B(Z))$. Since $A(Z) \subseteq S$, we must have $S \cap B(Z) \neq \emptyset$. Since $A(Z) \cap B(Z) = \emptyset$, this implies $S_A \cap B(Z) \neq \emptyset$. This contradicts Claim 6.33. So S is a separator of $(A(Z), B(Z))$. It remains to show $F(Z) \neq L(M_i^S)$.

By our construction, $0^{k+1} \in F(Y')$ if and only if $M_i^{S_A}(0^{k+1})$ rejects. Since $Z \supseteq_{k+1} Y'$, it holds that $0^{k+1} \in F(Z)$ if and only if $M_i^{S_A}(0^{k+1})$ rejects. Assume that there exists a query q that is answered differently in the computations $M_i^{S_A}(0^{k+1})$ and $M_i^S(0^{k+1})$ (take the first such query). Since $S_A \subseteq S$, we obtain $q \in S - S_A$, i.e., $q \in A(Z)$. If q is in $B(X)$, then q is in $B(Z) \subseteq \overline{S}$, which is not possible. So q is neither in S_A nor in $B(X)$, but q is asked in the computation $M_i^{S_A}(0^{k+1})$. It follows that $q = q_j$ for some j with $1 \leq j \leq n$, and during the construction we added q_j to S_B. So we have $q \in S_B \cap A(Z)$, which contradicts Claim 6.33. Therefore, $M_i^{S_A}(0^{k+1})$ accepts if and only if $M_i^S(0^{k+1})$ accepts. This shows $0^{k+1} \in F(Z)$ if and only if $M_i^S(0^{k+1})$ rejects, i.e., $F(Z) \neq L(M_i^S)$. □

This finishes the proof of Theorem 6.7. □

COROLLARY 6.34. *The oracle O_2 of Theorem 6.7 has the following additional properties:*

(i) $\mathrm{UP}^{O_2} \neq \mathrm{NP}^{O_2} \neq \mathrm{coNP}^{O_2}$ *and* $\mathrm{NPMV}^{O_2} \not\subseteq_c \mathrm{NPSV}^{O_2}$.

(ii) *Relative to O_2, no optimal propositional proof systems exist.*

(iii) *There exists a $\leq_{sm}^{pp}$-complete disjoint NP^{O_2}-pair (A, B) that is P^{O_2}-inseparable but symmetric.*

Proof. It is known that Conjecture 2.4 implies item (i) [ESY84, GS88, Sel94]. Relative to O_2, NP∩SPARSE does not have $\leq_m^{p,O_2}$-complete sets. Messner and Torán [MT98] proved that this implies that there are no optimal propositional proof systems. This shows (ii).

Since (A, B) is $\leq^{pp}_{sm}$-complete, it is symmetric. If (A, B) is P^{O_2}-separable, then every disjoint NP^{O_2}-pair is P^{O_2}-separable, and therefore symmetric. This contradicts item (ii) of Theorem 6.7. So (A, B) is P^{O_2}-inseparable. □

7. Relationship to optimal propositional proof systems. It is known that existence of optimal propositional proof systems implies existence of $\leq^{pp}_{m}$-complete disjoint NP-pairs. Messner and Torán [MT98] state that this result was communicated to them by Impagliazzo and Pitassi. Ben-David and Gringauze [BDG98] cite Razborov [Raz94] for this result. Köbler, Messner, and Torán [KMT03] cite Razborov, and they prove the stronger result that existence of optimal propositional proof systems implies existence of $\leq^{pp}_{sm}$-complete disjoint NP-pairs.[7] For the sake of completeness, we provide here a straightforward proof of the weaker result.

THEOREM 7.1. *If optimal propositional proof systems exist, then there is a* $\leq^{pp}_{m}$*-complete disjoint NP-pair.*

Proof. Let f be an optimal propositional proof system. We define the canonical pair [Raz94, Pud03] for this proof system, $(\mathrm{SAT}^*, \mathrm{REF}_f)$, where

$$\mathrm{SAT}^* = \{(x, 0^n) \mid x \in \mathrm{SAT}\}$$

and

$$\mathrm{REF}_f = \{(x, 0^n) \mid \neg x \in \mathrm{TAUT} \text{ and } \exists y[|y| \leq n \text{ and } f(y) = \neg x]\}.$$

Note that since f is polynomial-time computable, both SAT^* and REF_f are in NP. Also, for any n, if $(x, 0^n) \in \mathrm{SAT}^*$, then $x \in \mathrm{SAT}$, and if $(x, 0^n) \in \mathrm{REF}_f$, then $x \notin \mathrm{SAT}$. Therefore, these sets are disjoint, and so $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is a disjoint NP-pair. We will prove that this pair is $\leq^{pp}_{m}$-complete.

