Selectivity: Reductions, Nondeterminism, and Function Classes[1]

Lane A. Hemaspaandra[2] Albrecht Hoene[3] Ashish V. Naik[4] Mitsunori Ogiwara[5]
Alan L. Selman[6] Thomas Thierauf[7] Jie Wang[8]

August 1993
Key words: Complexity theory, semi-decision algorithms, membership testing, selector functions, lowness, nonuniform complexity

[1]Some of these results appear in preliminary form in "Computing Solutions Uniquely Collapses the Polynomial Hierarchy" (Department of Computer Science, State University of New York technical report TR93-28, August 1993; L. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman) and "Selectivity" (a 1993 ICCI Conference contribution; L. Hemachandra, A. Hoene, M. Ogiwara, and A. Selman, T. Thierauf, and J. Wang).

[2]Department of Computer Science, University of Rochester, Rochester, NY 14627, USA. Supported in part by the NSF under grants CCR-8957604 and NSF-INT-9116781/JSPS-ENG-207. Work done in part while visiting the University of Electro-Communications–Tokyo.

[3]Fachbereich 20, Informatik, Technische Universität Berlin, D-10587 Berlin, Germany. Supported in part by a DFG Postdoctoral Fellowship and the NSF under grant CCR-8957604. Work done in part while visiting the University of Rochester and the University of Washington–Seattle.

[4]Department of Computer Science, SUNY–Buffalo, Buffalo, NY 14260, USA. Supported in part by the NSF under grant CCR-9002292.

[5]Department of Computer Science and Information Mathematics, University of Electro-communications, Tokyo 182, Japan. Supported in part by the NSF under grant CCR-9002292 and the JSPS under grant NSF-INT-9116781/JSPS-ENG-207. Work done in part while visiting SUNY–Buffalo.

[6]Department of Computer Science, SUNY–Buffalo, Buffalo, NY 14260, USA. Supported in part by the NSF under grant CCR-9002292.

[7]Abt. Theoretische Informatik, Universität Ulm, D-89069 Ulm, Germany. Supported in part by a DFG Postdoctoral Stipend and by the NSF under grant CCR-8957604. Work done in part while visiting the University of Rochester.

[8]Department of Mathematics, University of North Carolina, Greensboro, NC 27412, USA. Supported in part by the NSF under grant CCR-9108899.

Abstract

A set is P-selective [Sel79] if there is a polynomial-time semi-decision algorithm for the set—an algorithm that given any two strings decides which is "more likely" to be in the set. This paper studies two natural generalizations of P-selectivity: the NP-selective sets and the sets reducible or equivalent to P-selective sets via polynomial-time reductions. We establish a strict hierarchy among the various reductions and equivalences to P-selective sets. We show that the NP-selective sets are in (NP $\bigcap$ coNP)/poly, are extended low, and (those in NP) are Low_2; we also show that NP-selective sets cannot be NP-complete unless NP = coNP. By studying more general notions of nondeterministic selectivity, we conclude that all multivalued NP functions have single-valued NP refinements only if the polynomial hierarchy collapses to its second level.

Keywords: computational complexity, complexity classes, NPSV, NPMV, refinements, selectivity, lowness, advice classes, nonuniform classes, polynomial hierarchy NPSV-total, reductions, equivalences

ACM Computing Reviews Subject Categories: F.1.2, F.1.3

1 Introduction

Given the disappointingly small class of sets that are known to have polynomial-time decision algorithms, researchers have explored more flexible approaches to efficient set recognition (or near-recognition): almost polynomial time [MP79], average polynomial time (see [Gur91]), implicit membership testability [HH91], near-testability [GHJY91], P-closeness [Sch86,Yes83], P-selectivity [Sel79], and others. One of the more natural of these more general recognition notions is P-selectivity. Intuitively, a set is P-selective if there is a 2-ary polynomial-time function that chooses which of its inputs is "more likely" (or, actually, is "not less likely") to belong to the set.

Definition 1.1 [Sel79] A set A is P-selective if there is a total polynomial-time function f so that, for every $x, y \in \Sigma^*$,

1. $f(x, y) \in \{x, y\}$, and
2. if $x \in A$ or $y \in A$, then $f(x, y) \in A$.

We will use P-sel to denote the class of P-selective sets. The function f is said to be a *selector* (function) for A.

Since its introduction by Selman [Sel79,Sel82a,Sel82b] as a complexity-theoretic analog to the semirecursive sets [Joc68] from recursive function theory, P-selectivity has attracted a good deal of interest, and there have recently been many advances in our understanding of this class [Tod91,HNOS93,NOS93,BvHT93,BTvEB93]. Nonetheless, though polynomial-time reductions are usually considered natural in complexity theory, the classes of sets reducing to (and equivalent to) P-selective sets have not yet been adequately studied. Section 2 studies such classes and proves tight collapse and separation results. Let $\mathrm{R}_r^p(\text{P-sel})$ denote $\{A \mid (\exists L \in \text{P-sel})[A \leq_r^p L]\}$, that is, the class of sets that reduce to some P-selective set via r-type reductions. We prove that $\mathrm{R}_m^p(\text{P-sel}) \neq \mathrm{R}_{1\text{-}tt}^p(\text{P-sel}) \neq \cdots \mathrm{R}_{k\text{-}tt}^p(\text{P-sel}) \neq \mathrm{R}_{(k+1)\text{-}tt}^p(\text{P-sel}) \neq \cdots$, and $\mathrm{R}_{btt}^p(\text{P-sel}) \neq \mathrm{R}_{tt}^p(\text{P-sel}) \neq_{[\text{due to Watanabe}]} \mathrm{R}_T^p(\text{P-sel}) = \mathrm{P/poly}$. Also, we establish similarly strong separations for the classes of sets equivalent to P-selective sets via polynomial-time reductions. We also observe some collapses, in particular, the class of sets 1-truth-table equivalent to P-selective sets is identical to the class of sets 1-truth-table reducible to P-selective sets. Though our techniques bear no relation to the techniques used to study sparse sets, we note that reductions and equivalences to *sparse* sets have been satisfyingly studied in a long line of research (see, e.g., [BK88,TB91,AHOW92,GW,Ko89, AHH+,Gav92b,BLS93]); Section 2 of the present paper constructs, for P-selectivity, a theory roughly comparable in scope to the theory for sparse sets constructed in the just-mentioned line of papers.

The power of nondeterministic computation is one unifying theme of complexity theory. In light of this and the fact that P is the largest level of the polynomial hierarchy that is known to contain only P-selective sets, Section 3 studies the class of NP-selective sets—sets having an "NP function [BLS85]" that serves as a selector. (Formal definitions are given in Section 3.)

Self-reducibility (see the survey [JY90]) has been widely discussed as a property possessed by most "natural" sets, e.g., SAT. It is known that a Turing self-reducible set A is in P if and only if it is P-selective [BvHT93]. Analogously, we show that if a Turing self-reducible set A is NP-selective, then it is in NP $\bigcap$ coNP. This suggests that we might hope to use NP-selectivity as a tool for comparing NP and NP $\bigcap$ coNP. In fact, all sets in NP $\bigcap$ coNP are NP-selective, thus a Turing self-reducible set is in NP $\bigcap$ coNP if and only if it is NP-selective. Thus, if there is an NP-complete NP-selective set then NP = coNP. We generalize such claims to sets hard for NP with respect to flexible reductions, and we prove a related relationship between NP-hardness and promise problems modeling P-selectivity.

Section 3 also studies the lowness and nonuniform complexity of NP-selective sets. We show that NP-selective sets are of simple nonuniform complexity; all NP-selective sets are in (NP $\bigcap$ coNP)/poly. Though inclusion in the third level of the low hierarchy [Sch83] for all NP-selective sets in NP follows immediately from this, we show the stronger result that NP-selective sets are as low as P-selective sets: all NP-selective sets in NP are in the second level of the low hierarchy. This upper bound on the lowness of the NP-selective sets is optimal (with respect to relativizable proof techniques), due to the recently proven lower bound on the lowness of P-selective sets [AH92]. As to extended lowness [BBS86], we note that all NP-selective sets are $\mathrm{ExtendedLow}\Theta_3$, and we prove that a more general class, (NP/poly) $\bigcap$(coNP/poly), is $\mathrm{ExtendedLow}_3$.

We also introduce a more flexible notion of selectivity. This notion removes the requirements that the selector be total and single-valued. We obtain for this notion results on lowness, extended lowness, nonuniform characterizations, and characterizations in terms of the collapse of the polynomial hierarchy. As a consequence of our results, multivalued NP functions have single-valued NP refinements (cf. [FHOS93,Sel])—equivalently, some NP function can select a *single* satisfying assignment of any satisfiable formula given as input—only if the polynomial hierarchy collapses to $\mathrm{NP}^{\mathrm{NP}}$. We briefly discuss #P-selectivity, OptP-selectivity, SpanP-selectivity, and give a "nonuniform complexity" result that applies to these and other selectivity classes.

2 Equivalence and Reducibility to P-Selective Sets

For any class $\mathcal{C}$ and any reducibility denoted $\leq_r^t$, let $\mathrm{R}_r^t(\mathcal{C})$ and $\mathrm{E}_r^t(\mathcal{C})$ respectively denote $\{A \mid (\exists L \in \mathcal{C})[A \leq_r^t L]\}$ and $\{A \mid (\exists L \in \mathcal{C})[A \leq_r^t L \wedge L \leq_r^t A]\}$ [AHOW92]. In this section, we study the structure of the E(P-sel) and R(P-sel) classes stretching from P-sel = E_m^p(P-sel) = R_m^p(P-sel) up to R_T^p(P-sel), which equals P/poly (see [Sel79,Sel82b,Ko83]), the well-studied class of sets having small circuits [KL80]. Previous results along this line include only Buhrman et al.'s result[1] that R_{pos}^p(P-sel) = P-sel [BTvEB93] and Watanabe's result (referenced in [Tod91]) that $\mathrm{R}_{tt}^p(\text{P-sel}) \neq \mathrm{R}_T^p(\text{P-sel})$. For the case of sparse sets, reductions and equivalences have been studied intensely in a long line of papers [BK88,TB91, AHOW92,GW,Ko89,AHH$^+$,Gav92b,BLS93] that, unfortunately, rely upon techniques that fail to apply to the case of P-selective sets.

Below, we state our theorems summarizing the structure of reductions and equivalences to P-selective sets. Note that Selman [Sel82b] proved that the class of P-selective sets is strictly contained in E_{tt}^p(P-sel), a result that follows from either of Theorem 2.1 and Theorem 2.2.

Theorem 2.1

1. [Sel82b,BTvEB93] P-sel = $\mathrm{R}_m^p(\text{P-sel}) = \mathrm{R}_{ptt}^p(\text{P-sel}) = \mathrm{R}_{pos}^p(\text{P-sel}) = \mathrm{E}_m^p(\text{P-sel})$.
2. P-sel $\subsetneq \mathrm{R}_{1\text{-}T}^p(\text{P-sel}) = \mathrm{R}_{1\text{-}tt}^p(\text{P-sel}) = \mathrm{E}_{1\text{-}T}^p(\text{P-sel}) = \mathrm{E}_{1\text{-}tt}^p(\text{P-sel}) \subsetneq \mathrm{R}_{2\text{-}tt}^p(\text{P-sel}) \subsetneq \cdots \subsetneq \mathrm{R}_{k\text{-}tt}^p(\text{P-sel}) \subsetneq \mathrm{R}_{(k+1)\text{-}tt}^p(\text{P-sel}) \subsetneq \cdots$.
3. P-sel $\subsetneq \mathrm{R}_{1\text{-}T}^p(\text{P-sel}) \subsetneq \mathrm{R}_{2\text{-}T}^p(\text{P-sel}) \subsetneq \cdots \subsetneq \mathrm{R}_{k\text{-}T}^p(\text{P-sel}) \subsetneq \mathrm{R}_{(k+1)\text{-}T}^p(\text{P-sel}) \subsetneq \cdots$.
4. $\mathrm{R}_{btt}^p(\text{P-sel}) \subsetneq \mathrm{R}_{tt}^p(\text{P-sel})$.
5. (Watanabe, referenced in [Tod91]) $\mathrm{R}_{tt}^p(\text{P-sel}) \subsetneq \mathrm{R}_T^p(\text{P-sel})$.
6. ([Sel79][Sel82b, Theorem 12][Ko83, Theorem 3]) R_T^p(P-sel) = P/poly. (Meyer, as cited in [BH77], has noted that P/poly = R_T^p(SPARSE), and P/poly is also well-known to equal R_{tt}^p(TALLY).)

Theorem 2.2

1. P-sel $\subsetneq \mathrm{E}_{1\text{-}tt}^p(\text{P-sel}) \subsetneq \mathrm{E}_{2\text{-}tt}^p(\text{P-sel}) \subsetneq \cdots \subsetneq \mathrm{E}_{k\text{-}tt}^p(\text{P-sel}) \subsetneq \mathrm{E}_{(k+1)\text{-}tt}^p(\text{P-sel}) \subsetneq \cdots$.
2. P-sel $\subsetneq \mathrm{E}_{1\text{-}T}^p(\text{P-sel}) \subsetneq \mathrm{E}_{2\text{-}T}^p(\text{P-sel}) \subsetneq \cdots \subsetneq \mathrm{E}_{k\text{-}T}^p(\text{P-sel}) \subsetneq \mathrm{E}_{(k+1)\text{-}T}^p(\text{P-sel}) \subsetneq \cdots$.

[1] pos denotes positive Turing reductions. We use, for this and other polynomial-time reductions, the notations and definitions standard in the literature [LLS75,Sel82b].

3. $\mathrm{E}^p_{btt}(\text{P-sel}) \subsetneqq \mathrm{E}^p_{tt}(\text{P-sel}) \subsetneqq \mathrm{E}^p_T(\text{P-sel})$.

Theorem 2.3 For every $k \geq 1$ and every P-selective set A it holds that $\mathrm{R}^p_{k\text{-}T}(A) = \mathrm{R}^p_{(2^k-1)\text{-}tt}(A)$. In particular, for every $k \geq 1$ it holds that $\mathrm{R}^p_{k\text{-}T}(\text{P-sel}) = \mathrm{R}^p_{(2^k-1)\text{-}tt}(\text{P-sel})$. For every $k \geq 2$ it holds that $\mathrm{E}^p_{k\text{-}T}(\text{P-sel}) \subsetneqq \mathrm{E}^p_{(2^k-1)\text{-}tt}(\text{P-sel})$.

In fact, we will prove these results via more general results—Theorems 2.4, 2.10, 2.13, and 2.15—that in some cases achieve very tight "cross-separations" between equivalence and reducibility classes.

Theorem 2.4 There is a sparse set in $\mathrm{R}^p_{2\text{-}tt}(\text{P-sel}) - \mathrm{E}^p_T(\text{P-sel})$.

We now introduce some machinery that will be used in the proofs in this section.

We say that an arity two function f is a *partial selector on* a finite set $Q \subseteq \Sigma^*$ if for every $x, y \in Q$, it holds that $f(x, y) \in \{x, y\}$. Given a finite set Q and a partial selector f on Q, define a directed graph $G_{f,Q} = (Q, A)$ as follows:

$$(a, b) \in A \text{ if and only if either } f(a, b) = b \text{ or } f(b, a) = b.$$

We say that a and $b \in Q$ are *equivalent with respect to f in Q* if there is a loop in $G_{f,Q}$ containing both a and b; namely, a is reachable from b and b is reachable from a in $G_{f,Q}$. Obviously, the equivalence relation is reflexive, symmetric, and transitive.

