The Complexity of Finding Top-Toda-Equivalence-Class Members*

Lane A. Hemaspaandra and *Mitsunori Ogihara*
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

Mohammed J. Zaki
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Marius Zimand
Department of Computer and Information Sciences
Towson University
Towson, MD 21252, USA

December 7, 2003; revised September 30, 2004

Abstract

We identify two properties that for P-selective sets are effectively computable. Namely we show that, for any P-selective set, finding a string that is in a given length's top Toda equivalence class (very informally put, a string from Σ^n that the set's P-selector function declares to be most likely to belong to the set) is $\mathrm{FP}^{\Sigma_2^p}$ computable, and we show that each P-selective set contains a weakly-$\mathrm{P}^{\Sigma_2^p}$-rankable subset.

1 Overview

P-selectivity is a generalization of P. A set A is in P if there is a polynomial-time algorithm which, given any string x, determines whether x belongs to A. In contrast, a set A is P-selective [Sel79,Sel82b] if there is a polynomial-time algorithm (called a P-selector function) that, given any two strings x and y, outputs one of those strings, and such that the algorithm

*Email: `{lane,ogihara}@cs.rochester.edu`, `zaki@cs.rpi.edu`, `mzimand@towson.edu`. Supported in part by NSF grants CCR-9322513, INT-9815095/DAAD-315-PPP-gü-ab, EIA-0080124, DUE-9980943, and EIA-0205061, and NIH grant P30-AG18254. A preliminary version of this paper appeared in the LATIN 2004 conference.

has the property that if at least one of x or y is in A, then the one the algorithm outputs belongs to A. Informally, it always places a bet on one of them being in the set, and it wins whenever winning such a bet is possible.

The book [HT03] provides a recent overview of the state of research regarding P-selectivity theory (see also the somewhat older article [DHHT94]). Nickelsen's thesis [Nic01] and the recent survey article by Nickelsen and Tantau [NT03] are also very good starting points regarding the study of partial information classes such as the P-selective sets.

A key notion used in P-selectivity theory is the notion of a Toda equivalence class which, very loosely put, is a strongly connected component of the graph induced by a given P-selector function on the strings of a given length. This paper studies the complexity of finding a string from a given P-selector function's top Toda equivalence class—that is, a string from the unique strongly connected component that can be reached from no other strongly connected component.

2 Introduction and Background

P-selectivity theory has many features making it an interesting complexity-related research area. P-selectivity represents a natural generalization of feasible decidability. P-selectivity has a well-studied analog in computability theory, namely, the semi-recursive sets [Joc68]. The exploration of P-selectivity had a strong, unexpected impact on the study of NP functions, solutions, and ambiguity, namely, the nondeterministic version of selectivity is the central tool used to show that SAT has unique solutions only if the polynomial hierarchy collapses [HNOS96b]; and, relatedly, selectivity has proven central in understanding more generally whether one can reduce the number of solutions of NP functions [HNOS96b,Ogi96, NRRS98,HOW02]. Selectivity has also given insight into the separation of reducibility notions within NP [Sel82a,Sel79].

Informally, P-selectivity captures the notion of sets for which there is a polynomial-time algorithm f telling which of any two given elements is "logically no less likely to be in the set" (see Definition 3.1). Such sets are called P-selective. P-selective sets can be arbitrarily complex: For every tally set A, there is a P-selective set that is Turing-equivalent to A [Sel79, Sel82a]. In particular, some P-selective sets are uncomputable. Despite this, in the present paper we identify natural tasks that are computable for all P-selective sets. Indeed, these natural tasks are even computable within relatively low levels of the polynomial hierarchy.

The first such task is to produce, for an arbitrary P-selective set A (which, w.l.o.g., has at least one commutative P-selector function), at each length n, a string that is "most likely" to be in A. Let us explain a bit more what we mean by this. Each commutative polynomial-time P-selector function f for A will implicitly specify a structure of equivalence classes of strings (at a given length) that are equally likely (according to f) to be in A, and these classes can be ordered with respect to the order that one might informally call *no-less-likely-to-be-in-A* (we will explain how to define such equivalence classes, which themselves depend on only the P-selector function and not on A, in rigorous detail in Section 3 after introducing the tournament-graph model that is useful in their definition; in brief, two strings of the same length are said to be equivalent exactly if there are chains of applications of the P-selector function leading from each to the other). We will call those classes Toda equivalence classes, in light of Toda [Tod91], and the related order will be called a Toda order.[1] Informally put, a Toda equivalence class ζ, with respect to commutative P-selector function f, of length n strings has the property that for *every* set A for which f is a P-selector, either all the strings in ζ are in A or none of the strings in ζ are in A. And (restricting ourselves as we will globally do to just looking at strings all of the same length) each Toda equivalence class is a maximal set of strings for which this can be said.

The Toda-class approach's ordering implications play a central role in a wide range of results, ranging for example from the study of whether P-selective sets can be hard for standard complexity classes [Tod91] to the study of associative P-selector functions [HHN04]. In this paper, we seek to better understand the Toda classes' own complexity. We prove that finding an element in the top Toda class of a length (very informally and intuitively—and not quite correctly—put, a string that among the strings of that length is "most likely" to be in A) can be done with an $\mathrm{FP}^{\Sigma_2^p}$ computation.

The second task we study is that of weak-P-rankability. A function f *weakly ranks* a set A if, for any string x that is in A, $f(x)$ returns the rank of x in A; in other words, it says how many strings lexicographically less than or equal to x are in A. A set A is weakly-P-rankable if it has a function f that is computable in polynomial time and that weakly ranks

[1] One must be a bit careful (regarding what we mean by a Toda Order and to avoid confusion with a different way some people speak of Toda Order(ings)—please see the end of this footnote; also, detailed definitions are provided in Section 3). These Toda equivalence classes indeed induce (within each length) an order. And for a fixed, commutative P-selector function f one can take all 2^n strings of length n and compute their relationships with respect to this order, namely, via making their tournament graph (a notion that will be defined in detail in Section 3) with respect to f and then collapsing all the cycle-linked nodes (i.e., each strongly connected component) into points (each representing one of our equivalence classes). Note that since our graph (related to all strings in Σ^n) is a tournament, after doing such collapsing it will always hold that what is left is simply a linearly ordered list of our equivalence classes. Thus, one can in polynomial time (in the number, 2^n, of strings we are speaking of) arrange the strings in a list in such a way that one knows that, for each set A for which f is a P-selector, the intersection between A and the list is guaranteed to be a (potentially empty) suffix of the list.