Consider any other disjoint NP-pair (A, B). We will define a proof system $f_{A,B}$ using this pair. Assume that $A \leq^{p}_{m} \mathrm{SAT}$ via $g \in \mathrm{PF}$ and there is a polynomial $p(\cdot)$ and a polynomial-time predicate $R(\cdot, \cdot)$ such that $z \in B \Leftrightarrow \exists w, |w| \leq p(|z|), R(z, w)$.

$$f_{A,B}(y) = \begin{cases} \neg g(z) & \text{if } y = (z, w), \text{ where } |w| \leq p(|z|) \text{ and } R(z, w), \\ z & \text{if } y = (z, w), \text{ where } |w| > 2^{|z|} \text{ and } z \in \mathrm{TAUT}, \\ z \vee \neg z & \text{otherwise.} \end{cases} \tag{19}$$

We claim that $f_{A,B}$ is a proof system. First, note that for every $z \in \mathrm{TAUT}$, $f_{A,B}(z, w)$, for some $w, |w| > 2^{|z|}$, will output z in time polynomial in $|(z, w)|$. Also, since $A \cap B = \emptyset$ and g reduces A to SAT, $g(B) \subset \overline{\mathrm{SAT}}$. Therefore, for every $z \in B$ (i.e., for every z such that $R(z, w)$ for some $w, |w| \leq p(|z|)$), $g(z) \notin \mathrm{SAT}$. Therefore, $f_{A,B}$ outputs all possible tautologies and does not output anything that is not in TAUT. Also, since g is polynomial-time computable, so is $f_{A,B}$. It is therefore clear that $f_{A,B}$ is a proof system; since f is an optimal proof system, there is a polynomial $q(\cdot)$ such that for every tautology ϕ, and for every w such that $f_{A,B}(w) = \phi$, there is a $w', |w'| \leq q(|w|)$ and $f(w') = \phi$.

Now we define $h \in \mathrm{PF}$ such that $(A, B) \leq^{pp}_{m} (\mathrm{SAT}^*, \mathrm{REF}_f)$ via h. On input x, h outputs $(g(x), 0^{r(|x|)})$, where $r(\cdot)$ is some polynomial that we will fix later. If $x \in A$, then $g(x) \in \mathrm{SAT}$, and therefore $h(x) \in \mathrm{SAT}^*$.

On the other hand, for all $x \in B$, $g(x) \notin \mathrm{SAT}$, i.e., $\neg g(x) \in \mathrm{TAUT}$. Since $x \in B$, there exists $y = (x, w)$, where $|w| \leq p(|x|)$ such that $f_{A,B}(y) = \neg g(x)$. So,

[7] However, a forthcoming paper [GSS04] proves that there exist $\leq^{pp}_{sm}$-complete disjoint NP-pairs if and only if there exist $\leq^{pp}_{m}$-complete disjoint NP-pairs.

there is some y', $|y'| \leq q(|y|)$, such that $f(y') = \neg g(x)$. We choose r to be large enough so that $r(|x|) > |y'|$, and since q and p are polynomial, r can be chosen to be a polynomial as well. This shows that $x \in B$ implies $h(x) \in \mathrm{REF}_f$. Therefore, $(A, B) \leq_m^{pp} (\mathrm{SAT}^*, \mathrm{REF}_f)$; i.e., $(\mathrm{SAT}^*, \mathrm{REF}_f)$ is $\leq_m^{pp}$-complete. □

8. Conclusions. We partially summarize the import of the oracle results we obtained in this paper. Various implications have been known and/or are observed here for the first time. For several of these, our oracles demonstrate that the converses do not hold robustly. The following are convenient lists of these instances:

- Existence of optimal proof systems implies existence of $\leq_{sm}^{pp}$-complete NP-pairs [Raz94, KMT03]. Relative to oracle O_2, the converse is false.

Relative to both oracles O_1 and O_2, the converses of the following implications are false:

1. Nonexistence of $\leq_T^{pp}$-complete NP-pairs implies Conjecture 2.4 (observed in section 3).
2. Nonsymmetric implies P-inseparable (observed in section 5).
3. Nonexistence of $\leq_T^{pp}$-complete NP-pairs implies NP $\neq$ coNP (observed in section 3).
4. Nonexistence of $\leq_m^{pp}$-complete NP-pairs implies NP $\neq$ coNP (observed in section 3).

Acknowledgments. The authors thank Avi Wigderson for informing them of the paper by Ben-David and Gringauze [BDG98]. Also we thank the anonymous referees for their careful reading and thoughtful comments.