Definition 2.5 Let f be a partial selector on a finite set Q and let $Q_1, \cdots, Q_m$ be nonempty subsets of Q. We say that $[Q_1, \cdots, Q_m]$ is a *decomposition of Q with respect to f* if the following conditions are satisfied:

1. $Q_1 \bigcup \cdots \bigcup Q_m = Q$,
2. for every i, j, $1 \leq i < j \leq m$, it holds that $Q_i \bigcap Q_j = \emptyset$,
3. for every i, j, $1 \leq i < j \leq m$, and for every $a \in Q_i$ and $b \in Q_j$, it holds that $f(a, b) = f(b, a) = b$ and a and b are not equivalent with respect to f in Q, and
4. for every i, $1 \leq i \leq m$, and for every $a, b \in Q_i$, a and b are equivalent with respect to f in Q.

Fact 2.6 Let f be a partial selector on Q. There exists a unique decomposition of Q with respect to f.

Proof Let f be a partial selector on Q. According to the equivalence relation, Q is partitioned into transitive closures, say $R_1, \cdots, R_m$, so that any two elements in different closures are not equivalent. For any a and b in different closures, it holds that $f(a,b) = f(b,a)$; otherwise, a and b are equivalent, yielding a contradiction. Moreover, for any $i, j, i \neq j$, and for any $a, b \in R_i$ and $c, d \in R_j$, it holds that $f(a,c) = c$ if and only if $f(b,d) = d$. Otherwise, for some i, j, $i \neq j$, and for some $a, b \in R_i$ and $c, d \in R_j$, it holds that $f(a,c) = c$ and $f(b,d) = b$, and since there is a path from c to d and a path from b to a in $G_{f,Q}$, there is a loop containing a and c, which yields a contradiction. For each $i, 1 \leq i \leq m$, let r_i be the lexicographically smallest string in R_i. By the above discussion, there is a unique ordering $r_{l_1}, \cdots, r_{l_m}$, such that for any $j, k, 1 \leq j < k \leq m$, it holds that $f(r_{l_j}, r_{l_k}) = f(r_{l_k}, r_{l_j}) = r_{l_k}$. For each $i, 1 \leq i \leq m$, let $Q_i = R_{l_i}$ Clearly, $[Q_1, \cdots, Q_m]$ is a decomposition. Since the partition according to the equivalence relation is uniquely determined, the decomposition is unique. This proves the fact. ∎

Fact 2.7 Let f be a selector for a set X and let $[Q_1, \cdots, Q_m]$ be the decomposition of a finite set Q with respect to f. Then there exists a unique i, $0 \leq i \leq m$, such that

$$Q_1, \cdots, Q_i \subseteq \overline{X} \text{ and } Q_{i+1}, \cdots, Q_m \subseteq X.$$

Proof Let X, Q, $Q_1, \cdots, Q_m$, and f be as in the hypothesis. By definition, for every i, $1 \leq i \leq m$, and for every $a, b \in Q_i$, there exist paths $(c_1, \cdots, c_k)$ with $c_1 = a$ and $c_k = b$ and $(d_1, \cdots, d_l)$ with $d_1 = b$ and $d_l = a$ in $G_{f,Q}$. It holds that

$$a \in X \rightarrow c_2 \in X \rightarrow \cdots \rightarrow c_{k-1} \in X \rightarrow b \in X \rightarrow d_2 \in X \rightarrow \cdots \rightarrow d_{l-1} \in X \rightarrow a \in X.$$

So, $a \in X$ if and only if $b \in X$. Thus, for any i, either $Q_i \subseteq X$ or $Q_i \subseteq \overline{X}$. Moreover, for every $i < j$ and for every $a \in Q_i$ and $b \in Q_j$, $f(a,b) = f(b,a) = b$, so $a \in X$ implies $b \in X$. Thus, $Q_i \subseteq X$ implies $Q_i, \cdots, Q_m \subseteq X$. ∎

Fact 2.8 Let f be a partial selector on Q with $||\,Q\,|| = m$ and $Q \subseteq (\Sigma^*)^{\leq n}$. Suppose $f(x,y)$ is computable in time $(|x| + |y|)^c$. Then the decomposition of Q with respect to f is computable time polynomial in $m \cdot n^c$.

Proof In order to find the decomposition of Q with respect to f, we have only to compute transitive closures in $G_{f,Q}$. Constructing the graph can be done in time polynomial in $m \cdot n^c$. Reachability between two vertices in the graph can be tested in time polynomial in $m \cdot n^c$, so computing transitive closures can be done in time polynomial in $m \cdot n^c$. Sorting the transitive closures can be done in time polynomial in $m \cdot n^c$, too. So, the total running time for computing the decomposition is bounded by some polynomial in $m \cdot n^c$. ∎

Throughout this paper, M_1, M_2, $\cdots$ is a standard enumeration of polynomial-time deterministic oracle Turing machines, where M_i runs, independent of its oracle, in time $n^i + i$

on inputs of length n. $N_1, N_2, \cdots$ is a similar standard enumeration of polynomial-time nondeterministic Turing machines. Often, to avoid subsubscripts, we refer to arbitrary machines or constructed machines as, e.g., N_1; this is not meant to imply that the machines necessarily are the actual first Turing machine, but is just a shorthand for a more precise notation, e.g., N_{i_1}. $f_1, f_2, \cdots$ is a standard enumeration of all polynomial-time computable 2-ary functions. We assume that $f_i(x, y)$ is computable in time $(|x| + |y|)^i + i$ for every x and y. Define $\mu(n)$ by: $\mu(0) = 2^{2^2}$, and $\mu(n) = 2^{2^{2^{\mu(n-1)}}}$ for $n \geq 1$.

Let U be a finite subset of Σ^*. Let $x_1, \cdots, x_m$ be the enumeration of all elements in U in increasing lexicographic order. For $A \subseteq U$ and $B \subseteq U$, we say that A is *smaller* than B if

$$\chi_A(x_1) \cdots \chi_A(x_m) \text{ is lexicographically smaller than } \chi_B(x_1) \cdots \chi_B(x_m),$$

where $\chi_A(x) = 1$ ($\chi_B(x) = 1$) if $x \in A$ ($x \in B$) and 0 otherwise. Via this order, 2^U is totally ordered for any finite subset U of Σ^*.

Proof of Theorem 2.4 We will construct a sparse set S in stages. S contains at most one string of each length and it contains only strings of length $\mu(s)$ for some s. Each stage s determines the membership of strings of length $n = \mu(s)$ and our construction is designed so that for every s, all strings that are put into S prior to stage s can be enumerated in time polynomial in $\mu(s + 1)$.

At stage $s = \langle i, j, l \rangle$, we will diagonalize against a pair of machines (M_i, M_j) and a function f_l, to establish the following requirement (R):

(R) for any set X with selector f_l, either $S \not\leq_T^p X$ via M_i or $X \not\leq_T^p S$ via M_j.

Our construction proceeds as follows. Let $n = \mu(s)$ and let S' be the set of all strings put into S prior to stage s. Let Q be the set of all possible query strings of M_i on x for some $x \in \Sigma^n$. Note that any element in Q is of length at most $n^i + i$ and that $\| Q \| \leq 2^n 2^{n^i + i} \leq 2^{2n^i}$.

First we check whether

(a) f_l is a partial selector on Q.

If this does not hold, then clearly (R) is satisfied. So we proceed to the next stage without adding any string to S.

Suppose that f_l is a partial selector on Q. Let $[Q_1, \cdots, Q_m]$ be the decomposition of Q with respect to f_l. We check whether the following conditions, (b) and (c), are satisfied.

(b) There exists some t such that

$$Q_1, \cdots, Q_t \subseteq \overline{L(M_j, S')} \text{ and } Q_{t+1}, \cdots, Q_m \subseteq L(M_j, S').$$

(c) For every $x \in \Sigma^n$, there exists some $t(x)$ such that

(c1) $Q_1, \cdots, Q_{t(x)} \subseteq \overline{L(M_j, S' \bigcup \{x\})}$ and

(c2) $Q_{t(x)+1}, \cdots, Q_m \subseteq L(M_j, S' \bigcup \{x\})$.

If (b) does not hold, we add nothing new to S. If (b) does hold and (c) does not hold, we pick the smallest x not satisfying (c) and put it into S. By doing this, we establish that for every X with selector f_l, $X \not\leq_T^p S$ via M_j, for there exist h and $a, b \in Q$ such that (i) $M_j^S(a)$ accepts, (ii) $M_j^S(b)$ rejects, and (iii) $a \in X$ implies $b \in X$. Thus (R) is satisfied. We proceed to the next stage.

Now suppose that (b) and (c) are satisfied. Note for any $x \in \Sigma^n$, $t(x)$ in condition (c) is uniquely determined. The following claim holds.

Claim 2.9 There exist $x, y \in \Sigma^n$ such that $x < y$ and $t(x) = t(y)$.

Proof of Claim 2.9 The proof is by contradiction. Assume that all $t(x)$ are distinct. Let r be the (2^{n-1})-th largest $t(x)$ and let $U = \{x \mid t(x) \geq r\}$ and and let $V = \{x \mid t(x) < r\}$. Let $q \in Q_r$. Then $M_j^{S' \cup \{x\}}(q)$ rejects for every $x \in U$ and $M_j^{S' \cup \{x\}}(q)$ accepts for every $x \in V$. Suppose that $M_j^{S'}(q)$ accepts. Since $M_j^{S' \cup \{x\}}(q)$ rejects for every $x \in U$, every $x \in U$ is queried by $M_j^{S'}(q)$. On the other hand, suppose that $M_j^{S'}(q)$ rejects. Since $M_j^{S' \cup \{x\}}(q)$ accepts for every $x \in V$, every $x \in V$ is queried by $M_j^{S'}(q)$. So $M_j^{S'}(q)$ makes at least 2^{n-1} queries. This is impossible, however, because $M_j^{S'}(q)$ makes at most $|q|^j + j \leq (n^i + i)^j + j < 2^{n-1} - 1$ queries. ▮ **Claim 2.9**

By the above claim, there exist x and y such that $x < y$ and $t(x) = t(y)$. Let u and v be the smallest such x and y. For every $q \in Q$, it holds that

$$M_j(q)^{S' \cup \{u\}} \text{ accepts if and only if } M_j(q)^{S' \cup \{v\}} \text{ accepts.}$$

So M_i relative to $L(M_j, S' \bigcup \{u\})$ and M_i relative to $L(M_j, S' \bigcup \{v\})$ accept the same language. We put u into S if v is in the language and v into S otherwise. Then either (i) $v \notin S$ and $v \in L(M_i, L(M_j, S))$, or (ii) $v \in S$ and $v \notin L(M_i, L(M_j, S))$. Thus, (R) is satisfied and we proceed to the next stage.

It remains to show that S is $\leq_{2\text{-}tt}^p$-reducible to some P-selective set. First we show that the above construction can be done in time polynomial in $\mu(s+1)$. Each string in Q is of length at most n^i, and $||\, Q \,|| \leq 2^{2n^i}$. Computing $f_l(x, y)$ for each $x, y \in Q$ can be done in time $\mathcal{O}(n^{il})$. So checking condition (a) can be done in time $\mathcal{O}((2^{2n^i})^2 \cdot n^{il}) \leq \mathcal{O}(2^{2^n})$. By Fact 2.8, the decomposition of Q can be computed in time polynomial in $||\, Q \,|| \cdot n^{il} \leq 2^{2^n}$.

Since there are 2^n strings in Σ^n and $||\,Q\,|| \leq 2^{2n^i}$, checking condition (b) and (c) and computing all $t(x)$ can be done in time $\mathcal{O}(2^n \cdot 2^{2n^i} \cdot n^{ij}) \leq \mathcal{O}(2^{2^n})$. After computing all $t(x)$, we can easily determine the string put into S. So the construction at stage s can be done in time polynomial in $2^{2^n} = \mu(s+1)$.

Define A to be the set of all strings x for which there is some $y \in S$ such that $x \leq y$ and $|x| = |y|$. Clearly, for every x, $x \in S$ if and only if $x \in A$ and $x' \notin A$, where x' is successor of x. So $S \leq^p_{2\text{-}tt} A$. We claim that A is P-selective. Let $g(x, y)$ be the function computed as follows:

(Case 1) $|x| = \mu(s)$ and $|y| = \mu(t)$:

(Subcase 1a) $|x| = |y|$: $g(x, y) = \min\{x, y\}$.

(Subcase 1b) $|x| < |y|$: Simulate the construction of S up to stage s, and compute $w = S^{|x|}$. $g(x, y) = x$ if $w \geq x$ and y otherwise.

(Subcase 1c) $|x| > |y|$: Simulate the construction of S up to stage t, and compute $w = S^{|y|}$. $g(x, y) = y$ if $w \geq y$ and x otherwise.

(Case 2) $|x| \neq \mu(s)$ for any s: $g(x, y) = y$.

(Case 3) $|y| \neq \mu(t)$ for any t: $g(x, y) = x$.

It is not hard to see that g is polynomial-time computable. We claim that g is a P-selector for A. In order to see this property, we only have to consider the case $x \in A$ and $y \notin A$. Since $x \in A$, only (Case 1) and (Case 3) can occur. If (Subcase 1a) occurs, $x < y$, so $g(x, y) = x$. If (Subcase 1b) occurs, $x \leq w$, so $g(x, y) = x$. If (Subcase 1c) occurs, $y > w$, so $g(x, y) = x$. If (Subcase 3) occurs, $|y| \neq \mu(t)$ for any t, so $g(x, y) = x$. In any case, $g(x, y) = x$, so g is a P-selector for A. This proves the theorem. ∎

Theorem 2.10 For each $k \geq 2$, there is a tally set in $\mathrm{E}^p_{k\text{-}tt}(\text{P-sel}) - \mathrm{R}^p_{(k-1)\text{-}tt}(\text{P-sel})$.

Proof Let $k \geq 2$. We will construct a tally set T and a P-selective set A in stages so that T and A are $\leq^p_{k\text{-}tt}$ equivalent. Also, at stage $s = \langle i, j\rangle$, we will diagonalize against machine M_i and function f_j so that the following requirement (R) is satisfied:

(R) for any set X with selector f_j, $T \not\leq^p_{(k-1)\text{-}tt} X$ via M_i.

T will contain only strings of length $k\mu(s) + l$, $1 \leq l \leq k$, A will contain only strings of length $k\mu(s)$, and $A^{=k\mu(s)}$ will be an initial segment of $\Sigma^{k\mu(s)}$ of size at most k^2. During the construction at stage s, for each l, $1 \leq l \leq k$, x_l denotes $0^{k\mu(s)+l}$ and for each l, $1 \leq l \leq k^2$, y_l denotes the l-th smallest string in $\Sigma^{k\mu(s)}$. We need one technical fact, which will be shown below.

Let g be a mapping of $\{1, \cdots, k^2\}$ to $\{1, \cdots, k\}$. For each $a, 0 \leq a \leq k^2$, and for each $l, 1 \leq l \leq k$, define $\gamma_g^a(l) = 1$ if the number of $b \leq a$ such that $g(b) = l$ is odd and $\gamma_g^a(l) = 0$ otherwise. For each $a, 0 \leq a \leq k^2$, define $\Gamma_g^a = (\gamma_g^a(1), \cdots, \gamma_g^a(k))$.

Fact 2.11 There is an onto mapping g of $\{1, \cdots, k^2\}$ to $\{1, \cdots, k\}$ satisfying the following conditions:

(c1) for every $l, 1 \leq l \leq k$, the number of a's such that $1 \leq a \leq k^2$ and $g(a) = l$ is k,

(c2) for every $a \geq k^2 - k + 2$, $g(a) = k$, and

(c3) for every a, b, $0 \leq a < b \leq k^2 - k + 1$, it holds that $\Gamma_g^a \neq \Gamma_g^b$.