One can take a similar approach on other tournament graphs. In particular, for a fixed, commutative P-selector function f one can, given as input any collection of (in our setting, same-length, but this also holds without that constraint) strings, in time polynomial in the maximum of the number of strings and the maximum size of any of the strings, compute and collapse each cycle-connected group of nodes (in the tournament graph constructed on the strings in the collection), and from this one can easily order the nodes into a list with the property that, for each set A for which f is a P-selector, the intersection between A and the list is guaranteed to be a (potentially empty) suffix of the list. This is often known as the Toda Ordering Lemma. As we just mentioned, one can see and prove it via invoking the flavor of the Toda Order except on a set of strings other than Σ^n. The Toda Ordering Lemma is an extremely powerful tool that, for example, allows one to—given access to an oracle for the P-selective set—when given m strings (in our setting same-length strings, though one can use the Toda Ordering Lemma without that restriction) determine, with only about $\log m$ queries to the P-selective set oracle, exactly which of the input strings belong to the P-selective set. For completeness and to avoid confusion, we mention that the term Toda Order(ing Lemma) is also sometimes used in the literature for any polynomial-time procedure that given any collection of strings reorders them in such a way that their intersection with each P-selective set for which f is a P-selector is guaranteed to be a (potentially empty) suffix of the list.

A. The relation between P-selectivity and weak-P-rankability has been studied extensively by Hemaspaandra, Zaki, and Zimand in a paper [HZZ96] that focused on polynomial-time semi-rankable sets, i.e., sets that are simultaneously P-selective and weakly-P-rankable. It is shown there that there are P-selective sets that are not weakly-P-rankable. In partial contrast, in the present paper we show that any infinite P-selective set has an infinite subset that is weakly rankable by an $\mathrm{FP}^{\Sigma_2^p}$ function. We also obtain a result about the relationship between P-selectivity and weak-P-rankability. All the P-selective sets that have been considered in the literature are either standard left cuts or are structurally similar to a standard left cut (more precisely $\leq_m^p$-reducible to a standard left cut). We prove that if a standard left cut is weakly-P-rankable, then it is in P. Regarding this section (Section 4), we particularly commend to the reader's attention the proof of Lemma 4.3, which we feel to be a novel technique. Lemma 4.3 is given prompt application in yielding the theorems that are stated immediately after it.

3 Instantiating the Top Toda Class

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ and let $\mathbb{N}^+ = \{1, 2, 3, \ldots\}$. Our alphabet will be $\Sigma = \{0, 1\}$. For any set A, $||A||$ denotes the cardinality of A. For any string $x \in \Sigma^*$, $|x|$ denotes the length of x. For any set A and any nonnegative integer n, $A^{=n}$ denotes $\{x \mid x \in A \wedge |x| = n\}$. For any set A and any string x, $A^{\leq x}$ denotes $\{y \mid y \in A \wedge y \leq_{lex} x\}$, where $\leq_{lex}$ denotes the standard lexicographical ordering. For the definitions of standard complexity classes such as P, NP, Σ_k^p, etc., we refer the reader to, for example, the complexity handbook by Hemaspaandra and Ogihara [HO02]. As is standard, FP denotes the class of all (total, single-valued) polynomial-time computable functions, $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$, $\Sigma_3^p = \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}$, $\mathrm{E} = \bigcup_{k \geq 0} \mathrm{DTIME}[2^{kn}]$, and $\mathrm{NE} = \bigcup_{k \geq 0} \mathrm{NTIME}[2^{kn}]$. Given classes $\mathcal{C}$ and $\mathcal{D}$, as is standard, we say that $\mathcal{C}$ *is* $\mathcal{D}$-*immune* (equivalently, $\mathcal{C}$ *is immune to* $\mathcal{D}$) if there is an infinite set $A \in \mathcal{C}$ such that no infinite subset of A is a member of $\mathcal{D}$. And, as is standard, we say that $\mathcal{C}$ *is* $\mathcal{D}$-*bi-immune* (equivalently, $\mathcal{C}$ *is bi-immune to* $\mathcal{D}$) if there is a set $A \in \mathcal{C}$ such that no infinite subset of A is a member of $\mathcal{D}$ and no infinite subset of $\overline{A}$ is a member of $\mathcal{D}$ (see [BS85]).

Definition 3.1 *[Sel79,Sel82b] A set A is* P-selective *if there is a (total, single-valued) polynomial-time computable function f such that, for every x and y, it holds that*

1. $f(x, y) = x$ *or* $f(x, y) = y$, *and*

2. $\{x, y\} \bigcap A \neq \emptyset \Rightarrow [(x \in A$ *and* $f(x, y) = x)$ *or* $(y \in A$ *and* $f(x, y) = y)]$.

We use P-sel *to denote the class of all sets that are* P-*selective.*

The function f appearing in Definition 3.1 is called a *P-selector* or a *P-selector function*.[2] A P-selector (function) f is *commutative* if it has the property that for all x and y in Σ^*, $f(x, y) = f(y, x)$. Each P-selective set A has a commutative P-selector function, because we can replace an arbitrary P-selector f for A with $f'(x, y) = f(\min(x, y), \max(x, y))$. It is easy to see that f' is a commutative P-selector for A. Since all P-selective sets have commutative P-selector functions, it is very common in the literature to focus on commutative P-selector functions. However, so that the proofs remain self-contained, we note explicitly each time we require a function to be commutative.