REFERENCES

[BGS75] T. Baker, J. Gill, and R. Solovay, *Relativizations of the $\mathcal{P} =? \mathcal{NP}$ question*, SIAM J. Comput., 4 (1975), pp. 431–442.

[BDG98] S. Ben-David and A. Gringauze, *On the Existence of Propositional Proof Systems and Oracle-Relativized Propositional Logic*, Technical Report 5, Electronic Colloquium on Computational Complexity, 1998.

[BFFvM00] H. Buhrman, S. Fenner, L. Fortnow, and D. van Melkebeek, *Optimal proof systems and sparse sets*, in Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1770, Springer-Verlag, Berlin, 2000, pp. 407–418.

[CR79] S. Cook and R. Reckhow, *The relative efficiency of propositional proof systems*, J. Symbolic Logic, 44 (1979), pp. 36–50.

[ESY84] S. Even, A. Selman, and J. Yacobi, *The complexity of promise problems with applications to public-key cryptography*, Inform. and Control, 61 (1984), pp. 159–173.

[FHOS97] S. Fenner, S. Homer, M. Ogihara, and A. Selman, *Oracles that compute values*, SIAM J. Comput., 26 (1997), pp. 1043–1065.

[FPS01] L. Fortnow, A. Pavan, and A. Selman, *Distributionally hard languages*, Theory Comput. Syst., 34 (2001), pp. 245–261.

[GW03] C. Glasser and G. Wechsung, *Relativizing function classes*, J. UCS, 9 (2003), pp. 34–50.

[GSS04] C. Glasser, A. Selman, and S. Sengupta, *Reductions between disjoint NP-pairs*, in Proceedings of the 19th IEEE Conference on Computational Complexity, IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 42–53.

[GS88] J. Grollmann and A. L. Selman, *Complexity measures for public-key cryptosystems*, SIAM J. Comput., 17 (1988), pp. 309–335.

[Gur83] Y. Gurevich, *Algebras of feasible functions*, in Proceedings of the 24th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1983, pp. 210–214.

[HY84] J. Hartmanis and Y. Yesha, *Computation times of NP sets of different densities*, Theoret. Comput. Sci., 34 (1984), pp. 17–32.

[HIS85] J. Hartmanis, N. Immerman, and V. Sewelson, *Sparse sets in NP − P: EXPTIME versus NEXPTIME*, Inform. and Control, 65 (1985), pp. 158–181.

[HJV93] L. Hemaspaandra, S. Jain, and N. Vereshchagin, *Banishing robust Turing completeness*, Internat. J. Found. Comput. Sci., 4 (1993), pp. 245–265.

[HS92] S. Homer and A. Selman, *Oracles for structural properties: The isomorphism problem and public-key cryptography*, J. Comput. System Sci., 44 (1992), pp. 287–301.

[KM00] J. Köbler and J. Messner, *Is the standard proof system for sat p-optimal?*, in Proceedings of the 20th Conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS), Lecture Notes in Comput. Sci. 1974, Springer-Verlag, Berlin, 2000, pp. 361–372.

[KMT03] J. Köbler, J. Messner, and J. Torán, *Optimal proof systems imply complete sets for promise classes*, Inform. and Comput., 184 (2003), pp. 71–92.

[KP89] J. Krajíček and P. Pudlák, *Propositional proof systems, the consistency of first order theories and the complexity of computations*, J. Symbolic Logic, 54 (1989), pp. 1063–1079.

[MT98] J. Messner and J. Torán, *Optimal proof systems for propositional logic and complete sets*, in Proceedings of the 15th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1373, Springer-Verlag, Berlin, 1998, pp. 477–487.

[PS02] A. Pavan and A. L. Selman, *Separation of NP-completeness notions*, SIAM J. Comput., 31 (2002), pp. 906–918.

[Pud86] P. Pudlák, *On the length of proofs of finitistic consistency statements in first order theories*, in Logic Colloquium '84, J. B. Paris et al., eds., North–Holland, Amsterdam, 1986, pp. 165–196.

[Pud03] P. Pudlák, *On reducibility and symmetry of disjoint NP-pairs*, Theoret. Comput. Sci., 1-3 (2003), pp. 323–339.

[Raz94] A. Razborov, *On Provably Disjoint NP-Pairs*, Technical Report TR94-006, Electronic Colloquium on Computational Complexity, 1994.

[Sel79] A. Selman, *P-selective sets, tally languages, and the behavior of polynomial-time reducibilities on NP*, Math. Systems Theory, 13 (1979), pp. 55–65.

[Sel88] A. Selman, *Promise problems complete for complexity classes*, Inform. and Comput., 78 (1988), pp. 87–98.

[Sel94] A. Selman, *A taxonomy of complexity classes of functions*, J. Comput. System Sci., 48 (1994), pp. 357–381.