Proof of Fact 2.11 The proof is by induction on k. For the induction's base case, let $k = 2$. Define $g(1) = 1$, $g(2) = 2$, $g(3) = 1$, and $g(4) = 2$. Clearly, (c1) is satisfied. Furthermore, $\Gamma_g^0 = (0, 0)$, $\Gamma_g^1 = (1, 0)$, $\Gamma_g^2 = (1, 1)$, $\Gamma_g^3 = (0, 1)$, and $\Gamma_g^4 = (0, 0)$. Thus (c2) and (c3) are satisfied.

For the inductive step, let $k \geq 2$ and suppose that the claim holds for k and this is witnessed by a mapping h. We define g for $k + 1$ as follows:

1. for every a, $0 \leq a \leq k^2 - k + 1$, $g(a) = h(a)$,
2. $g(k^2 - k + 2) = k + 1$,
3. $g(k^2 - k + 3) = g(k^2 - k + 5) = \cdots = g(k^2 + k - 1) = g(k^2 + k + 1) = k$,
4. for every l, $1 \leq l \leq k - 1$, $g(k^2 - k + 2 + 2l) = l$, and
5. for every $a \geq (k + 1)^2 - (k + 1) + 2$, $g(a) = k + 1$.

Clearly, condition (c2) is satisfied. We show that (c1) and (c3) are both satisfied. By inductive hypothesis, for every $l, 1 \leq l \leq k + 1$, the number of $a \leq k^2 - k + 1$ such that $g(a) = l$ is k if $l \leq k - 1$, 1 if $l = k$, and 0 if $l = k + 1$. By definition, for every $l, 1 \leq l \leq k + 1$, the number of $a > k^2 - k + 1$ such that $g(a) = l$ is 1 if $l \leq k - 1$, k if $l = k$, and $k + 1$ if $l = k + 1$. So, for every $l, 1 \leq l \leq k + 1$, the number of a such that $g(a) = l$ is $k + 1$, and thus, (c1) is satisfied.

From (1) above, for every $a, 0 \leq a \leq k^2 - k + 1$, $\Gamma_g^a = (\gamma_h^a(1), \cdots, \gamma_h^a(k), 0)$. So, by inductive hypothesis, for every $a, b, 0 \leq a < b \leq k^2 - k + 1$, $\Gamma_g^a \neq \Gamma_g^b$. There is only one $a, k^2 - k + 2 \leq a \leq (k + 1)^2 - (k + 1) + 1$, such that $g(a) = k + 1$. So, for every $a, k^2 - k + 2 \leq a \leq (k + 1)^2 - (k + 1) + 1$, Γ_g^a has a 1 as its $(k + 1)$-st entry, and thus, for every a and $b, 0 \leq a \leq k^2 - k + 1 < b \leq (k + 1)^2 - (k + 1) + 1$, $\Gamma_g^a \neq \Gamma_g^b$. For any sets L_1 and L_2, let $L_1 \oplus L_2 = \{0x \mid x \in L_1\} \bigcup \{1y \mid y \in L_2\}$.

For every d, $1 \le d \le 2k$,

$$\Gamma_g^{k^2-k+1+d} \oplus \Gamma_g^{k^2-k+2} = (\underbrace{1,\cdots,1}_{\lfloor d/2 \rfloor}, \underbrace{0,\cdots,0}_{k-1-\lfloor d/2 \rfloor}, e_d, 0),$$

where $e_d = 1$ if $d \equiv 1$ or 2 modulo 4 and 0 otherwise. So, for every a and b, $k^2 - k + 2 \le a < b \le (k+1)^2 - (k+1) + 1$, it holds that $\Gamma_g^a \ne \Gamma_g^b$. Thus for every a and b, $0 \le a < b \le (k+1)^2 - (k+1) + 1$, it holds that $\Gamma_g^a \ne \Gamma_g^b$, and this establishes (c3). Thus, the claim holds for $k+1$, and this proves the fact. ▮ **Fact 2.11**

Now we describe the construction. Let T' be the set of all strings put into T prior to stage s and let $n = k\mu(s)$. First we check whether M_i makes $k-1$ nonadaptive queries on inputs $x_1, \cdots, x_k$. If this is not the case, M_i cannot be a $\le^p_{(k-1)\text{-}tt}$ reduction, and (R) is trivially satisfied, so we proceed to the next stage without adding any new elements to T.

Suppose M_i passes the above check. Let Q be the set of all query strings of M_i on $x_1, \cdots, x_k$. It holds that $\|Q\| \le k(k-1)$. We check whether f_j is a partial selector on Q. If this is not the case, (R) is satisfied, so we proceed to the next stage without adding any new elements to T.

Suppose that f_j is a partial selector on Q. Let $[Q_1, \cdots, Q_m]$ be the decomposition of Q with respect to f_j. Since $\|Q\| \le k(k-1)$, $m \le k(k-1)$. By Fact 2.7, for every X with selector f_j, there is some t such that $Q_1, \cdots, Q_t \subseteq \overline{X}$ and $Q_{t+1}, \cdots, Q_m \subseteq X$. For t, $0 \le t \le m$, and l, $1 \le l \le k$, let $\tau_t(l)$ be 1 if $M_i(x_l)$ accepts relative to $T' \bigcup (Q_{t+1} \bigcup \cdots \bigcup Q_m)$ and 0 otherwise, and let $\tau_t = (\tau_t(1), \cdots, \tau_t(k))$. Let g be the function in Fact 2.11 for k. Since $0 \le m \le k^2 - k$ and $\Gamma_g^a \ne \Gamma_g^b$ for every a and b, $0 \le a < b \le k^2 - k + 1$, there is at least one a such that $\Gamma_g^a \ne \tau_t$ for every t. Let c be the smallest such a. For each $l, 1 \le l \le k$, we add x_l to T if and only if $\gamma_g^c(l) = 1$, and we add $y_1, \cdots, y_c$ to A. Then, for any set X with selector f_j, there is some l such that $x_l \in T$ if and only if $M_i^X(x_l)$ rejects. So (R) is satisfied. We proceed to the next stage.

We claim that A and T are $\le^p_{k\text{-}tt}$ equivalent. Note for any l, that

$$\begin{aligned} x_l \in T &\iff \gamma_g^c(l) = 1 \\ &\iff \#\{b \mid b \le c, g(b) = l\} \text{ is odd} \\ &\iff \#\{b \mid 1 \le b \le k^2, g(b) = l, b \le c\} \text{ is odd} \\ &\iff \#\{b \mid 1 \le b \le k^2, g(b) = l, y_b \in A\} \text{ is odd.} \end{aligned}$$

Let B_l be the set of all b such that $g(b) = l$. Then $x_l \in T$ if and only if $\|B_l \bigcap A\|$ is odd. By definition, $\|B_l\| = k$. Since B_l is easily computable, we have $T \le^p_{k\text{-}tt} A$. On the other hand, it holds that $\Gamma_g^c = (\chi_T(x_1), \cdots, \chi_T(x_k))$. So, by Fact 2.11, c can be computed from $\chi_T(x_1), \cdots, \chi_T(x_k)$. For any l, $y_l \in A$ if and only if $l \le c$. So, $A \le_{k\text{-}tt}^p T$.

Now we show that A is P-selective. Let $g(x, y)$ be a function defined as follows:

(Case 1) $|x| = k\mu(s)$ and $|y| = k\mu(t)$ for some s, t:

(Subcase 1a) $|x| = |y|$: $g(x, y) = \min\{x, y\}$.

(Subcase 1b) $|x| < |y|$: Simulate the construction of A up to stage s, and compute the largest string w in $A^{=|x|}$. $g(x, y) = x$ if $w \ge x$ and y otherwise.

(Subcase 1c) $|x| > |y|$: Simulate the construction of A up to stage t, and compute the largest string w in $A^{=|y|}$. $g(x, y) = y$ if $w \ge y$ and x otherwise.

(Case 2) $|x| \ne k\mu(s)$ for any s: $g(x, y) = y$.

(Case 3) $|y| \ne k\mu(t)$ for any t: $g(x, y) = x$.

From Fact 2.8, the construction at stage s can be done in time polynomial in $2^{2^{k\mu(s)}}$, so it is done in time $\mathcal{O}(\mu(s+1))$. Thus g is polynomial-time computable. So it suffices to show that g is a selector for A. In order to see this, we have only to consider the case in which exactly one of x or y is in A. Suppose $x \in A$ and $y \notin A$. Since $x \in A$, only (Case 1) and (Case 3) can occur. If (Subcase 1a) occurs, $x < y$, so $g(x, y) = x$. If (Subcase 1b) occurs, $x \le w$, so $g(x, y) = x$. If (Subcase 1c) occurs, $y > w$, so $g(x, y) = x$. If (Case 3) occurs, $|y| \ne \mu(t)$ for any t, so $g(x, y) = x$. Thus $g(x, y) = x$. The above argument is symmetric, so $g(x, y) = y$ when $x \notin A$ and $y \in A$. Hence, g is a P-selector for A. ∎

Corollary 2.12 For each $k \ge 1$, there is a tally set in $\mathrm{E}_{2^k\text{-}tt}^p(\text{P-sel}) - \mathrm{R}_{k\text{-}T}^p(\text{P-sel})$.

Proof Follows immediately from the previous theorem and the standard fact that $A \le_{\ell\text{-}T}^p B$ implies $A \le_{(2^\ell-1)\text{-}tt}^p B$. ∎

Theorem 2.13 There is a tally set in $\mathrm{E}_{tt}^p(\text{P-sel}) - \mathrm{E}_{btt}^p(\text{P-sel})$.

Proof We will construct a tally set T and a P-selective set A in stages. At stage $s = \langle i, j, k, l\rangle$, we will diagonalize against M_i, M_j so that the following requirement is satisfied:

(R) for any set X with selector f_l, either $T \not\le_{k-tt}^p X$ via M_i or $X \not\le_{k-tt}^p T$ via M_j.

The construction at stage s proceeds as follows: Let T' be the set of all strings put into T prior to stage s. Let $n = 2^{\mu(s)}$. Throughout this paper, we will take all logs to be base two. For each $p, 1 \le p \le \log n = \mu(s)$, let $x_p = 0^{n+p}$, and for each $p, 1 \le p \le n$, let y_p be the p-th smallest string in Σ^n. Note that $\{y_1, \ldots, y_n\} = \{0^{n-\log n} a \mid a \in \Sigma^{\log n}\}$.

Let Q be the set of all query strings of M_i on $x_1, \cdots, x_{\log n}$. First, as in the previous theorems, we check the following conditions:

(a) M_i makes at most k nonadaptive queries on inputs $x_1, \cdots, x_{\log n}$.

(b) f_l is a partial selector on Q.

(c) For every $q \in Q$, M_j on q makes at most k nonadaptive queries.

If one of (a), (b), and (c) does not hold, (R) is trivially satisfied. If this is the case, we proceed to the next stage without putting any new elements into T. Suppose (a), (b), and (c) hold. Let $[Q_1, \cdots, Q_m]$ be the decomposition of Q with respect to f_l. We check whether the following condition is satisfied:

(d) For any subset W of $\{x_1, \cdots, x_{\log n}\}$, there exists $t(W)$ such that

$$Q_1, \cdots, Q_{t(W)} \subseteq \overline{L(M_j, W)} \text{ and } Q_{t(w)+1}, \cdots, Q_m \subseteq L(M_j, W).$$

If (d) does not hold for some W, then let W_0 be the smallest such W and set $T = T \bigcup W_0$. Clearly, (R) is satisfied. Now suppose that (d) holds. We make the following claim.

Claim 2.14 There exist $W, W' \subseteq \{x_1, \cdots, x_{\log n}\}$ such that $W \neq W'$ and $t(W) = t(W')$.

Proof of Claim 2.14 The proof is by contradiction. Assume for every $W, W' \subseteq \{x_1, \cdots, x_{\log n}\}$ with $W \neq W'$, it holds that $t(W) \neq t(W')$. Let $t(W_0)$ be the largest amongst all $t(W), W \subseteq \{x_1, \cdots, x_{\log n}\}$, and let $q \in Q_{t(W_0)}$. For every $W \neq W_0$, $q \in L(M_j, W) - L(M_j, W_0)$. So M_j on q relative to W_0 makes queries to all $x_1, \cdots, x_n$. Since $n > l$, this is impossible. ▮ **Claim 2.14**

Let W_0, W_0' be the smallest pair such that $W \neq W'$ and $t(W) = t(W')$. Let $T_1 = T' \bigcup W_0$ and $T_2 = T' \bigcup W_0'$. Let $L = L(M_i, L(M_j, T_1))$. Since $t(W_0) = t(W_0')$, $L = L(M_i, L(M_j, T_2))$. So either $T_1 \triangle L$ or $T_2 \triangle L$ is nonempty. We add all strings in W_0 to T if $T_1 \triangle L \neq \emptyset$ and all strings in W_0' to T otherwise. Then there exists some $i, 1 \leq i \leq \log n$, such that $x_i \notin T$ if and only if $x_i \in L = L(M_i, L(M_j, T))$. Thus (R) is satisfied.

Now, define $z = 0^{n - \log n} b_1 \cdots b_{\log n}$, where $b_i = 1$ if $x_i \in T$ and 0 otherwise. Put all $y_i \leq z$ in A. $A^{=n}$ is an initial segment of Σ^n. Clearly, $A \leq_{tt}^p T$. For every p, x_p is in T if and only if the p-th bit of the largest $y_i \in A$ is a 1, so $T \leq_{tt}^p A$. By an argument similar to those of the other proofs in this section, A is P-selective. ▮

Theorem 2.15 For each $k \geq 2$, there is a tally set in $\mathrm{E}_{k\text{-}tt}^p(\text{P-sel}) - \mathrm{E}_{(k-1)\text{-}T}^p(\text{P-sel})$.

Proof Let k be at least two. We will construct a tally set T and a P-selective set A in stages. At stage $s = \langle i, j, l\rangle$, we will diagonalize against M_i and M_j so that the following requirement (R) is satisfied:

(R) for any set X with selector f_l, either $T \not\leq^p_{(k-1)\text{-}T} X$ via M_i or $X \not\leq^p_{(k-1)\text{-}T} T$ via M_j.

The construction at stage s proceeds as follows: Let T' be the set of all strings put into T prior to stage s. Let $n = k\mu(s)$. For each p, $1 \leq p \leq k+1$, let $x_p = 0^{n+p}$, and let $y_1, \cdots, y_{k+1}$ be the smallest $k+1$ strings of Σ^n, in increasing order. Let Q be the set of all possible query strings of M_i on inputs $x_1, \cdots, x_{k+1}$. First, we check whether each of the following conditions is satisfied.

(a) M_i makes at most $k-1$ adaptive queries on inputs $x_1, \cdots, x_{k+1}$.

(b) f_l is a partial selector on Q.

(c) M_j makes at most $k-1$ adaptive queries on every input in Q.

If any one of (a), (b), or (c) fails to hold, then (R) is trivially satisfied. So we may proceed to the next stage without updating T. Suppose (a), (b), and (c) hold. Let $[Q_1, \cdots, Q_m]$ be the decomposition of Q with respect to f_l. We check whether the following condition is satisfied:

(d) For each p, $1 \leq p \leq k+1$, there exists t_p such that

$$Q_1, \cdots, Q_{t_p} \subseteq \overline{L(M_j, T \bigcup \{x_p\})} \text{ and } Q_{t_p+1}, \cdots, Q_m \subseteq L(M_j, T \bigcup \{x_p\}).$$

If (d) does not hold, let p_0 be the smallest such p, and set $T = T' \bigcup \{x_{p_0}\}$. Clearly, (R) is satisfied. Now suppose that (d) holds. We claim that

(e) there exist p and p', $1 \leq p < p' \leq k+1$, such that $t_p = t_{p'}$.