A tournament graph $G = (V_G, E_G)$ is a complete oriented graph, i.e., a directed graph having the property that, for every two (possibly equal) nodes a and b, $||\{(a, b), (b, a)\} \cap E_G|| = 1$. A commutative P-selector induces a tournament graph G_f on Σ^*. The nodes are the strings in Σ^* and there is an edge (x, y) (also denoted $x \geq_f y$) exactly if $f(x, y) = x$. When speaking in plain text, we will use synonymously with this the terms x *beats* y, x *wins at* y, and y *loses to* x. (Note that each x both wins at itself and loses to itself, and our tournament graphs have self-loops at each node.) We say $x >_f y$ if $x \geq_f y$ and $x \neq y$.

Let $G_{f,n}$ be the subgraph of G_f induced by the nodes in Σ^n. Two nodes $x, y \in \Sigma^n$ are *Toda-equivalent*, notated $x \equiv_{Toda} y$, if $G_{f,n}$ contains a path from x to y and a path from y to x. The relation $\equiv_{Toda}$ is an equivalence relation. We will denote the equivalence class of $x \in \Sigma^n$ by $[x]_f$. For strings x and y of the same length, we order their equivalence classes as follows: $[x]_f \geq [y]_f$ holds exactly if $x \geq_f y$. For strings x and y of the same length, we say that $[x]_f > [y]_f$ exactly if $[x]_f \geq [y]_f$ and $[x]_f \neq [y]_f$.

(It is easy to see that these relations are consistent when applied only among strings all of the same length. Note that we neither define nor ever use $[w]_f > [z]_f$ or $[w]_f \geq [z]_f$ for the case where $|w| \neq |z|$. Our focus will always be on collections of same-length strings. However, if one wanted to compare different-length strings, for the purposes of this paper let us say that equivalence classes of different lengths are always, by definition, incomparable, and so viewed as being over all of Σ^* our classes form a partial rather than a total order. Since our focus is within a length and there our order is never undefined, we will for simplicity simply use the term "order."[3])

Of course, $[y]_f < [x]_f$ means the same as $[x]_f > [y]_f$, and $[y]_f \leq [x]_f$ means the same as $[x]_f \geq [y]_f$.

The following fact holds.

[2] To be formally correct, the term should be FP-selector, since the function f is taken from the class FP, the single-valued, total, polynomial-time computable *functions*. Nonetheless, the term P-selective is by long tradition the one used for this notion. However, when later we wish to indicate other class of functions with a broader class of possible functions, we for clarity will notate it in a way that is rigorously correct, that is, with the "F" included, e.g., $\mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}}$-selector.

[3] It is also true that one can apply the notion of collapsing cycles to all of Σ^* in one fell swoop, thus in fact allowing equivalence classes to potentially span multiple lengths and to potentially be infinite in size. Doing so crisply creates a total order. However, the study of P-selective sets centrally focuses on strings of one length at a time, and builds tournaments based on those. So, throughout this paper, we will have a "best-at-length" focus, and will use equivalence classes that focus on a single length.

Fact 3.2 *1. If A is a* P*-selective set having commutative function f as a* P*-selector then for all $x \in \Sigma^n$*

$$A \cap [x]_f = [x]_f \text{ or } A \cap [x]_f = \emptyset.$$

2. Let A be a P*-selective set having commutative function f as a* P*-selector. If $A \cap [x]_f = [x]_f$ then for all y such that $|y| = |x|$ and $[y]_f \geq [x]_f$, it holds that $A \cap [y]_f = [y]_f$. (In fact, it even holds that if $A \cap [x]_f = [x]_f$ then for* all *y (regardless of $|y|$) such that $f(x, y) = y$ it holds that $A \cap [y]_f = [y]_f$.)*

These equivalence classes (related to strings of length n) form a partition of Σ^n. In particular, there are strings $x_1^n, x_2^n, \ldots, x_k^n \in \Sigma^n$ such that the equivalence classes $[x_1^n]_f, \ldots, [x_k^n]_f$ form a partition of Σ^n and

$$[x_1^n]_f > [x_2^n]_f > \ldots > [x_k^n]_f.$$

The set $[x_1^n]_f$ is called the *top Toda class* (at length n with respect to f).

Definition 3.3 *Let f be a commutative* P*-selector function. A function g is a* BestAtLength function (for f) *if, for each n, $g(1^n) \in [x_1^n]_f$, i.e., it outputs an element of the top Toda class at length n with respect to f.*

Each string that belongs to the top Toda class (at length n with respect to f) will be called a *top Toda element* (at length n with respect to f).

Note that Definition 3.3 does not mention what set f is a P-selector for, since each BestAtLength function will work equally well for each of the potentially uncountably many sets for which f is a P-selector.

Theorem 3.4 *Every commutative* P*-selector f has a BestAtLength function computable in* $\mathrm{FP}^{\Sigma_2^p}$.

Proof. On input 1^n we must produce a string x in Σ^n such that for any string $y \in \Sigma^n$ there is a path from x to y. If $n = 0$, the BestAtLength function outputs ϵ (the empty string). So let us consider for the rest of the proof only the case $n \in \mathbb{N}^+$. From the work of Ko ([Ko83], see also [HT03]), it is known that for each $n \in \mathbb{N}^+$ (the claim actually fails at $n = 0$ but that case is already handled for us above) there is a set $H_n \subseteq \Sigma^n$, $||H_n|| \leq n$, such that for every $x \in \Sigma^n$ there is a string $h \in H_n$ with $f(x, h) = h$. We will call such a set a *dominating set*. To simplify the discussion, we will assume without loss of generality that $||H_n|| = n$. This is a legal "without loss of generality," since we have restricted ourselves to the case $n \in \mathbb{N}^+$ and for all such n and for all H_n having cardinality less than n we can extend the dominating set by adding $n - ||H_n||$ new, appropriate-length elements. Let $H_n = \{y_1, \ldots, y_n\}$, with $y_i \in \Sigma^n$, and $y_1 < y_2 < \ldots < y_n$ ($<$ here being with respect to lexicographical order). Then H_n will be encoded by the concatenation of the strings y_i, i.e., by $y_1 y_2 \ldots y_n$. We will denote this string by $enc(H_n)$. Note that $|enc(H_n)| = n^2$. Also, if a string y is the encoding of some set $B \subseteq \Sigma^n$ with cardinality n, then we define $dec(y)$ to be B (otherwise, $dec(y)$ is not defined). Observe that deciding whether an arbitrary string

z is the prefix of some $enc(H_n)$, with H_n being a dominating set, is a Σ_2^p predicate (where the input of the predicate is $\langle 1^n, z\rangle$). Indeed, z has this property if and only if

$$(\exists w \in \Sigma^{(n^2-|z|)})(\forall y \in \Sigma^n)[dec(zw) \text{ is defined and } y \text{ loses to at least one string in } dec(zw)].$$

Thus, in $\mathrm{FP}^{\Sigma_2^p}$, we can construct, bit by bit, a dominating set using prefix search. Let us denote this set by H_n. Observe that H_n must contain at least one string x in the top Toda class at length n. Thus, all we need to do is to find a string $x \in H_n$ such that for every other string $y \in H_n$ there is a path from x to y in the subgraph of $G_{n,f}$ induced by H_n. Since $||H_n|| = n$, we can do this in polynomial time. For example x can be taken to be any node of maximum outdegree in the subgraph of $G_{n,f}$ induced by H_n (since it is known—see, e.g., [Wes96]—that all the other nodes are reachable from such a node by paths of length at most 2, and so such a node has the desired property). ∎

It is natural to ask if there is a more efficient BestAtLength function. The next result shows that it is unlikely that the BestAtLength function of every commutative P-selector is in FP because this would imply E = NE. Since the proof is relativizable, it follows that there is an oracle relative to which the BestAtLength function for some commutative P-selector is not in FP. The issue of whether the statement of Theorem 3.4 can be improved to $\mathrm{FP}^{\mathrm{NP}}$ remains open.

Theorem 3.5 *1. If* P = NP*, then every commutative* P*-selector f has a BestAtLength function in* FP*.*

2. If every commutative P*-selector f has a BestAtLength function in* FP *then* E = NE*.*

Proof. Part 1 follows immediately from Theorem 3.4. We now prove part 2. Consider a set A that is complete for NE, and let $B = \{1^{1x} \mid x \in A\}$, where 1^{1x} denotes a string of 1's of length the number whose binary encoding is $1x$. Note that B is in NP. Thus there is a nondeterministic polynomial-time machine M and a constant k such that M accepts B and all the computation paths of M on any input of size n have length exactly n^k. We consider the following function $\langle\cdot,\cdot\rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$: If $x = x_1 \ldots x_k$, $x_i \in \Sigma$ for all $i \in \{1, \ldots, k\}$, and y are two strings, then $\langle x, y\rangle = x_1\overline{x_1} \ldots x_k\overline{x_k}00y$, where $\overline{b}$ denotes the bit b flipped (i.e., $\overline{b} = 1-b$). We have specified this (pairing-like, except non-onto) function explicitly because the particular lengths its outputs have on particular inputs will be incorporated below. Let

$$C = \{\langle 1^{1x}, w\rangle \mid 1^{1x} \in B \text{ and } w \text{ is an accepting path of } M \text{ on input } 1^{1x}\}.$$

Clearly C is in P and so it certainly is P-selective via some commutative P-selector f. By hypothesis, f has a BestAtLength function g in FP. Let $x \in \Sigma^*$ and let n be the integer whose binary encoding is $1x$. Then

1. $x \in A \Rightarrow$ there is a string of length $2n + 2 + n^k$ in C of the form

$$\langle 1^{1x}, \text{ accepting path of } M \text{ on } 1^{1x}\rangle,$$

2. $x \notin A \Rightarrow$ there is no string of length $2n + 2 + n^k$ in C.

Thus, we calculate $g(1^{2n+2+n^k})$, which gives us a string of length $2n + 2 + n^k$ having the property that it is in C if C contains any strings of this length. If the output of this calculation is not a string of the form $\langle 1^{1x}, w\rangle$, then it means that x is not in A. If the output of $g(1^{2n+2+n^k})$ is of the form $\langle 1^{1x}, w\rangle$ then we check whether this string is in C. If it is in C, that means by the definition of C that $x \in A$. If it is not in C, then C does not have strings of length $2n + 2 + n^k$, and thus $x \notin A$. These computations take deterministic exponential time in $|x|$. It follows that $A \in \mathrm{E}$, and thus NE = E. ∎

Since there are relativized worlds in which E and NE differ, Theorem 3.5 (in its relativized version, which holds also by the analogous proof) yields the following corollary.

Corollary 3.6 *There is an oracle relative to which there is a commutative* P*-selector f such that no BestAtLength function for f is in* FP*.*

Theorem 3.5 relativizes. That is, for each set B it holds that: If every commutative P^B-selector f has a BestAtLength function in FP^B then $\mathrm{E}^B = \mathrm{NE}^B$. Though neither $\mathrm{NP} \cap \mathrm{coNP}$ nor $\mathrm{coNE} \cap \mathrm{NE}$ is currently known to have complete sets (see [Sip82,Gur83, HI85] regarding $\mathrm{NP} \cap \mathrm{coNP}$), one nonetheless can (by a "set by set" argument—[HHN$^+$95, Corollary 6] for example employs a set-by-set argument in the completely different setting of Karp–Lipton-type results), in light of the facts that $\mathrm{E}^{\mathrm{NP}\cap\mathrm{coNP}} = \mathrm{coNE} \cap \mathrm{NE}$ and $\mathrm{NE} = \mathrm{coNE} \cap \mathrm{NE} \iff \mathrm{NE} = \mathrm{coNE}$, as an application of this see that the following holds.