This is seen as follows: Assume (e) does not hold. Let d be such that t_d is the largest amongst all t_p and let $q \in Q_{t_d}$. For every $p \neq d$, it holds that $q \in L(M_j, T' \bigcup \{x_p\}) - L(M_j, T' \bigcup \{x_d\})$. Each oracle $T' \bigcup \{x_p\}$ contains only one string in $\{x_1, \cdots, x_{k+1}\}$. So M_j on q relative to T makes queries to all $x_1, \cdots, x_{k+1}$ except x_d. Thus M_j on q relative to T makes k queries. But, as we are assuming that (c) is satisfied, this is impossible. Thus (e) holds.

Let p_0 and p'_0 be the smallest p and p' such that $p < p'$ and $t_p = t_{p'}$. Set $T_1 = T \bigcup \{x_{p_0}\}$, let $T_2 = T \bigcup \{x_{p'_0}\}$, and let z be the smallest string that belongs to exactly one of T_1 and T_2. It holds that $z \in L(M_i, L(M_j, T_1)) = L(M_i, L(M_j, T_2))$. So either $T_1 \neq L(M_i, L(M_j, T_1))$ or

$T_2 \neq L(M_i, L(M_j, T_2))$. If the primary condition holds, set $T = T_1$; otherwise, set $T = T_2$. Then (R) is satisfied.

Now, define $Y \subseteq \Sigma^n$ so that $Y = \{y_1, \cdots, y_d\}$ if and only if T contains x_d. Set $A = A \bigcup Y$. The size of the segment $\{y_1, \cdots, y_d\}$ can be computed by checking the membership of all $y_2, \cdots, y_{k+1}$ (y_1 is always in A). So T is $\leq^p_{k\text{-}tt}$-reducible to A. On the other hand, y_1 is always in A, and for every $p \geq 2$, y_p is in A if and only if $\{x_p, \cdots, x_{k+1}\} \bigcap T \neq \emptyset$. So $A \leq^p_{k\text{-}tt} T$. Thus $T \equiv^p_{k\text{-}tt} A$. Via an analysis similar to that of the proof of Theorem 2.13, A is P-selective. ∎

Theorems 2.1 (excluding the parts cited to others), 2.2, and 2.3 follow from results 2.4 through 2.15, plus Proposition 2.16 below. Results 2.4 through 2.15 clearly yield such strong "cross-separations" as, for example, results 2.17 through 2.19.

Proposition 2.16 $\mathrm{R}^p_{1\text{-}T}(\text{P-sel}) = \mathrm{R}^p_{1\text{-}tt}(\text{P-sel}) = \mathrm{E}^p_{1\text{-}T}(\text{P-sel}) = \mathrm{E}^p_{1\text{-}tt}(\text{P-sel})$.

Proof It suffices to show that $\mathrm{R}^p_{1\text{-}T}(\text{P-sel}) \subseteq \mathrm{E}^p_{1\text{-}T}(\text{P-sel})$. Let L be $\leq^p_{1\text{-}tt}$-reducible to a P-selective set A via a machine M. Define $B = \{\langle x, y\rangle \mid M$ on x queries y, the acceptance of $M(x)$ depends on the oracle answer to y, and $y \in A\}$. Clearly, B is P-selective and $L \leq^p_{1\text{-}tt} B$. Consider a machine D that, given $w = \langle x, y\rangle$, simulates M on x and, behaves as follows:

(1) if $M(x)$ does not query y, then D rejects w,

(2) if $M(x)$ queries y, but ignores the answer to y, then D rejects w,

(2) if $M(x)$ queries y and accepts x iff the answer is YES, then D accepts w if and only if $x \in L$, and

(3) if $M(x)$ queries y and accepts x iff the answer is NO, then D accepts w if and only if $x \notin L$.

It is not hard to see that D is a $\leq^p_{1\text{-}tt}$ reduction of B to L. ∎

Corollary 2.17 $\mathrm{R}^p_T(\text{P-sel})$, $\mathrm{E}^p_T(\text{P-sel})$, $\mathrm{R}^p_{tt}(\text{P-sel})$, and $\mathrm{E}^p_{tt}(\text{P-sel})$ are pairwise distinct.

Corollary 2.18

1. $\mathrm{R}^p_{btt}(\text{P-sel}) \not\subseteq \mathrm{E}^p_T(\text{P-sel})$.
2. $\mathrm{E}^p_{btt}(\text{P-sel}) \subsetneq \mathrm{R}^p_{btt}(\text{P-sel})$.

Corollary 2.19 For each $k \geq 2$:

1. there is a tally set in $\mathrm{E}^p_{k\text{-}tt}(\text{P-sel}) - \mathrm{E}^p_{(k-1)\text{-}tt}(\text{P-sel})$,

2. there is a tally set in $\mathrm{E}^p_{k\text{-}T}(\text{P-sel}) - \mathrm{E}^p_{(k-1)\text{-}T}(\text{P-sel})$,

3. there is a tally set in $\mathrm{R}^p_{k\text{-}tt}(\text{P-sel}) - \mathrm{R}^p_{(k-1)\text{-}tt}(\text{P-sel})$, and

4. there is a tally set in $\mathrm{R}^p_{k\text{-}T}(\text{P-sel}) - \mathrm{R}^p_{(k-1)\text{-}T}(\text{P-sel})$.

Finally, we note that Selman's early result [Sel79, Theorem 5] that "every tally set is $\leq^p_T$-equivalent to some P-selective set" is essentially optimal. It is not hard to see that this optimality follows as a consequence of a recent result of Watanabe (see part 5 of Theorem 2.1). Alternatively, it is not hard to see the optimality directly. As Watanabe's result is cited by Toda without proof as a personal communication, for completeness we include a direct proof below of the optimality of Selman's result.

Corollary 2.20 There is a tally set that is not in $\mathrm{R}^p_{tt}(\text{P-sel})$.

Proof We will construct T in stages. At stage $s = \langle i, j\rangle$, we diagonalize against machine M_i and function f_j. More precisely, we put tally strings into T (if necessary) so that the following requirement (R) is satisfied:

(R) for any set X with selector f_j, M_i does not $\leq^p_{tt}$ reduce T to X.

The construction at stage $s = \langle i, j\rangle$ proceeds as follows: Let T' be the set of all strings put into T prior to stage s. At the initial stage, $T' = \emptyset$. For simplicity, let $n = \mu(s)$ and for each $i, 1 \leq i \leq n$, let $x_i = 0^{n+i}$.

Let Q be the set of all query strings of M_i on inputs $x_1, \cdots, x_n$. First we check the following conditions:

(a) M_i makes nonadaptive queries on inputs $x_1, \cdots, x_n$.

(b) f_j is a partial selector on Q.

If (a) is not satisfied, then M_i is not a $\leq^p_{tt}$ reduction, and thus (R) is satisfied. If (b) is not satisfied, then f_j is not a selector, and thus (R) is satisfied. So, if either (a) or (b) does not hold, we proceed to the next stage without adding any new elements to T,

Let $[Q_1, \cdots, Q_m]$ be the decomposition of Q with respect to f_j. From Fact 2.7, for any set X with selector f_j, there exists a unique $t(X)$ such that $Q_1, \cdots, Q_{t(X)} \subseteq \overline{X}$ and $Q_{t(X)+1}, \cdots, Q_m \subseteq X$. Since Q is the set of all query strings of M_i on inputs $x_1, \cdots, x_n$, for every set X with selector f_j, there uniquely exists $t(X), 0 \leq t(X) \leq m$ such that for every $i, 1 \leq i \leq n$, $x_i \in L(M_i, X)$ if and only if $x_i \in L(M_i, Q_{t(x)} \bigcup \cdots \bigcup Q_m)$. Since M_i makes nonadaptive queries and M_i runs in time $n^i + i$ on inputs $x_1, \cdots, x_n$, $Q \subseteq (\Sigma^*)^{\leq n^i+i}$ and $||\, Q\,|| \leq n((2n)^i + i) < 2^n - 1$. So $t(X)$ has less than 2^n possible values. On the other hand, there are 2^n different possible assignments of $x_1, \cdots, x_n$ into T. So there exists $U \subseteq \{x_1, \cdots, x_n\}$ such that for every $t, 0 \leq t \leq m$, there exists some x_d such that $x_d \in U$ if and only if $x_d \notin L(M_i, Q_t \bigcup \cdots \bigcup Q_m)$. Let V be the smallest such U. For every P-selective set X with selector f_j, there exists some x_d such that $x_d \in V$ if and only if $x_d \notin L(M_i, X)$. Set $T = T' \bigcup V$. So we have satisfied (R). ∎

Related to Corollary 2.20 is the recent result of E. Hemaspaandra et al. [HNOS93] that $\mathrm{E} \neq \mathrm{UE}$ implies there is a tally set T in $\mathrm{R}^p_{tt}(\text{P-sel}) \bigcap (\mathrm{UP} - \mathrm{P})$, where UP is Valiant's [Val76] notion of unambiguous polynomial time, and, analogously, UE is the class of sets computable in unambiguous time $2^{\mathcal{O}(n)}$.

3 Nondeterministic Selectivity

Though all sets in P are trivially P-selective, Selman [Sel79] proved that not all NP sets are P-selective unless P = NP, and that not all sets in NP $\bigcap$ coNP are P-selective unless P = NP $\bigcap$ coNP. Motivated by the limitations of P-selectivity in capturing NP $\bigcap$ coNP, NP, and other classes, and also motivated by the centrality of nondeterministic computation in complexity theory, we introduce the notion of NP-selectivity. Later in this section, we introduce selectivity notions that are more inclusive still: #P-selectivity, OptP-selectivity, SpanP-selectivity, and selectivity versions of three types of NP functions more general than the type defined below.

Definition 3.1 [BLS85] NPSV_t is the class of total (single-valued) functions $f : \Sigma^* \longrightarrow \Sigma^*$ such that there is a nondeterministic polynomial-time Turing machine N such that, for every input $x \in \Sigma^*$:

1. $N(x)$ has at least one accepting computation path, and
2. every accepting computation path of $N(x)$ outputs value $f(x)$.

We say that $A \leq_m^{NPSV_t} B$ if there is an NPSV_t function f such that for every $x \in \Sigma^*$ it holds that $x \in A \iff f(x) \in B$ (see [Ric89]). It will be clear from the definition below of NP-selectivity that if $A \leq_m^{NPSV_t} B$ and B is NP-selective, then A is also NP-selective.

Definition 1.1 defined the P-selective sets. We now propose a natural extension of that definition to selectivity via any class of functions. At first, one might be tempted to use as one's definition something like the following.

> Let $\mathcal{FC}$ be any class of total functions mapping from Σ^* to Σ^*. A set A is $\mathcal{FC}$-selective if there is a function $f \in \mathcal{FC}$ so that, for every $x, y \in \Sigma^*$,
>
> 1. $f(x, y) \in \{x, y\}$, and

2. if $x \in A$ or $y \in A$, then $f(x, y) \in A$.

However, the above definition assumes the functions to be single-valued and total. Instead, we propose below a more flexible definition. Definition 3.3 has the nice property that for single-valued, total classes of selector functions, such as is the case for both the P-selective sets and the NP-selective sets, Definition 3.3 coincides exactly with the definition above and also with Definition 1.1.

Definition 3.2 (see, e.g., [Sel]) For every (possibly partial, possibly multivalued) function $f : \Sigma^* \longrightarrow \Sigma^*$, let $\mathit{set}\text{-}f(x) = \{z \mid z \text{ is an output of } f(x)\}$.

Definition 3.3 [Selectivity by Classes of Functions]

1. Let $\mathcal{FC}$ be any class of (possibly multivalued, possibly partial) functions mapping from Σ^* to Σ^*. A set A is $\mathcal{FC}$-selective if there is a function $f \in \mathcal{FC}$ so that, for every $x, y \in \Sigma^*$,

   (a) $\mathit{set}\text{-}f(x, y) \subseteq \{x, y\}$, and

   (b) if $x \in A$ or $y \in A$, then $\emptyset \neq \mathit{set}\text{-}f(x, y) \subseteq A$.

2. Let $\mathcal{FC}$ be any class of functions mapping from Σ^* to Σ^*. We define $\mathcal{FC}\text{-sel} = \{A \mid A$ is $\mathcal{FC}$-selective$\}$.

We will use NP-sel to denote the class of NPSV_t-selective sets. We will call the functions f *selector* functions.

We immediately note that NP-selectivity is a more inclusive notion than P-selectivity exactly if $\mathrm{P} \neq \mathrm{NP} \bigcap \mathrm{coNP}$.

Theorem 3.4 P-sel $\neq$ NP-sel if and only if $\mathrm{P} \neq \mathrm{NP} \bigcap \mathrm{coNP}$.

Proof If $\mathrm{P} = \mathrm{NP} \bigcap \mathrm{coNP}$ then all NPSV_t functions are computable in polynomial time ([BLS85], or see Part 2 of Lemma 3.5), and thus P-sel = NP-sel. If $\mathrm{P} \neq \mathrm{NP} \bigcap \mathrm{coNP}$, then there is a set B in $\mathrm{NP} \bigcap \mathrm{coNP} - \mathrm{P}$ such that B is many-one reducible to $\overline{B}$ [Sel79, Theorem 1], and thus B cannot be P-selective [Sel79, Theorem 2]. However, we'll soon observe (Proposition 3.11) that all sets in $\mathrm{NP} \bigcap \mathrm{coNP}$ are NP-selective. ∎

We note the following easy observations, whose final three parts essentially reflect the effects of strong (in the sense of Long [Lon82] and Selman [Sel78]) computation; informally put, strong computation is computation such that at least one nondeterministic path gets a correct answer and no paths get wrong answers. More concretely, the second part holds as the ith bit of an NPSV_t function can easily be computed in $\mathrm{NP} \bigcap \mathrm{coNP}$, and the final two parts hold by direct simulation. The first part holds by transforming a non-symmetric selector f' into the symmetric selector $f(a,b) = f'(\min(a,b), \max(a,b))$. As a notational convenience, we will use FP, rather than the more rigorous FP_t, to denote the total functions computable in polynomial time.

Lemma 3.5

1. If L is NP-selective, then it has some NPSV_t selector function that satisfies the condition $(\forall x,\, y \in \Sigma^*)[f(x,y) = f(y,x)]$.

2. $\mathrm{NPSV}_t = \mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}}$.

3. $\mathrm{NP} = \mathrm{NP}^{\mathrm{NPSV}_t}$.[2]

4. $\mathrm{NPSV}_t = (\mathrm{NPSV}_t)^{\mathrm{NPSV}_t}$.

However, we will see in this section that one can go a bit beyond this lemma. Taking as an example the third part of the lemma, we see that an NP machine accessing some function *that is single-valued and total on every input* accepts indeed only an NP language. But note that an NP machine accessing some multivalued partial NP function such that *every query actually asked in the actual query tree happens to be one on which the multivalued NP function is both defined and single-valued* also accepts only an NP language. Access of this sort is known as "guarded" or "smart" access; it was introduced by Grollmann and Selman [GS88] and has been recently studied in detail by Cai, Hemachandra, and Vyskoč [CHV]. So, informally, what the following refined version of the last three parts of the lemma states is that not only does Turing access to strong computation save a quantifier, but so also does even guarded access to strong computation. Below, the "guarded" subscripts indicate the querying machine ensures that every question actually asked to the oracle maintains the stated "guarded" property. This notion of guardedness encapsulates the flavor of the approach that will be central in the proofs of Theorem 3.28 and Theorem 3.32 (see also the result stated as Comment 3.40).

Before stating the refined lemma, we include definitions of multivalued partial NP functions (used in the lemma), and of two other function types that will play a role later in this paper: single-valued partial NP functions and multivalued total NP functions. Recall that

[2]We use the natural notion of access to a single-valued function oracle; the value of the function on the queried string is returned.

$set\text{-}f(x) = \{z \mid z \text{ is an output of } f(x)\}$.

Definition 3.6 [BLS85]

1. NPMV is the class of partial multivalued functions $f : \Sigma^* \longrightarrow \Sigma^*$ such that there is a nondeterministic polynomial-time Turing machine N such that, for every input $x \in \Sigma^*$, all accepting paths have output and:
   (a) $f(x)$ is defined if and only if $N(x)$ has at least one accepting computation path, and
   (b) $set\text{-}f(x) = \{z \mid$ some accepting computation path of $N(x)$ outputs $z\}$.
2. NPSV is the class of all NPMV functions that are single-valued (that is, functions $f \in$ NPMV such that, for every x, it holds that $||\, set\text{-}f(x)\, || \leq 1$).
3. NPMV_t is the class of all NPMV functions that are total (that is, functions $f \in$ NPMV such that, for every x, it holds that $set\text{-}f(x) \neq \emptyset$).

Lemma 3.7

1. $\mathrm{NPSV}_t = \mathrm{FP}^{\mathrm{NP}}_{\text{guardedly strong}}$.
2. $\mathrm{NP} = \mathrm{NP}^{\mathrm{NPMV}}_{\text{guardedly total, guardedly single-valued}}$.
3. $\mathrm{NPSV}_t = (\mathrm{NPSV}_t)^{\mathrm{NPMV}}_{\text{guardedly total, guardedly single-valued}}$.

We will use the γ *reductions* of Adleman and Manders, which are the same as *many-one strong nondeterministic reductions* [AM77,Lon82]. We say that $A \leq_\gamma B$ if there is a nondeterministic Turing machine N (which on each accepting path p outputs some value, call it $output(x,p)$) such that (i) for each x, $N(x)$ has at least one accepting path, and (ii) for each x, and each accepting path p of $N(x)$, it holds that

$$x \in A \iff output(x,p) \in B.$$

Note that if $A \leq_m^{NPSV_t} B$ then $A \leq_\gamma B$.

The following results address the inclusiveness of NP-sel with respect to NP $\bigcap$ coNP and NP: all NP $\bigcap$ coNP sets are NP-selective, but it is unlikely that any NP-complete sets are. In fact, one can go a bit beyond completeness, and build a more general theory of NP-selectivity, loosely analogous to the theory that has been built for P-selectivity (see especially [Sel79,Sel82b]), and we do so in the following discussion.

Crucial will be the following theorem. Recall that $L_1 \oplus L_2 = \{0x \mid x \in L_1\} \bigcup \{1y \mid y \in L_2\}$.

Theorem 3.8 If a set A is polynomial-time Turing self-reducible and is $\leq_\gamma$-reducible to $\overline{S} \oplus S$, for some NPMV_t-selective set S, then A is in NP $\bigcap$ coNP.

Proof Let A be polynomial-time Turing self-reducible via machine M, let A be $\leq_\gamma$-reducible to $\overline{S} \oplus S$ via a nondeterministic machine N, and let S be NPMV_t-selective via a nondeterministic machine F. Let x be a string whose membership in A we are testing. Suppose that N on x outputs cu for some computation path so that $x \in A$ if and only if $\chi_S(u) = c$. Let us fix such c and u. For any v and w, let us write $v <_F w$ if $F(v, w)$ outputs w; that is, it is witnessed by F that $v \in S \to w \in S$. By convention, let $\perp$ and $\top$ be strings such that $\perp <_F v$ for any v, and $v <_F \top$ for any v.

Consider a simulation of M on x defined as follows: The simulation will use two strings, a and b. Initially, a is set to $\top$ and b is set to $\perp$. When M makes its i-th query y_i, instead of making query, do the following:

1. Simulate N on y_i to compute $d_i v_i$ such that $y_i \in A \iff \chi_S(v_i) = d_i$.
2. If $a <_F v_i$, then choose the branch corresponding to $v_i \in S$.

   If $v_i <_F b$, then choose the branch corresponding to $v_i \notin S$.

   If $b <_F v_i <_F a$, then simulate F on (u, v_i). If $u <_F v_i$, then set a to v_i and choose the branch corresponding to $v_i \in S$. If $v_i <_F u$, then set b to v_i and choose the branch corresponding to $v_i \notin S$.

Let r denote 1 if M accepts in the simulation and 0 if M rejects in the simulation. Let a_0 and b_0 be the final values of a and b, respectively. Let i and j be such that v_i is set to a_0 and v_j is set to b_0. The following properties hold:

1. $b_0 <_F u <_F a_0$.
2. If $b_0 \notin S$ and $a_0 \in S$, then M^A on x accepts if and only if $r = 1$, so $\chi_A(x) = r$.
3. If $b_0 \in S$, then $u \in S$, so $\chi_A(x) = c$.
4. If $a_0 \notin S$, then $u \notin S$, so $\chi_A(x) = 1 - c$.

Suppose $r = c$. Then we have $\chi_A(x) = r$ if and only if $a_0 \in S$. So, $\chi_A(x) = r$ if and only if $v_i \in S$ if and only if $\chi_A(y_i) = d_i$. If this case holds, let $z = y_i$ and $e = 1$ if $r = d_i$ and 0 otherwise. Suppose $r = 1 - c$. Then we have $\chi_A(x) = r$ if and only if $b_0 \notin S$. So, $\chi_A(x) = r$ if and only if $v_i \notin S$ if and only if $\chi_A(y_j) = 1 - d_j$. If this case holds, let $z = y_j$ and $e = 0$ if $r = d_j$ and 0 otherwise. It holds that $x \in A$ if and only if $\chi_A(z) = e$.

Thus, by the above simulation, we will find z and e such that $\chi_A(x) = 1$ if and only if $\chi_A(z) = e$. It is not hard to see that (i) there is at least one computation path that makes us find such z and e, (ii) the simulation runs in time polynomial in $|x|$, and (iii) z is a string appearing in the self-reducing tree of M on x. By repeating the above simulation polynomially many times, we eventually finds a string z' and e' such that $x \in A$ if and only if $\chi_A(z') = e'$, and M on x determines the membership of z' in polynomial time without making any queries. Clearly, $x \in A$ if either $e' = 1$ and M on z' accepts or $e' = 0$ and M on z' rejects, and $x \notin A$ if either $e' = 0$ and M on z' accepts or $e' = 1$ and M on z' rejects. Thus, we have nondeterministic polynomial-time procedures for both membership in A and non-membership in A. ∎

Theorem 3.8 should be viewed as a nondeterministic analog of the result of Buhrman, van Helden, and Torenvliet [BvHT93] that "for all sets A, if A is P-selective and polynomial-time Turing self-reducible, then $A \in$ P." However, though mirroring closely the structure and proof of their result, Theorem 3.8 is sufficiently general that it does not seem to be provable via relativization of Buhrman, van Helden, and Torenvliet's result, though a corollary of our theorem, Corollary 3.10, does have a proof via relativization.[3] Note that $\mathrm{R}_\gamma(\mathrm{R}^p_{1\text{-}tt}(\text{NP-sel}))$ merely denotes the class of all sets that γ-reduce to some set that itself 1-truth-table reduces to some NP-selective set.

Corollary 3.10 If A is polynomial-time Turing self-reducible and NP-selective (or even in $\mathrm{R}_\gamma(\mathrm{R}^p_{1\text{-}tt}(\text{NP-sel}))$), then $A \in \mathrm{NP} \bigcap \mathrm{coNP}$.

The partial converse is clear.

Proposition 3.11 $\mathrm{NP} \bigcap \mathrm{coNP} \subseteq \text{NP-sel}$.

From the above corollaries and proposition, one may draw conclusions above the likelihood that classes having Turing self-reducible complete sets, such as NP, PP, and PSPACE, are NP-selective. For example:

Corollary 3.12

1. $\mathrm{NP} \subseteq \mathrm{R}_\gamma(\mathrm{R}^p_{1\text{-}tt}(\text{NP-sel}))$ if and only if NP = coNP.
2. If PSPACE (respectively, $\oplus$P, PP) is contained in $\mathrm{R}_\gamma(\mathrm{R}^p_{1\text{-}tt}(\text{NP-sel}))$ (and, in particular, if it is contained in NP-sel), then $\mathrm{NP} \bigcap \mathrm{coNP} = \mathrm{PSPACE}$ (respective, $\mathrm{NP} \bigcap \mathrm{coNP} \supseteq \oplus\mathrm{P}$, $\mathrm{NP} \bigcap \mathrm{coNP} = \mathrm{PP}$).

[3] The proof via relativization of the non-parenthetical part of Corollary 3.10 is as follows. A relativized form of the Buhrman, van Helden, and Torenvliet result is:

Fact 3.9 (Relativization of a result of [BvHT93]) For all sets L and A, if A is FP^L-selective and polynomial-time Turing self-reducible, then $A \in \mathrm{P}^L$.

Let A be NP-selective via NPSV_t selector function f. By Lemma 3.5, Part 2, $f \in \mathrm{FP}^L$ for some $L \in \mathrm{NP} \bigcap \mathrm{coNP}$. So by Fact 3.9, $A \in \mathrm{P}^{\mathrm{NP} \cap \mathrm{coNP}} = \mathrm{NP} \bigcap \mathrm{coNP}$. The 1-tt part of the parenthetical remark (but not, as far as we know, the γ part of the parenthetical remark) can also be obtained by relativizing the related 1-tt result of Buhrman, van Helden, and Torenvliet [BvHT93].

Simplifying our result to its quintessential form, we have the following.

Corollary 3.13 There is an NP-complete set that is NP-sel if and only if NP = coNP. Equivalently, NP $\subseteq$ NP-sel if and only if NP = coNP.

The above results may be interpreted as saying that NP is unlikely to have NP-selective complete sets, or even NP-selective sets hard with respect to such powerful reductions as, for example, γ reductions, unless NP = coNP.

We may also note the following result.

Theorem 3.14 $(\forall A \neq \Sigma^*, A \neq \emptyset)[(A \in \mathrm{NP} \bigcap \mathrm{coNP}) \iff (\overline{A} \leq_m^{NPSV_t} A$ and $A \in$ NP-sel$) \iff (\overline{A} \leq_\gamma A$ and $A \in$ NP-sel$) \iff (\overline{A} \leq_{pos}^p A$ and $A \in$ NP-sel$)]$.

Proof The claim about positive reductions is analogous to [Sel82b, Theorem 8]. The other parts are easy to see. ∎

Related to the positive reduction claim in the previous theorem, we have the following claim.

Theorem 3.15 If $A \leq_{pos}^p B$ and B is NP-selective, then

1. A is NP-selective, and
2. if $B \neq \Sigma^*$ and $B \neq \emptyset$ then $A \leq_m^{NPSV_t} B$.

Lemma 3.16 (Relativization of a result of [BTvEB93]) If $A \leq_{pos}^p B$, B is FP^L-selective for some set L, $B \neq \emptyset$, and $B \neq \Sigma^*$, then $A \leq_m^{p,L} B$, and hence A is FP^L-selective.

Proof of Theorem 3.15 Let B be NP-selective with selector $f \in \mathrm{NPSV}_t$. There exists a language $L \in \mathrm{NP} \bigcap \mathrm{coNP}$ such that B is FP^L-selective. Thus by Lemma 3.16, $A \leq_m^{p,L} B$. Since $L \in \mathrm{NP} \bigcap \mathrm{coNP}$, by Lemma 3.5, Part 2, it follows that $A \leq_m^{NPSV_t} B$, and that A is NP-selective. ∎

Corollary 3.17 If NE $\neq$ coNE, then there is a set A in NP $-$ (NP $\bigcap$ coNP) that is not γ-reducible to its complement (furthermore, A is even P-selective), where E = $\bigcup_{c>0} \mathrm{DTIME}[2^{cn}]$, and NE = $\bigcup_{c>0} \mathrm{NTIME}[2^{cn}]$

Note that as an immediate consequence of Theorem 3.15 and Corollary 3.13, NP-selective sets cannot be positive-Turing-hard for NP unless NP = coNP.[4] This has a natural analog,

[4] Can we conclude NP = coNP from the weaker assumption that NP has *Turing*-hard NP-selective sets? This remains open. However, a relativized counterexample is implicitly in the literature. Cai et al. [CGH$^+$89] proved that the existence of sparse Turing-hard (which is equivalent to tally Turing-hard, as sparse sets

Theorem 3.21, in the realm of "promise problems ([EY80,ESY84], see also [GS88,CHV])." Informally, PP-A [Sel88,LS93] is the promise problem corresponding to P-selectivity; a solution to PP-A will—if the promise is met that exactly one of x and y is in A—contain $\langle x, y\rangle$ exactly if $x \in A$.

Definition 3.18 ([LS93], see also [Sel88])

1. Given any set A, we say that a set B *is a solution to* PP-A if $\{\langle x, y\rangle \mid (x \in A) \oplus (y \in A)\} \bigcap B = \{\langle x, y\rangle \mid (x \in A) \text{ and } (y \notin A)\}$.
2. For any sets C and D, we say PP-C *is Turing hard for* D if for every solution L of PP-C it holds that $D \leq_T^p L$. For any class $\mathcal{E}$, we say PP-F is Turing-hard for $\mathcal{E}$ if PP-F is Turing-hard for each set in $\mathcal{E}$.

Just as A is P-selective if and only if PP-A has a solution in P [LS93], we have the following (note that NP $\bigcap$ coNP = $\mathrm{P}^{\mathrm{NP}\cap\mathrm{coNP}}$ [Sel74,Sel79]).

Lemma 3.19 For each set A, A is NP-selective if and only if PP-A has a solution in NP $\bigcap$ coNP.

Lemma 3.20

1. Let $A \leq_{pos}^p B$ and let L be any solution of PP-B. Then there is a function g computable by a polynomial-time machine with oracle L such that $x \in A \iff g(x) \in B$.
2. If A is polynomial-time Turing self-reducible and $A \leq_{pos}^p B$, then PP-B is Turing-hard for A.

Proof Let L be a solution for PP-B. Define $f_L(x, y) = x$ if $\langle x, y\rangle \in L$ and $f_L(x, y) = y$ otherwise. f_L is a selector for B, and f_L is computable in polynomial time relative to L.

Since $A \leq_{pos}^p B$ and B is FP^L-selective, by Lemma 3.16, $A \leq_m^{p,L} B$. Part 1 now follows immediately since this implies that there exists $g \in \mathrm{FP}^L$ such that $x \in A \iff g(x) \in B$.

To prove Part 2, observe that A is FP^L-selective. By hypothesis, A is Turing self-reducible, and thus, by Buhrman et al.'s Fact 3.9, $A \in \mathrm{P}^L$. Thus, $A \leq_T^p L$, which proves the lemma. ∎

Turing reduce to tally sets) sets for NP does not robustly (i.e., in every relativized world) collapse the "Boolean hierarchy [CGH+88]" and Kadin [Kad89] established the apparently incomparable result that the same hypothesis does not robustly imply the collapse of the polynomial hierarchy to P with (constructibly) less than $\mathcal{O}(\log n)$ queries to NP. The standard relationships between P-selective sets and tally sets imply that in the oracle worlds constructed by Cai et al. and Kadin, respectively, the existence of NP Turing-hard P-selective (and thus NP-selective) sets does not, respectively, collapse the Boolean hierarchy or collapse the polynomial hierarchy to P with (constructibly) less than $\mathcal{O}(\log n)$ queries to NP.