Corollary 3.7 *If every commutative* $\mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}}$*-selector f has a BestAtLength function in* $\mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}}$ *then* coNE = NE*.*

The reader may naturally wonder whether the E = NE conclusion of part 2 of Theorem 3.5 can be strengthened to a P = NP conclusion. We do not have a definitive answer; the fact that BestAtLength functions have tally inputs seems a difficult impediment to proving this. We do note that it is the only impediment. That is, if we make a new notion of BestAtLength function, let us call it BestBelowUsAtLength, that (for a fixed commutative P-selector function f) takes as its input $\langle 1^n, z\rangle$ and (i) when $|z| \neq n$ outputs "illegal z"; and (ii) when $|z| = n$ and $[z]_f$ is not in the bottom Toda equivalence class (with respect to f at length n) outputs a string w, $|w| = n$, such that $[w]_f$ is the topmost Toda equivalence class (with respect to f at length n) that is less than $[z]_f$ (i.e., such that $[z]_f > [w]_f$ yet, for each $v \in \Sigma^n$, it holds that $[w]_f < [v]_f \Rightarrow [v]_f \geq [z]_f$). (When $|z| = n$ and $[z]_f$ is in the bottom Toda equivalence class (with respect to f at length n), a BestBelowUsAtLength function can output whatever lie it likes.) Under this definition, it is easy to see that a P = NP conclusion holds.

Proposition 3.8 *If every commutative* P*-selector f has a BestBelowUsAtLength function in* FP *then* P = NP*.*

An element x in the top Toda class at length n has the property that for every string $y \in \Sigma^n$ there is a path from x to y in the induced tournament graph. We next investigate a related notion. It is known (this was noted as early as the 1950s [Lan53]) that in a tournament graph there is at least one node v such that for every other node w there is a path of length at most 2 from v to w. This property has played an important role in improving from quasilinear to linear the amount of nondeterminism used to accept the P-selective sets with optimal nonuniform advice ([HNP98], see also [HT96]) and in understanding the complexity of the reachability problem in tournaments [Tan01]. A node v with the above property is called a *king*. A function g is a Find-a-King function for a commutative P-selector f if for all n, on input 1^n, f outputs a king of the graph $G_{f,n}$.

Proposition 3.9 *Every commutative* P*-selector function f has a Find-A-King function computable in* $\mathrm{FP}^{\Sigma_3^p}$.

Proof. Given as input z and 1^n, $|z| \leq n$, note that the string z is a prefix of a king of $G_{n,f}$ if and only if

$$(\exists w \in \Sigma^{n-|z|})(\forall y_1 \in \Sigma^n)(\exists y_2 \in \Sigma^n)[(zw \text{ beats } y_1) \vee ((zw \text{ beats } y_2) \wedge (y_2 \text{ beats } y_1))].$$

It follows that using prefix search we can build a king of $G_{n,f}$ in $\mathrm{FP}^{\Sigma_3^p}$. ∎

It is interesting to note that building a top Toda element seems to be easier than building a king (as indicated by the previous theorems), yet recognizing a top Toda element seems (given our current stage of knowledge) to be a more difficult task than recognizing a king. This holds because a string $x \in \Sigma^n$ is a top Toda element at length n exactly if for any string $y \in \Sigma^n$ there is a path from x to y; checking this condition is in PSPACE. One can observe that this problem is also in the advice class [KL80] PP/ linear (that is it can be done in PP given an advice string of size $O(n)$). We now sketch a proof of this. For an arbitrary P-selector f, let us consider the Toda-equivalence classes for the strings in Σ^n sorted according to the order relation defined above:

$$[x_1]_f > [x_2]_f > \ldots > [x_k]_f.$$

Then any element x in $[x_1]_f$ beats at least the elements in $\{x\} \cup (\bigcup_{2 \leq i \leq k} [x_i]_f)$ and thus has outdegree at least $1 + \sum_{2 \leq i \leq k} ||[x_i]_f||$. Also note that any element (of Σ^n) that is not in $[x_1]_f$ cannot beat any element in $[x_1]_f$ and thus its outdegree is at most $\sum_{2 \leq i \leq k} ||[x_i]_f||$. Thus we will use as the advice string the binary encoding of $1 + \sum_{2 \leq i \leq k} ||[x_i]_f||$ (prefixed as needed with leading zeros to make it hit on the head the exact linear length if one uses the definition of advice that requires that the linear advice be of length exactly that of the specified linear function, rather than just bounded by it, see also the discussion in [HT96, Section 2]; regardless of which of these two different definitional approachs one uses, the class PP/linear remains the same, so this is not a critical issue, and similarly it is not a critical issue for NP/linear). If we check, versus that threshold, the outdegree of whatever node our PP/linear algorithm is interested in, this is a PP computation (given as input the advice and the node), and we can in this fashion check whether the node is in the top Toda class at that length. So, we have membership in PP/linear.

On the other hand, a string x in $G_{n,f}$ is a king node if and only if $(\forall y_1 : |y_1| = |x|)(\exists y_2 : |y_2| = |x|)[(x \text{ beats } y_1) \vee ((x \text{ beats } y_2) \wedge (y_2 \text{ beats } y_1))]$. Thus checking whether a string x is a king in $G_{n,f}$ can be done with a Π_2^p computation. We note that this fact, and also Proposition 3.9, could be shown indirectly using Tantau's recent work on the complexity of succinct tournament reachability [Tan01].

Hemaspaandra et al. [HNOS96a] (without quite explicitly stating it) showed P-sel $\subseteq$ PP/ linear. This was quickly improved to P-sel $\subseteq$ NP/ linear ([HT96], see also [HNP98]). So it is reasonable to wonder whether the above-given PP/ linear result for top Toda element recognition can itself be strengthened to NP/ linear. We conjecture that it cannot; the method that yielded NP/ linear does not seem to apply here in any obvious way. Also, we note in passing that in the profoundly different model in which the tournament—far from being uniformly specified via a P-selector function—can be explored only via queries to a black box, and our input additionally includes the set of nodes inducing via that black box a tournament over which a king (or a certain sequence of kings) is sought, bounds on the numbers of queries to find such have been studied in, for example, [SSW03].