The following theorem follows immediately from Part 2 of Lemma 3.20.

Theorem 3.21 If A is $\leq_{pos}^{p}$-hard for NP, then PP-A is Turing-hard for NP.

The above theorem should be compared with the techniques of [Sel88], which "apply only when $A \in$ NP and, at that, only when A is $\leq_{d}^{p}$-equivalent to a disjunctive self-reducible set" [LS93]. Those techniques suffice to prove (see [Sel88]) that "If A is $\leq_{d}^{p}$-complete for NP, then PP-A is NP Turing-hard," but note that Theorem 3.21 goes beyond the limitations of disjunctive reductions.

The P-selective sets were shown by Ko [Ko83] to be in P/poly, the class of sets having small circuits, and the P-selective sets in NP were shown by Ko and Schöning [KS85] to fall into the second level of the low hierarchy [Sch83] (that is, $A \in \mathrm{NP} \bigcap \text{P-sel} \Rightarrow \mathrm{NP}^{\mathrm{NP}^{A}} = \mathrm{NP}^{\mathrm{NP}}$). Do similar results hold for NP-selectivity? The NP-selective sets in fact have the same lowness properties as the P-selective sets (Theorem 3.25), and have nonuniform representations.

Lowness and extended lowness are used here as defined, respectively, by Schöning [Sch83], and by Long and Sheu [LS91] and Balcázar, Book, and Schöning [BBS86].

Definition 3.22

1. **[Sch83]** For each $k \geq 1$, define $\mathrm{Low}_k = \{L \in \mathrm{NP} \mid \Sigma_k^{p,L} = \Sigma_k^p\}$, where the Σ_k^p are the Σ levels of the polynomial hierarchy [MS72,Sto77].
2. **[LS91,BBS86]** For each $k \geq 2$, define $\mathrm{ExtendedLow}_k = \{L \mid \Sigma_k^{p,L} = \Sigma_{k-1}^{p,\mathrm{SAT}\oplus L}\}$. For each $k \geq 3$, define $\mathrm{ExtendedLow}\Theta_k = \{L \mid \mathrm{P}^{(\Sigma_{k-1}^{p,L})[\mathcal{O}(\log n)]} \subseteq \mathrm{P}^{(\Sigma_{k-2}^{p,\mathrm{SAT}\oplus L})[\mathcal{O}(\log n)]}\}$.

Theorem 3.23 NP-sel $\subseteq$ (NP $\bigcap$ coNP)/poly.

Lemma 3.24 ([LS93], see also [Ko83]) For all sets A and L, if A is FP^L-selective, then $A \in \mathrm{P}^L$/poly.

Proof of Theorem 3.23 Follows immediately from Lemma 3.24 and Part 2 of Lemma 3.5. ∎

We stress that Theorem 3.23 should not be confused with the potentially weaker (though also correct) claim that NP-sel $\subseteq$ (NP/poly) $\bigcap$(coNP/poly). (The reason this is potentially weaker is that (NP/poly) $\bigcap$(coNP/poly) $=$ $(\mathrm{NP} \bigcap \mathrm{coNP})^{\mathrm{SPARSE}} \supseteq$ (NP $\bigcap$ coNP)/poly [GB91].)

From Theorem 3.23, it immediately follows that the NP-selective sets in NP are Low_3 (since (NP $\bigcap$ coNP)/poly $\bigcap$ NP $\subseteq$ (NP/poly) $\bigcap$(coNP/poly) $\bigcap$ NP $\subseteq$ (coNP/poly) $\bigcap$ NP, which due to Kämper [Käm91] is Low_3). Kämper [Käm91] proves also that all disjunctive self-reducible sets in (NP $\bigcap$ coNP)/poly are Low_2, but this does not yield a Low_2 result for the NP-selective sets as not even all P-selective sets are disjunctive (or even Turing) self-reducible (this follows immediately from the facts that all Turing-self-reducible P-selective sets are in P [BvHT93], yet there are P-selective sets of arbitrary complexity [Sel79,Sel82a]). However, we can prove that the NP-selective sets are even lower. Indeed, they are as low as the P-selective sets. The following theorem follows immediately from a more general result we will establish later, Corollary 3.33.

Theorem 3.25 The NP-selective sets in NP are Low_2.

The above result is optimal with respect to relativizable proof techniques, as Allender and Hemachandra [AH92] have proven that there is a relativized world in which even some P-selective NP sets are not lower than Low_2.

A set A is said to be $\mathrm{GeneralizedLow}_2$ if $\Sigma_2^{p,A} = \Sigma_2^p$ [BBS86]. It is not hard to see—by Lemma 3.5, the standard equality $\mathrm{NP} = \mathrm{NP}^{\mathrm{NP} \bigcap \mathrm{coNP}}$ [Sch83], and relativizing the result of Longpré and Selman [LS93] that "If A is Turing self-reducible and A is Turing reducible to a P-selective set then A is $\mathrm{GeneralizedLow}_2$"—that we obtain the following generalized lowness result.

Theorem 3.26 If A is Turing self-reducible and $A \in \mathrm{R}_T^p(\text{NP-sel})$, then A is $\mathrm{GeneralizedLow}_2$.

As to extended lowness, Köbler [Köb93] has shown that that (NP $\bigcap$ coNP)/poly is $\mathrm{ExtendedLow}\Theta_3$. From this and Theorem 3.23, we can immediately conclude that the NP-selective sets are $\mathrm{ExtendedLow}\Theta_3$.

An interesting question left open by Köbler's paper is whether (NP/poly) $\bigcap$ (coNP/poly) is extended low. We resolve this issue by showing that it is. It is an interesting open issue whether our result can itself be strengthened via the techniques of Gavaldà and Köbler [Gav92a,Köb93] to an $\mathrm{ExtendedLow}\Theta_3$ result; we conjecture that it can.

Theorem 3.27 (NP/poly) $\bigcap$ (coNP/poly) is $\mathrm{ExtendedLow}_3$.

Proof Let $H \in$ (NP/poly) $\bigcap$ (coNP/poly). Let $B \in \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^H}}$ (let's say, for convenience, $B = L(N_1^{L(N_2^{L(N_3^H)})})$). We will show that $B \in \mathrm{NP}^{\mathrm{NP}^{\mathrm{SAT} \oplus H}}$.

Let S_1 (S_2) be an NP (coNP) set certifying $H \in$ NP/poly ($H \in$ coNP/poly). Let $p(\cdot)$ be a polynomial bounding the size of the correct advice sequences for each. Let $q(\cdot)$ be a polynomial composing the polynomial running times of N_1, N_2, and N_3.

Let $\langle \ldots \rangle$ be any nice, "multi-arity onto" pairing function (that is, the ranges of different arities are disjoint, and the union over all arities covers Σ^*, see, e.g., [HHT93]). On input x, our base NP machine of our $\mathrm{NP}^{\mathrm{NP}^{\mathrm{SAT}\oplus H}}$ machine guesses nondeterministically strings r_1, …, $r_{q(|x|)}$, and s_1, …, $s_{q(|x|)}$, satisfying, for each i, $|r_i| \leq p(i)$ and $|s_i| \leq p(i)$. Via a single call to $\mathrm{NP}^{\mathrm{SAT}\oplus H}$, the base machine checks whether $r_1, \ldots, r_{q(|x|)}$ is a good advice set for helping S_1. In particular, we make one query, $\langle x, r_1, \ldots, r_{q(|x|)}\rangle$, to the $\mathrm{NP}^{\mathrm{SAT}\oplus H}$ set:

$$E' = \{\langle x, r_1, \cdots, r_z\rangle \mid z = q(|x|) \text{ and}$$

$$(\forall i : 1 \leq i \leq z)\,[|r_i| \leq p(i)] \text{ and } (\exists y : |y| \leq q(|x|))\,[y \in H \iff \langle y, r_{|y|}\rangle \notin S_1]\},$$

and if the answer is "no," we know the "r" advice collection is good. Similarly, with one question to an $\mathrm{NP}^{\mathrm{SAT}\oplus H}$ set E'' (defined analogously), we determine whether the "s" advice collection is good for helping S_2.

Note that *when given the correct advice strings*, an NP machine can strongly (in the sense of Long [Lon82] and Selman [Sel78]) check whether $x \in H$ or $x \notin H$, by nondeterministically guessing which is true and checking an $x \in H$ guess via checking whether $\langle x, r_{|x|}\rangle \in S_1$, and checking an $x \notin H$ guess via checking whether $\langle x, s_{|x|}\rangle \in S_2$.

Our simulation of $B = L(N_1^{L(N_2^{L(N_3^H)})})$ in $\mathrm{NP}^{\mathrm{NP}^{\mathrm{SAT}\oplus H}}$ proceeds as follows (for simplicity, let's call our base machine N_4). N_4 guesses and checks good advice sets as already described. N_4 now simulates N_1, except each time N_1 asks a query y to $L(N_2^{L(N_3^H)})$, N_4 asks the query $\langle y, \langle r_1, \ldots, r_{q(|x|)}\rangle, \langle s_1, \ldots, s_{q(|x|)}\rangle\rangle$ to an $\mathrm{NP}^{\mathrm{SAT}\oplus H}$ set E''', which itself will satisfy $E''' = L(N_5^{\mathrm{SAT}\oplus H})$ for a machine N_5 to be defined. (Since we only have one $\mathrm{NP}^{\mathrm{SAT}\oplus H}$ oracle, the actual set we will use is $E = E' \oplus E'' \oplus E'''$.) N_5 on input $\langle y, \langle r_1, \ldots, r_t\rangle, \langle s_1, \ldots, s_t\rangle\rangle$ simulates N_2 on input y, except every time $L(N_2)$ asks a query z to $L(N_3)$ on input y, N_5 asks the query $\langle z, \langle r_1, \ldots, r_t\rangle, \langle s_1, \ldots, s_t\rangle\rangle$ to the NP set G (since SAT is NP-complete, we implicitly convert the query to an appropriate query to SAT):

> $G = \{\langle z, \langle r_1, \ldots, r_t\rangle, \langle s_1, \ldots, s_t\rangle\rangle \mid$ if we simulate $N_3^H(z)$, replacing each call to H (say "$w \in H$?") by nondeterministically checking whether $\langle w, r_{|w|}\rangle \in S_1$ (in which case we proceed along the path certifying $\langle w, r_{|x|}\rangle$ as of $w \in H$) and (separately, nondeterministically) whether $\langle w, s_{|w|}\rangle \notin S_2$ (in which case we proceed as if $w \notin H$), we have an accepting path of our simulated $N_3\}$. Note: if any of the w have $|w| > t$, we act as if $s_{|w|} = r_{|w|} = \epsilon$, as in actual runs this case will not occur.

We make no claim that $G \in \mathrm{NP} \bigcap \mathrm{coNP}$. In fact, with "bad" advice as inputs, the simulation defining G will be quite chaotic: a query "$w \in H$?" might be treated as being answered both "yes" and "no," or neither "yes" nor "no." However, *when given good advice sets*, the machine will in fact correctly simulate $N_3^H(z)$: each query w of $N_3(z)$ will be answered either "yes" or "no," will not be answered both "yes" and "no," and will be answered correctly. That is, G's simulation of H is, *when the advice is correct*, an example of strong computation. Crucially, for every query actually asked of G during an actual run of our $\mathrm{NP}^{E' \oplus E'' \oplus L(N_5^G)}$ algorithm, the advice will be correct (and thus the strong computation going on within G will be correct). Recall that this behavior, in which every *actual access* to an oracle maintains a certain nice property of the oracle computation (such as computing strongly), though some queries that are never asked might taint the property, is known as "guarded" access. We've now given an $\mathrm{NP}^{\mathrm{NP}^{\mathrm{SAT} \oplus H}}$ simulation of an arbitrary set $B \in \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^H}}$, for arbitrary $H \in (\mathrm{NP/poly}) \bigcap (\mathrm{coNP/poly})$. ∎

The theorems earlier in this section indicate that even NP-selectivity—whose selector functions, recall, are NPSV_t functions—is unlikely to capture all of NP. We now discuss the selection power possessed by other function classes of complexity theory. We commend a more detailed investigation of these functions to the reader as an open issue, and, as a start in this research direction, we note below some initial results.

In particular, the classes that seem to us to be of the most interest are the NPSV-selective sets, the NPMV-selective sets, and the NPMV_t-selective sets. What results hold for these classes? NPMV-sel (and thus NPSV-sel) is contained in NP/poly, and NPMV_t-sel is contained in NP/poly $\bigcap$ coNP/poly, via using a divide-and-conquer approach to find an appropriate advice set (see the proofs of Theorem 3.28 and Theorem 3.32). It follows immediately (respectively via Theorem 3.27 and [Yap83]) that NPMV_t-sel is $\mathrm{ExtendedLow}_3$, and that NPMV_t-sel $\bigcap$ NP is Low_3. We now state some less immediate results.

Theorem 3.28 NPSV-sel $\bigcap$ NP $\subseteq$ (NP $\bigcap$ coNP)/poly.

Proof Let $L \in \mathrm{NP}$ be NPSV-selective, with selector function $f \in \mathrm{NPSV}$. For the same reason as the proof of Part 1 of Lemma 3.5, without loss of generality assume f satisfies $(\forall x, y)[\textit{set-}f(x,y) = \textit{set-}f(y,x)]$. Recall that $\textit{set-}f(x,y) = \{y \mid y \text{ is a value of } f(x)\}$. We refer to *set-f* here, and later, even though f is single-valued, to avoid ambiguity about what happens if the function is undefined.

For each n, there is a (possibly empty) set B_n so that:

1. $B_n \subseteq L^{=n}$,
2. B_n has at most n elements, and
3. $(\forall x \in \Sigma^n)\,[x \in L \iff (\exists y \in B_n)\,[\textit{set-}f(x,y) = \{x\}]]$.

This is true, as in the case of Ko's proof that P-sel $\subseteq$ P/poly [Ko83], as we may repeatedly choose to add to B_n some element of $L^{=n}$ that loses to (that is, proves less likely to be in L according to f) at least half the elements that both are not yet in B_n and don't yet beat some element in B_n.

Let

> $L' = \{\langle x, \langle \alpha_1, \alpha_2, \ldots, \alpha_z, wit_1, wit_2, \ldots, wit_z\rangle\rangle \mid z \leq |x|$ and $(\forall i \in \{1, 2, \cdots, z\})[wit_i$ is a witness certifying (with respect to some fixed polynomial-time binary relation defining NP set L) that $\alpha_i \in L]$ and $(\exists i \in \{1, 2, \ldots, z\})[\textit{set-}f(x, \alpha_i) = \{x\}]\}$.

Note that $L' \in \mathrm{NP} \bigcap \mathrm{coNP}$. The containment in NP is immediate. The containment in coNP follows from the fact that, since the α_i are certifiably in L, by the definition of NPSV-selectivity, the partial function f is guaranteed to be defined when it is invoked in the actual run of the NP machine for $\overline{L'}$; informally, one might say that f is functioning in a guardedly total manner, in this setting. Furthermore, the set $\{x \mid \{\langle x, \langle \alpha_1, \alpha_2, \ldots, \alpha_z, wit_1, wit_2, \ldots, wit_z\rangle\rangle \in L'\}$, is exactly L, where $B_n = \{\alpha_1, \cdots, \alpha_z\}$, and the *wit* strings are witnesses of the membership of the elements of B_n in L. This proves that $L \in (\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly}$. ∎

It is not hard to see that essentially the same proof as that of the above theorem also establishes the following result.