4 P-Selectivity and Ranking

While P-selectivity is an extension of polynomial-time decidability, the notion of polynomial-time weak rankability goes in the opposite direction. It describes sets that are so simple that there exists a polynomial-time algorithm that, on inputs that are elements in the set, outputs the number of elements in the set up to that element. Weak-P-rankability can be an attribute of extremely complex sets. One further generalization is to allow the ranking functions to belong to some broader family of functions (e.g., having some complexity bound less stringent than polynomial time), so that we can understand what computational power is needed to capture cardinality issues for prefixes of sets/subsets from classes of interest (see, e.g., Theorem 4.5). There have been many papers studying the issue of which sets can be ranked [GS91,HR90,BGS91,Huy90].

Definition 4.1 *For any set B and any string x, define $rank_B(x) = ||B^{\leq x}||$.*

1. *[HR90] A set A is* strongly-P-rankable *if there is a polynomial-time computable function f such that $(\forall x \in \Sigma^*)\,[f(x) = rank_A(x)]$. We also use* strongly-P-rankable *to denote the class of all sets that are strongly-*P*-rankable.*

2. *[GS91] A set A is* P-rankable *if there is a polynomial-time computable function f such that (a) $(\forall x \in A)\,[f(x) = rank_A(x)]$ and (b) $(\forall x \notin A)\,[f(x) =$ "not in A"$]$. We also use* P-rankable *to denote the class of all sets that are* P*-rankable.*

3. *[HR90] A set A is* weakly-P-rankable *if there is a polynomial-time computable function f such that $(\forall x \in A)\,[f(x) = rank_A(x)]$. We also use* weakly-P-rankable *to denote the class of all sets that are weakly-*P*-rankable.*

4. *Let $\mathcal{F}$ be a family of functions mapping strings into natural numbers. A set A is* weakly-$\mathcal{F}$-rankable *if there is a function $f \in \mathcal{F}$ such that $(\forall x \in A)\,[f(x) = rank_A(x)]$. We also use* weakly-$\mathcal{F}$-rankable *to denote the class of all sets that are weakly-$\mathcal{F}$-rankable.*

Note that, immediately from the definitions, strongly-P-rankable $\subseteq$ P-rankable $\subseteq$ weakly-P-rankable. (The first inclusion is easy to see in light of the fact that, for each $x \neq \epsilon$, $x \in A \iff rank_A(x) > rank_A(x-1)$, where $x-1$ denotes the immediate lexicographic predecessor of x.) Also note that for $x \notin A$, the definition of weakly-P-rankable sets puts no constraint on the behavior of f on input x other than that f must run in polynomial time. This is a point of similarity with P-selectivity useful for the following refinement of P-selectivity, which has been introduced in [HZZ96]. The refinement adds the requirement that when at least one of the inputs belongs to the set, we output not merely an input that is in the set, but also output its correct ranking information.

Definition 4.2 *[HZZ96] A set A is* polynomial-time semi-rankable *if there is a (total, single-valued) function f such that, for every x and y,*

1. $(\exists n)\,[f(x,y) = \langle x,n\rangle$ *or* $f(x,y) = \langle y,n\rangle]$, *and*
2. $\{x,y\} \bigcap A \neq \emptyset \;\Rightarrow\; [(x \in A$ *and* $f(x,y) = \langle x, rank_A(x)\rangle)$ *or* $(y \in A$ *and* $f(x,y) = \langle y, rank_A(y)\rangle)]$.

In such a case, we say that f is a semi-ranking function for A. We use P-sr *to denote the class of sets that are polynomial-time semi-rankable.*

As noted in [HZZ96], P-sr = P-sel $\cap$ weakly-P-rankable. Among other results, it is shown in [HZZ96] that P-sr is a proper set of P-sel, i.e., there are P-selective sets that are not weakly-P-rankable (there are also sets that are weakly-P-rankable but not P-selective).

For completeness and as an example of how one proves things regarding (non-)weak-P-rankability, we provide a new and short proof of this fact. Let $\{M_i\}_{i\in\mathbb{N}^+}$ be an enumeration of all deterministic Turing machines (viewed as computing single-valued, possibly partial functions). Let HALT $= \{i \mid M_i$ halts on input $i\}$. Of course, HALT is not recursive. For any real $0 \leq \gamma < 1$ the *standard left cut of* γ, denoted $L(\gamma)$, is defined to be

$$L(\gamma) = \{\beta_1\beta_2\cdots\beta_z \mid z \in \mathbb{N} \wedge (\forall j : 1 \leq j \leq z)[\beta_j \in \{0,1\}] \wedge \sum_{1\leq i\leq z} \frac{\beta_i}{2^i} < \gamma\},$$

i.e., it is the strings that when interpreted as binary fractional values are strictly less than γ (see [Sel81,Ko82,Ko83]). Let α, $0 < \alpha < 1$, be the real number whose infinite binary notation is given as follows: The ith bit after the decimal point is 1 if $i \in$ HALT, and is 0 if $i \notin$ HALT. It is known, due to Selman [Sel81], that each standard left cut is P-selective. So, certainly, the left cut $L(\alpha)$ is P-selective. Suppose that $L(\alpha)$ is weakly-P-rankable by some function f. Since $\alpha > 0$ it holds that, for all $j \geq 0$, $0^j \in L(\alpha)$. So $f(0^j)$ gives the correct rank of 0^j in $L(\alpha)$. If we compute, successively, $f(0^j)$ for $j = 0, 1, 2, \ldots$, then we can (keeping in mind that if a standard left cut has, say, k strings at a given length, then those strings will always be the lexicographically first k strings at that length) reconstruct, bit by bit, the binary expansion of α. This implies that HALT is recursive, which is untrue. So our assumption that (the P-selective set) $L(\alpha)$ is weakly-P-rankable is contradicted.

Note that the P-selective set $L(\alpha)$ exhibited above cannot, in fact, be weakly ranked by any recursive function. By modifying the above construction (taking in lieu of HALT an even harder set), one can see that there even are P-selective sets that are not weakly-rankable by any function in the arithmetical hierarchy.

It is shown in [HZZ96] that P-sr has structural properties different from those of P-sel. For example, unlike P-sel, P-sr is not closed under complementation, union with P sets, or join with P sets. It is natural to ponder whether P-sel can be separated from the class of weakly-P-rankable sets or from P-sr in a stronger sense, namely with immunity. We note first that the above approach is hopeless regarding stronger separation from P-sr because $L(0) = \emptyset$ and any standard left cut $L(\gamma)$, $0 < \gamma < 1$, contains the subset $\{0^j \mid j \in \mathbb{N}\}$, which belongs to P-sr. Perhaps somewhat surprisingly it turns out, as we will show as Theorem 4.5, that the statement "P-sel is weakly-P-rankable immune" implies $\mathrm{P} \neq \Sigma_2^p$ (equivalently, $\mathrm{P} \neq \mathrm{NP}$) and thus it seems beyond reach at this time.

A set S if P-printable [HY84] if there exists a polynomial-time algorithm such that, for each $n \in \mathbb{N}$, on input 0^n the algorithm outputs exactly the members of S having length at most n. Note that every P-printable set is sparse and belongs to P. The above definition can be relativized in the standard way.

Lemma 4.3 *Let Q be any set. If A is* P*-selective, S is* P^Q*-printable, and $A \cap S$ is infinite, then $A \cap S$ (and thus also S, and most particularly also A) has an infinite weakly-*FP^Q*-rankable subset.*

Proof. Let A, Q, and S be any sets such that A is P-selective, S is P^Q-printable, and $A \cap S$ is infinite. As noted earlier, we may without loss of generality assume that A has a commutative P-selector. Let E be an FP^Q-algorithm witnessing that S is FP^Q-printable. Let $B = A \cap S$. Clearly, $B \subseteq A$ and $B \subseteq S$. We consider the following two cases.

Case 1: There is a string y in B such that there are infinitely many elements of B that beat y.

We take y_0 to be the lexicographically first such string and define

$$C = \{x \in B \mid \ x >_{lex} y_0 \wedge \ x \text{ beats } y_0\}.$$

$C \subseteq B$, so $C \subseteq A$. The set C is weakly-FP^Q-rankable. This is true because of the following procedure: Note that

$$rank_C(x) = ||\{w \mid \ y_0 <_{lex} w \leq_{lex} x \wedge w \in S \wedge w \text{ beats } y_0\}||.$$

(To see this, it may help to recall that, since $y_0 \in A$, any w that beats y_0 is certainly a member of A; and if additionally $w \in S$, then $w \in A \cap S = B$.) To compute $rank_C(x)$ we run E on input $0^{|x|}$, so E outputs $S \cap \Sigma^{\leq |x|}$. From $S \cap \Sigma^{\leq |x|}$, we keep only those strings w that both beat y_0 and satisfy $y_0 <_{lex} w \leq_{lex} x$, and we count the number of such strings. This is the desired rank. This can be done in FP^Q, so C is weakly-FP^Q-rankable. (In fact, the procedure just given even shows that, in this case, C is strongly-FP^Q-rankable.)

Case 2: It is not the case that there is a string y in B such that there are infinitely many elements of B that beat y.

We build the weakly-FP^Q-rankable set C, $C \subseteq B \subseteq A$, as follows. First let y_0 be the lexicographically first element in B. We insert y_0 in C. Then, inductively for each $i \geq 0$, we let y_{i+1} be the lexicographically least string such that

(1) $y_{i+1} \in B$,

(2) y_{i+1} loses to each of $y_i, y_{i-1}, \ldots, y_0$, and

(3) $y_{i+1} >_{lex} y_i$.

Since each of $y_0, \ldots, y_i$ loses to only finitely many strings in B, there exists such a string y_{i+1}. We insert y_{i+1} in C.

We show next that, as claimed, C is weakly-FP^Q-rankable. Let z be a string for which we wish to assert a rank (in C) that will be correct if $z \in C$. (For strings $z \notin C$, our only constraint is not to ruin the overall required polynomial running time.) If z belongs to C, there must be an index i such that $z = y_i$ and the rank of z in C will be $i+1$. Thus, we seek to determine i. Observe that

$$y_0 <_{lex} y_1 <_{lex} \cdots <_{lex} y_i$$

and

$$y_0 >_f y_1 >_f \cdots >_f y_{i-1} >_f y_i.$$

So, if $z \in C$, we have $y_i = z \in C \subseteq B$ and so (still assuming $z \in C$) it follows that $y_1, y_2, \ldots, y_{i-1}$ are all in B. If $i > 1$, it can be seen that (still assuming $z \in C$) y_1 is the shortest string in S that

(a) is, with respect to lexicographic order, greater than y_0 and less than z (which if $z \in C$ satisfies $z = y_i$),

(b) loses to y_0, and

(c) wins at z (which if $z \in C$ equals y_i).

Thus, in time polynomial in $|z|$, using the FP^Q-printing algorithm E (to let us check membership in S), we can ourselves via an FP^Q-procedure find y_1. (Note that the true y_0 will be hardcoded into our algorithm.) Then inductively, for $j = 2, \ldots, i-1$, it can be checked that (still assuming $y_i = z$) y_j is the shortest string in S that

(a) is, in lexicographic order, greater than y_{j-1} and less than z,

(b) loses to $y_0, \ldots, y_{j-1}$, and

(c) wins at z.

So with an FP^Q computation we can (still assuming $z \in C$) determine one by one the strings $y_2, \ldots, y_{i-1}$ (going until the above process stops producing y_j's), and thus determine i. If $z \notin C$, the i value we compute by doing the above is nonsense (because z is not a y_i and so the construction in this case was flawed in its belief that beating z is a proof of membership), but that is not a problem; our process is, overall, an FP^Q process. ∎

We feel that the construction in Case 2 above is a novel one. The following results immediately follow from Lemma 4.3.