Theorem 3.29 NPSV-sel $\subseteq$ NP/poly $\bigcap$ coNP/poly.

From Theorem 3.27 and the above results, we may conclude the following.

Corollary 3.30 The NPSV-selective sets are ExtendedLow$_3$.

What can be said about the lowness of the NPSV-selective sets in NP? Theorem 3.29 and Köbler's "(NP $\bigcap$ coNP)/poly $\bigcap$ NP is LowΘ_3" result imply a LowΘ_3 result. However, below, we establish that the NPSV-selective sets in NP are in fact Low$_2$.

Lemma 3.31 [LS93] If A is in Σ_i^p and B is a solution of PP-A, then $\Sigma_{i+1}^{p,A} \subseteq \Sigma_{i+1}^{p,B}$.

Theorem 3.32 If $A \in$ NPSV-sel $\bigcap$ NP, then PP-A has a solution L that is Low$_2$.

Corollary 3.33 NPSV-sel $\bigcap$ NP $\subseteq$ Low$_2$.

Proof of Corollary 3.33 Follows immediately from Theorem 3.32 via Lemma 3.31. ∎

Proof of Theorem 3.32 Let $A \in$ NPSV-sel $\bigcap$ NP, with selector function $f \in$ NPSV. Recall we use the notation *set-f*, even though f is single-valued, to avoid ambiguity about the "f is undefined" case. Define $L_1 = \{\langle x,y\rangle \,|\, set\text{-}f(x,y) = \{x\} \text{ and } x \in A\}$ and $L_2 = \{\langle x,y\rangle \,|\, x \neq y \text{ and } set\text{-}f(x,y) = \{y\}\,\}$. Note that L_1 and L_2 are in NP. Since L_1 is a solution of PP-A, it suffices to show that $\Sigma_2^{p,L_1} \subseteq \Sigma_2^p$. Let $C \in \Sigma_2^{p,L_1}$. There exist NP machines N_1 and N_2 such that $C = L(N_1^{L(N_2^{L_1})})$. We will describe an NP machine F and define sets D and E in NP so that $C = L(F^{D\oplus E})$. Recall that, as in the proof of Theorem 3.28, for each n, there exists a set B_n such that:

1. $B_n \subseteq A \bigcap \Sigma^n$,
2. $||\, B_n\, || \leq n$, and
3. $(\forall x \in \Sigma^n)[x \in A \iff (\exists y \in B_n)[set\text{-}f(x,y) = \{x\}]]$.

Let p be a polynomial such that for every x and for every query string y of N_1 on input x, the query strings of N_2 on input y are of length at most $p(|x|)$; such a polynomial exists as both N_1 and N_2 are polynomial-time (nondeterministic) machines. Let x be a string whose membership in C we are testing. Let $n = |x|$. On input x, the machine F behaves as follows:

(a) For each $i, 1 \leq i \leq p(n)$, F guesses $B_i \subseteq \Sigma^i$ with $||\, B_i\, || \leq i$.

For each $i, 1 \leq i \leq p(n)$, and for each $y \in B_i$, F tests whether $y \in A$ by simulating an NP machine witnessing $A \in$ NP. If the test is not successful for some i and y, F rejects x.

(b) F asks whether $\langle 0^{p(n)}, B_1, \cdots, B_{p(n)}\rangle \in D$, where D is a set in NP defined by:

$$D = \{\langle 0^n, B_1, \cdots, B_n\rangle \,|\, (\exists i : 1 \leq i \leq n)(\exists y \in A^{=i} - B_i)(\forall z \in B_i)[set\text{-}f(y,z) = \{z\}]\}.$$

If the query is answered affirmatively, then F rejects x. Note that on questions to D asked during an actual run of F, the B_i are all subsets of A, so f will never be undefined during such runs.

(c) F simulates N_1 on x by substituting query r of N_1 to $L(N_2^{L_1})$ with query $\langle r, B_1, \cdots, B_{p(n)}\rangle$ to E.

F accepts x if and only if N_1 on x accepts in the above simulation.

The set E above is recognized by the following NP machine H.

(i) On input $\langle r, B_1, \cdots, B_{p(n)}\rangle$, H simulates N_2 on r. When N_2 asks L_1 about $\langle y_1, y_2\rangle$, H nondeterministically chooses one of the following three tests:

(A) Nondeterministically verify that $\langle y_1, y_2\rangle \in L_1$.

(B) Nondeterministically verify that $\langle y_1, y_2 \rangle \in L_2$.

(C) Check that $y_1 \notin B_{|y_1|}$ and $y_2 \notin B_{|y_2|}$ and nondeterministically verify that $(\forall z \in B_{|y_1|})[\mathit{set}\text{-}f(y_1, z) = \{z\}]$ and that $(\forall z \in B_{|y_2|})[\mathit{set}\text{-}f(y_2, z) = \{z\}]$.

Again, on actual calls to this machine made during actual runs of F, the B_i will be subsets of A, so f will always be defined.

(ii) If the chosen test above is not successful, then H rejects. If (A) was chosen and successful, then H assumes the answer to $\langle y_1, y_2 \rangle$ to be affirmative. If either (B) or (C) was chosen and successful, then H assumes the answer to $\langle y_1, y_2 \rangle$ to be negative.

H continues the simulation of N_2.

(iii) H accepts if and only if its simulation of N_2 leads to acceptance.

We claim that $F^{D \oplus E}$ accepts x if and only if $x \in C$. Note for any $B_1, \cdots, B_{p(n)}$ satisfying (1) and (2) above, it holds that

$$(\forall i : 1 \leq i \leq p(n))[B_i \text{ satisfies (3) }] \iff \langle 0^{p(n)}, B_1, \cdots, B_{p(n)} \rangle \notin D.$$

So, for some computation path, F enters step (c), and if it enters that step, the sets $B_1, \cdots, B_{p(n)}$ will satisfy (1), (2), and (3). Thus, in order to prove the claim, we have only to show that for any query $\langle y_1, y_2 \rangle$ of N_2 on r in H's simulation, it holds that

$$\langle y_1, y_2 \rangle \in L_1 \iff \text{neither (B) nor (C) is successful for } \langle y_1, y_2 \rangle.$$

Suppose that $\langle y_1, y_2 \rangle \in L_1$. Since $L_1 \bigcap L_2 = \emptyset$, (B) is not successful for $\langle y_1, y_2 \rangle$. Since $\langle y_1, y_2 \rangle \in L_1$, $y_1 \in A$. This implies $(\exists z \in B_{|y_1|})[\mathit{set}\text{-}f(y_1, z) \neq \{z\}]$. For such z, $\mathit{set}\text{-}f(y_1, z) = \{y_1\}$, as it cannot be undefined in this case. Thus, (C) is not successful.

Conversely, suppose that $\langle y_1, y_2 \rangle \notin L_1$. If $\langle y_1, y_2 \rangle \in L_2$, then (B) is successful and we are done. Suppose that $\langle y_1, y_2 \rangle \notin L_2$. Then $y_1 \notin A$ and $\mathit{set}\text{-}f(y_1, y_2)$ is either $\emptyset$ or $\{y_1\}$. This implies $y_2 \notin A$. So for any $z \in B_{|y_1|}$, $\mathit{set}\text{-}f(y_1, z) = \{z\}$ because $B_{|y_1|} \subseteq A$, and for much the same reason, for any $z \in B_{|y_2|}$, $\mathit{set}\text{-}f(y_2, z) = \{z\}$. Thus, (C) is successful. This completes the proof. ∎

We now turn to results on whether NP and coNP can be selective via selector functions more general than NPSV_t.

Theorem 3.34 The following results hold:

$\mathcal{C}$	NP $\subseteq \mathcal{C}$-selective	coNP $\subseteq \mathcal{C}$-selective
NPSV$_t$	holds iff NP = coNP	holds iff NP = coNP
NPSV	holds if NP = coNP holds only if $\mathrm{NP}^{\mathrm{NP}} = \mathrm{coNP}^{\mathrm{NP}}$	holds iff NP = coNP
NPMV$_t$	holds iff NP = coNP	holds iff NP = coNP
NPMV	holds (without any assumption)	holds iff NP = coNP

Corollary 3.35 The NPMV-selective sets are closed under complementation if and only if NP = coNP.

In the proof of Theorem 3.34 we will use the following lemma, which can be contrasted with Corollary 3.35.

Lemma 3.36

1. NP-sel is closed under complementation.

2. NPMV$_t$-sel is closed under complementation.

Proof of Lemma 3.36 Given an NP-selective set A with selector function $f \in \mathrm{NPSV}_t$, clearly $f'(x, y) = \{x, y\} - f(x, y)$ is an NPSV$_t$ function that is an NP-selector for $\overline{A}$. Given an NPMV$_t$-selective set A with selector function $f \in \mathrm{NPMV}_t$, consider the function f' defined so that (i) $\textit{set-}f'(x, y)$ contains x if and only if $\textit{set-}f(x, y)$ contains y, and (ii) $\textit{set-}f'(x, y)$ contains y if and only if $\textit{set-}f(x, y)$ contains x. Clearly f' is an NPMV$_t$-selector function for $\overline{A}$. ∎

Proof of Theorem 3.34 By Corollary 3.13, NP $\subseteq$ NPSV$_t$-sel if NP = coNP. So by Lemma 3.36, NP $\bigcup$ coNP $\subseteq$ NPSV$_t$-sel, and thus, clearly, NP $\bigcup$ coNP is also NPSV-, NPMV$_t$-, and NPMV-selective. By Theorem 3.8, we know that even NP $\subseteq$ NPMV$_t$-sel implies NP = coNP. Via Lemma 3.36 above, we may immediately also conclude that coNP $\subseteq$ NPMV$_t$-sel $\Rightarrow$ NP = coNP, and that coNP $\subseteq$ NPSV$_t$-sel $\Rightarrow$ NP = coNP.

Clearly, NP $\subseteq$ NPMV-sel. On the other hand, coNP $\subseteq$ NPMV-sel $\Rightarrow$ NP = coNP (and thus certainly coNP $\subseteq$ NPSV-sel $\Rightarrow$ NP = coNP). This is true as there is an NP algorithm for UNSAT under this assumption. In particular, the algorithm descends the disjunctive self-reducibility tree of the input formula, and if the NPMV function on input (a,b)—the two children of the current node—outputs a on a nondeterministic path that path explores the subtree rooted at child b, and if the NPMV function outputs b on a nondeterministic path that path explores the subtree rooted at child a. At each leaf reached, we accept if and only if an assignment denoted by that leaf does not satisfy the formula.

Finally, we must show that $\mathrm{NP} \subseteq \mathrm{NPSV\text{-}sel} \Rightarrow \mathrm{NP}^{\mathrm{NP}} = \mathrm{coNP}^{\mathrm{NP}}$. This is a corollary of Theorem 3.28, since it is well-known [Käm91,AFK89] that $\mathrm{NP} \subseteq (\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly} \Rightarrow \mathrm{NP}^{\mathrm{NP}} = \mathrm{coNP}^{\mathrm{NP}}$. Alternatively, it follows from Corollary 3.33. ∎

From Theorem 3.4, if NP-sel is bigger than P-sel, then $\mathrm{NP} \bigcap \mathrm{coNP} \neq \mathrm{P}$. Below, we derive a weaker conclusion from a weaker hypothesis.

Theorem 3.37 If there is an NPSV-selective set that is not P-selective, then NP contains a pair of P-inseparable sets, i.e., disjoint sets $A \in \mathrm{NP}$ and $B \in \mathrm{NP}$ such that for no set $C \in \mathrm{P}$ do we have $A \subseteq C$ and $B \bigcap C = \emptyset$.

Proof If f is our NPSV-selector, then $\{\langle x, y\rangle \mid f \text{ is defined and } x = f(x,y)\} \in \mathrm{NP}$ and $\{\langle x, y\rangle \mid f \text{ is defined and } y = f(x,y)\} \in \mathrm{NP}$ clearly are P-inseparable unless the NPSV-set is P-selective. ∎

An interesting open issue highlighted by Theorem 3.34 is whether some complexity class collapse completely characterizes whether all NP sets are NPSV-selective. However, the necessary and sufficient conditions we've obtained already address an important problem raised by Fenner et al. [FHOS93] and Selman [Sel], namely, whether all NPMV functions have NPSV refinements (that is, whether for each NPMV function there is an NPSV function having the same domain whose output set when defined is a subset of the outputs of the NPMV function). One can pose this question, equivalently and more concretely, as: Is there a function $f \in \mathrm{NPSV}$ such that $domain(f) = \mathrm{SAT}$ and $(\forall F \in SAT)[f(F)$ is a satisfying assignment of $F]$ [Sel]? Informally, can an NP function "pick off" a single satisfying assignment of a satisfiable formula? (This question can be compared with the results of Valiant and Vazirani [VV86] on probabilistically reducing SAT to a question about (solutions to a promise problem regarding) unambiguous computation.) It follows from Theorem 3.34 that all NPMV functions have NPSV refinements if the polynomial hierarchy collapses to NP (this can be seen also from a result of [Sel]), and that all NPMV functions have NPSV refinements only if the polynomial hierarchy collapses to $\mathrm{NP}^{\mathrm{NP}}$. The latter result follows from the fact that, from the NPMV box of the table of Theorem 3.34, all NP sets are NPMV-selective, and thus if NPMV functions all have NPSV refinements, then all NP sets are NPSV-selective (as it is not hard to see that any NPSV refinement of an NPMV-selector is itself an NPSV-selector for the same set), and the results of the appropriate NPSV-selective box of Theorem 3.34 apply.

Corollary 3.38 If every NPMV function has an NPSV refinement, then $\mathrm{NP}^{\mathrm{NP}} = \mathrm{PH}$.

Corollary 3.38 provides the most compelling evidence to date that not all NPMV functions have NPSV refinements. Previously, it was known only that (a) relative to a random oracle not all NPMV functions have NPSV refinements [Roy93], and (b) if all NPMV functions have NPSV refinements, then there exist disjoint NP-Turing-complete sets that are separated only by NP-hard sets [Sel].

Corollary 3.38, though establishing strong evidence that NPMV functions lack NPSV refinements, leaves open the question of whether all NPMV functions have single-valued refinements computable in $\mathrm{FP}_{tt}^{\mathrm{NP}}$ (see [WT93,AFW92,HNOS93,NOS93,Sel] for background and discussion). In particular, if every NPMV function has a single-valued $\mathrm{FP}_{tt}^{\mathrm{NP}}$ refinement, does it immediately follow that the polynomial hierarchy collapses? It seems that such a conclusion would require techniques substantially different than those of this paper. In particular, our result that "NPMV has NPSV refinements only if PH = $\mathrm{NP}^{\mathrm{NP}}$" itself relativizes. However, any relativizable proof of the $\mathrm{FP}_{tt}^{\mathrm{NP}}$ analog would immediately—due to the result of Watanabe and Toda [WT93] that the analog holds relative to a random oracle—imply that the polynomial hierarchy collapses relative to a random oracle. Though it remains plausible that the polynomial hierarchy collapses relative to a random oracle, the conventional wisdom is that the hierarchy is does not collapse relative to a random oracle (see [Cai86] for some indication of the difficulties surrounding this issue).

Note that, in the table contained in the statement of Theorem 3.34, six of the eight boxes have the characterization NP = coNP. This might lead one to wonder to what extent the three NP function classes other than NPSV_t listed in Theorem 3.34 share the results of this paper. Already, in the lowness and nonuniform characterization results, we have seen cases where we proved stronger results for NPSV_t-selectivity than for its three brethren. On the other hand, Theorem 3.8 applies to NPMV_t reductions, and thus the NPSV_t theorems projected as corollaries of that theorem also apply to NPMV_t reductions, as do such claims as, e.g.,

If $A \leq_{pos}^{p} B$ and B is NPMV_t-selective, then A is also NPMV_t-selective.

Making no claim to exhaustively study the three brethren, we nonetheless note in the comments below some results on the extent to which other results do, or do not, extend to the three brethren.

Comment 3.39

1. The proof of Lemma 3.5, Part 3, is quite specific to NPSV_t, as it is based on "strong" computation. It is easy to see that $\mathrm{NP}^{\mathrm{NPSV}} = \Sigma_2^p$, and that $\mathrm{NP}^{\mathrm{NPPV}_t} = \mathrm{NP}^{\mathrm{NP2V}_t} = \Sigma_2^p$, where NPPV_t indicates an NPMV_t function having at most polynomially many

values per input ([BLS84,Sel] have studied the partial analog of this class), where $\mathrm{NP2V}_t$ indicates an NPMV_t function having at most two values per input, and where our model (call it the *set model*) of accessing a multivalued function oracle f having at most polynomially many values per input is that the returned value is *set-f*. Fenner et al. [FHOS93] have introduced a model, different from that mentioned in the previous sentence, in which a machine accessing a multivalued function oracle must compute robustly with respect to every single-valued refinement of the oracle; we will call this model the *value model*. Analogously to results of Fenner et al. [FHOS93], we note that in the value model $\mathrm{NP}^{\mathrm{NPMV}} = \Sigma_2^p$ and $\mathrm{NP}^{\mathrm{NPMV}_t} = \mathrm{NP}$.

2. Regarding Lemma 3.5, Part 4, the proof is also class-specific. However, it is easy to note that ${\mathrm{NPSV}_t}^{\mathrm{NPSV}} = {\mathrm{NPSV}_t}^{\mathrm{NP}}$, that in the value model ${\mathrm{NPSV}_t}^{\mathrm{NPMV}} = {\mathrm{NPSV}_t}^{\mathrm{NP}}$, and that in the set model ${\mathrm{NPSV}_t}^{\mathrm{NPPV}_t} = {\mathrm{NPSV}_t}^{\mathrm{NP2V}_t} = {\mathrm{NPSV}_t}^{\mathrm{NP}}$. Analogously to a result of Fenner et al. [FHOS93], in the value model ${\mathrm{NPSV}_t}^{\mathrm{NPMV}_t} = \mathrm{NPSV}_t$.

Comment 3.40 If $\overline{A} \leq_\gamma A$ and A is NPMV-selective [sic.], then $A \in \mathrm{NP} \bigcap \mathrm{coNP}$.

Comment 3.41

1. We already know from Theorem 3.8 that if A is NP-hard with respect to $\leq_{1\text{-}tt}^p$ reductions and A is NPMV_t-selective, then NP=coNP. Note that, from Theorem 3.34 this cannot be generalized to NPMV-selectivity unless NP = coNP. It remains an interesting open issue whether it can be generalized to NPSV-selectivity.

2. $\mathrm{coNP} \subseteq \mathrm{R}_\gamma(\text{NP-sel}) \iff \mathrm{NP} = \mathrm{coNP}$.

Comment 3.42 If $\overline{A} \leq_{pos}^p A$ and A is NPMV-selective, then $A \in \mathrm{NP}$. If $\overline{A} \leq_{pos}^p A$ and A is NPMV_t-selective, then $A \in \mathrm{NP} \bigcap \mathrm{coNP}$.

Finally, we briefly introduce, as open topics commended to the reader for future research, selectivity via #P, OptP, and SpanP functions. Since these functions map to integers rather than to strings, for them we will adopt an appropriately adjusted definition of selectivity. Notationally, we will not distinguish between this and the previous definition of selectivity via functions (Definition 3.3), as the appropriate definition will be implicit from the codomain of the function being discussed.

For each $z \in \Sigma^*$, let $int(z)$ be the integer j such that z is the lexicographically $(j+1)$-st string in Σ^*. Note that $int(z)$ is an honest, polynomial-time computable and invertible bijection between Σ^* and $\{0, 1, \cdots\}$.

Definition 3.43 (Selectivity for Function Classes Whose Codomain is the Natural Numbers) Let $\mathcal{FC}$ be any class of single-valued total functions mapping from Σ^* to

$\{0, 1, \cdots\}$. A set A is $\mathcal{FC}$-selective if there is a function $f \in \mathcal{FC}$ so that, for every $x, y \in \Sigma^*$,

1. $f(\langle x, y\rangle) \in \{int(x), int(y)\}$, and
2. if $x \in A$ or $y \in A$, then $int^{-1}(f(\langle x, y\rangle)) \in A$.

$\mathcal{FC}$-sel will denote the class of $\mathcal{FC}$-selective sets.

Let #P [Val79], OptP [Kre88], and SpanP [KST89] be as defined in the standard literature. Very informally, #P is the class of functions that count accepting paths of nondeterministic polynomial-time Turing machines, SpanP is the class of functions that count the number of distinct output values of accepting paths of nondeterministic polynomial-time Turing machines, and OptP is the class of functions that compute the maximum output value among the accepting paths of nondeterministic polynomial-time Turing machines.

It is not hard to see, via the intuitive direct constructions, that $\mathrm{R}^p_{1\text{-}tt}(\mathrm{UP}) \subseteq \#\mathrm{P}$-sel and $\mathrm{R}^p_{1\text{-}tt}(\mathrm{NP}) \subseteq \mathrm{OptP}$-sel. In particular, $\mathrm{UP} \bigcup \mathrm{coUP} \subseteq \#\mathrm{P}$-sel and $\mathrm{NP} \bigcup \mathrm{coNP} \subseteq \mathrm{OptP}$-sel. Though OptP is closely associated with Δ^p_2, to subsume Δ^p_2 with an OptP-like selectivity class one would have to slightly enhance the power of OptP. Let us define OptP$'$ to be the classes of functions that for some nondeterministic polynomial-time machine on each input compute the output of the lexicographically maximum accepting path (this should not be confused with OptP, where the maximum output over all accepting paths is computed). Though this class of functions is "metrically" equivalent (that is, 1-Turing interreducible) to OptP, it nonetheless does subsume Δ^p_2: $\Delta^p_2 \subseteq \mathrm{OptP}'$-sel.

Combining the meta-result that "for any 'nice' function class $\mathcal{FC}$, $\mathcal{FC}$-sel $\subseteq$ $(\mathrm{R}^p_{tt}(\mathcal{FC}))$/poly" with appropriate class-specific manipulations [PZ83,CH90,OH93] converting truth-table reductions to 1-truth-table reductions, it is not hard to see that #P-sel $\subseteq$ $(\mathrm{R}^p_{1\text{-}tt}(\#\mathrm{P}))$/poly, OptP-sel $\subseteq (\mathrm{R}^p_{1\text{-}tt}(\mathrm{OptP}))$/poly, and SpanP-sel $\subseteq (\mathrm{R}^p_{1\text{-}tt}(\mathrm{SpanP}))$/poly.

Acknowledgments

We thank Edith Hemaspaandra for many helpful comments and suggestions. We thank Arun Jagota for suggesting a proof of Fact 2.11. We thank Johannes Köbler for bringing to our attention the issue of Theorem 3.27. We thank Marius Zimand and Yenjo Han for their careful, helpful proofreading of an earlier draft.

References

[AFK89] M. Abadi, J. Feigenbaum, and J. Killian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39:21–50, 1989.

[AFW92] K. Abrahamson, M. Fellows, and C. Wilson. Parallel self-reducibility. In *Proceedings of the 4th International Conference on Computing and Information*, pages 67–70. IEEE Computer Society Press, May 1992.

[AH92] E. Allender and L. Hemachandra. Lower bounds for the low hierarchy. *Journal of the ACM*, 39(1):234–250, 1992.

[AHH$^+$] V. Arvind, Y. Han, L. Hemachandra, J. Köbler, A. Lozano, M. Mundhenk, M. Ogiwara, U. Schöning, R. Silvestri, and T. Thierauf. Reductions to sets of low information content. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory.* Cambridge University Press. To appear.

[AHOW92] E. Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. *SIAM Journal on Computing*, 21(3):521–539, 1992.

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractibility. In *Proceedings of the 9th ACM Symposium on Theory of Computing*, pages 151–153, 1977.

[BBS86] J. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM Journal on Computing*, 15(3):739–746, 1986.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977.

[BK88] R. Book and K. Ko. On sets truth-table reducible to sparse sets. *SIAM Journal on Computing*, 17(5):903–919, 1988.

[BLS84] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[BLS85] R. Book, T. Long, and A. Selman. Qualitative relativizations of complexity classes. *Journal of Computer and System Sciences*, 30:395–413, 1985.

[BLS93] H. Buhrman, L. Longpré, and E. Spaan. SPARSE reduces conjunctively to TALLY. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 208–214. IEEE Computer Society Press, May 1993.

[BTvEB93] H. Buhrman, L. Torenvliet, and P. van Emde Boas. Twenty questions to a P-selector. Manuscript, 1993.

[BvHT93] H. Buhrman, P. van Helden, and L. Torenvliet. P-selective self-reducible sets: A new characterization of P. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 44–51. IEEE Computer Society Press, May 1993.

[Cai86] J. Cai. *On Some Most Probable Separations of Complexity Classes.* PhD thesis, Cornell University, Ithaca, NY, 1986.

[CGH+88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.

[CGH+89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.

[CH90] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23(2):95–106, 1990.

[CHV] J. Cai, L. Hemachandra, and J. Vyskoč. Promises and fault-tolerant database access. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory*. Cambridge University Press. To appear.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[EY80] S. Even and Y. Yacobi. Cryptocomplexity and NP-completeness. In *Proceedings of the 7th International Colloquium on Automata, Languages, and Programming*, pages 195–207. Springer-Verlag *Lecture Notes in Computer Science*, 1980.

[FHOS93] S. Fenner, S. Homer, M. Ogiwara, and A. Selman. On using oracles that compute values. In *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science*, pages 398–407. Springer-Verlag *Lecture Notes in Computer Science #665*, February 1993.

[Gav92a] R. Gavaldà. Bounding the complexity of advice functions. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 249–254. IEEE Computer Society Press, June 1992.

[Gav92b] R. Gavaldà. On conjunctive and disjunctive reductions to sparse sets. Manuscript, January 1992.

[GB91] R. Gavaldà and J. Balcázar. Strong and robustly strong polynomial time reducibilities to sparse sets. *Theoretical Computer Science*, 88(1):1–14, 1991.

[GHJY91] J. Goldsmith, L. Hemachandra, D. Joseph, and P. Young. Near-testable sets. *SIAM Journal on Computing*, 20(3):506–523, 1991.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

[Gur91] Y. Gurevich. Average case completeness. *Journal of Computer and System Sciences*, 42:346–398, 1991.

[GW] R. Gavaldà and O. Watanabe. On the computational complexity of small descriptions. *SIAM Journal on Computing.* To appear. Preliminary version appears in *Proceedings of the 6th Structure in Complexity Theory Conference (1991)*, IEEE Computer Society Press, pp. 89–101.

[HH91] L. Hemachandra and A. Hoene. On sets with efficient implicit membership tests. *SIAM Journal on Computing*, 20(6):1148–1156, 1991.

[HHT93] Y. Han, L. Hemachandra, and T. Thierauf. Threshold computation and cryptographic security. Technical Report TR-461, University of Rochester, Department of Computer Science, Rochester, NY, June 1993.

[HNOS93] E. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. Technical Report 93-21, State University of New York at Buffalo, Department of Computer Science, Buffalo, NY, 1993.

[Joc68] C. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the AMS*, 131(2):420–436, 1968.

[JY90] D. Joseph and P. Young. Self-reducibility: Effects of internal structure on computational complexity. In A. Selman, editor, *Complexity Theory Retrospective*, pages 82–107. Springer-Verlag, 1990.

[Kad89] J. Kadin. $P^{NP[\log n]}$ and sparse Turing-complete sets for NP. *Journal of Computer and System Sciences*, 39(3):282–298, 1989.

[Käm91] J. Kämper. Non-uniform proof systems: a new framework to describe non-uniform and probabilistic complexity classes. *Theoretical Computer Science*, 85(2):305–331, 1991.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, April 1980.

[Ko83] K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26:209–221, 1983.

[Ko89] K. Ko. Distinguishing conjunctive and disjunctive reducibilities by sparse sets. *Information and Computation*, 81(1):62–87, 1989.

[Köb93] J. Köbler. Locating P/poly optimally in the extended low hierarchy. In *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science*, pages 28–37. Springer-Verlag *Lecture Notes in Computer Science #665*, February 1993.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[KS85] K. Ko and U. Schöning. On circuit-size complexity and the low hierarchy in NP. *SIAM Journal on Computing*, 14(1):41–51, 1985.

[KST89] J. Köbler, U. Schöning, and J. Torán. On counting and approximation. *Acta Informatica*, 26:363–379, 1989.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[Lon82] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science*, 21:1–25, 1982.

[LS91] T. Long and M. Sheu. A refinement of the low and high hierarchies. Technical Report OSU-CISRC-2/91-TR6, Ohio State University, Department of Computer Science, Columbus, Ohio, February 1991.

[LS93] L. Longpré and A. Selman. Hard promise problems and nonuniform complexity. *Theoretical Computer Science*, 115(3):277–290, 1993.

[MP79] A. Meyer and M. Paterson. With what frequency are apparently intractable problems difficult? Technical Report MIT/LCS/TM-126, MIT Laboratory for Computer Science, Cambridge, MA, 1979.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[NOS93] A. Naik, M. Ogiwara, and A. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 52–64. IEEE Computer Society Press, May 1993.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory for closure properties. *Journal of Computer and System Sciences*, 46:295–325, 1993.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[Ric89] C. Rich. Positive relativizations of the P=?NP problem. *Journal of Computer and System Sciences*, 38(3):511–523, 1989.

[Roy93] J. Royer, August 1993. Personal Communication.

[Sch83] U. Schöning. A low and a high hierarchy in NP. *Journal of Computer and System Sciences*, 27:14–28, 1983.

[Sch86] U. Schöning. Complete sets and closeness to complexity classes. *Mathematical Systems Theory*, 19(1):29–42, 1986.

[Sel] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*. In press.

[Sel74] A. Selman. On the structure of NP. *Notices of the AMS*, 2:310, 1974.

[Sel78] A. Selman. Polynomial time enumeration reducibility. *SIAM Journal on Computing*, 7(4):440–457, 1978.

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

[Sel82a] A. Selman. Analogues of semirecursive sets and effective reducibilities to the study of NP complexity. *Information and Control*, 52:36–51, 1982.

[Sel82b] A. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[Sel88] A. Selman. Promise problems complete for complexity classes. *Information and Computation*, 78:87–98, 1988.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[TB91] S. Tang and R. Book. Reducibilities on tally and sparse sets. *Theoretical Informatics and Applications (RAIRO)*, 25(3):293–302, 1991.

[Tod91] S. Toda. On polynomial-time truth-table reducibilities of intractable sets to P-selective sets. *Mathematical Systems Theory*, 24:69–82, 1991.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5:20–23, 1976.

[Val79] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[WT93] O. Watanabe and S. Toda. Structural analysis of the complexity of inverse functions. *Mathematical Systems Theory*, 26:203–214, 1993.

[Yap83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.

[Yes83] Y. Yesha. On certain polynomial-time truth-table reducibilities of complete sets to sparse sets. *SIAM Journal on Computing*, 12(3):411–425, 1983.