Theorem 4.4 P-sel *is not bi-immune to the class of weakly-*P*-rankable sets.*

Proof. Let A be an arbitrary P-selective set and let $S = 0^*$. Clearly, S is P-printable. By Lemma 4.3, if $A \cap S$ is infinite it has an infinite weakly P-rankable subset (and thus so does A). By Lemma 4.3, if $\overline{A} \cap S$ is infinite it has an infinite weakly P-rankable subset (and thus so does $\overline{A}$). However, at least one of $A \cap S$ and $\overline{A} \cap S$ is infinite, so we are done. ∎

Theorem 4.5 *Any infinite* P*-selective set A has an infinite weakly-*$\mathrm{FP}^{\Sigma_2^p}$*-rankable subset.*

Proof. Let g be the $\mathrm{FP}^{\Sigma_2^p}$-computable function given by Theorem 3.4, which on input 1^n outputs some string of length n in the top Toda class of A at length n (recall that, like all P-selective sets, A has a commutative P-selector, and thus Theorem 3.4 can be applied). For any $n \in \mathbb{N}^+$, let $x_n = g(1^n)$. Let $S = \{x_1, x_2, x_3, \ldots\}$. Then S is $\mathrm{FP}^{\Sigma_2^p}$-printable. Since A is infinite, for infinitely many m, $A^{=m} \neq \emptyset$. For each $m \geq 1$ such that $A^{=m} \neq \emptyset$, $x_m \in A$. So, $A \cap S$ is infinite. Now, by Lemma 4.3, it follows that A has weakly-$\mathrm{FP}^{\Sigma_2^p}$-rankable subset. ∎

Corollary 4.6 *If* $\mathrm{P} = \mathrm{NP}$*, then* P-sel *is not immune to the class of weakly-*P*-rankable sets.*

The reader may wish to compare Theorem 4.5 with the following result—neither of which seems to imply the other—from [HHN04] regarding printability [HY84]: Each infinite P-selective set B has an infinite $\mathrm{FP}^{B \oplus \Sigma_2^p}$-printable subset.

An important subclass of P-sel is the class of sets that are standard left cuts, a class that we have already used. Recall that for each real $0 \leq \gamma < 1$ the standard left cut of γ is the set $L(\gamma) = \{\beta_1\beta_2 \cdots \beta_z \mid z \in \mathbb{N} \wedge (\forall j : 1 \leq j \leq z)[\beta_j \in \{0,1\}] \wedge \sum_{1 \leq i \leq z} \frac{\beta_i}{2^i} < \gamma\}$. All the P-selective sets that have been constructed in the literature are either standard left cuts or are $\leq_m^p$-equivalent to a standard left cut and, in fact, proving that there is a P-selective set that is not $\leq_m^p$-equivalent to a standard left cut is known to be as hard as showing $\mathrm{P} \neq \mathrm{PP}$ [HNOS96a]. In contrast, we observe that standard left cuts that are weakly-rankable are always in P.

Theorem 4.7 *If A is a standard left cut, then A is weakly-*P*-rankable if and only if A is strongly-*P*-rankable.*

Proof. Let A be an arbitrary standard left cut. In particular, let γ, $0 \leq \gamma < 1$, be such that $A = L(\gamma)$. Assume that A is weakly-P-rankable, and let r be a polynomial-time function that weakly ranks A (in the sense of having the properties of the function f of part 3 of Definition 4.1). We seek to show that A is strongly-P-rankable.

If $\gamma = 0$, then $L(\gamma) = \emptyset$, so $L(\gamma)$ indeed is strongly-P-rankable, via the function that always outputs zero. So we now consider only the case $0 < \gamma < 1$. We describe, for this case, a strong-P-ranking function for A. Our strong-P-ranking function will do the following. On input z, compute

$$u(z) = \sum_{0 \leq j \leq |z|-1} m_j,$$

where $m_j = r(0^{j+1}) - r(0^j)$. Also compute $v(z) = r(0^{|z|+1}) - r(0^{|z|})$. Note—crucially to our algorithm's correctness—that, since $0 < \gamma$, each string of the form 0^i belongs to $L(\gamma)$, and so the weak-P-ranking function r does not lie on strings of the form 0^i.

(Recall that if a standard left cut has, say, k strings at a given length, then those strings will always be the lexicographically first k strings at that length.) If z is one of the lexicographically first $v(z)$ strings of length $|z|$, recalling Definition 4.1 (regarding the definition of the function "rank"), output $u(z) + rank_{\Sigma^{|z|}}(z)$, e.g., if $z = 0^{|z|}$, output $u(z)+1$. Otherwise (i.e., if z is *not* one of the lexicographically first $v(z)$ strings of length $|z|$), output $u(z) + v(z)$. Note that this function is a strong-P-ranker for A since it in effect correctly counts, at each length starting at zero, the number of strings in A that are lexicographically less than or equal to z. ∎

Since strongly-P-rankable $\subseteq$ P-rankable $=$ P $\cap$ weakly-P-rankable, we have the following immediate corollary.

Corollary 4.8 *Each weakly-*P*-rankable standard left cut belongs to* P*.*

Acknowledgments: We are grateful to A. Kaplan and B. Serog for helpful conversations, to R. Tripathi for proofreading, and to the anonymous LATIN '04 and *Theory of Computing Systems* referees for helpful comments.

References

[BGS91] A. Bertoni, M. Goldwurm, and N. Sabadini. The complexity of computing the number of strings of given length in context-free languages. *Theoretical Computer Science*, 86(2):325–342, 1991.

[BS85] J. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory*, 18:1–10, 1985.

[DHHT94] D. Denny-Brown, Y. Han, L. Hemaspaandra, and L. Torenvliet. Semimembership algorithms: Some recent advances. *SIGACT News*, 25(3):12–23, 1994.

[GS91] A. Goldberg and M. Sipser. Compression and ranking. *SIAM Journal on Computing*, 20(3):524–536, 1991.

[Gur83] Y. Gurevich. Algebras of feasible functions. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 210–214. IEEE Computer Society Press, November 1983.

[HHN+95] L. Hemaspaandra, A. Hoene, A. Naik, M. Ogiwara, A. Selman, T. Thierauf, and J. Wang. Nondeterministically selective sets. *International Journal of Foundations of Computer Science*, 6(4):403–416, 1995.

[HHN04] L. Hemaspaandra, H. Hempel, and A. Nickelsen. Algebraic properties for selector functions. *SIAM Journal on Computing*, 33(6):1309–1337, 2004.

[HI85] J. Hartmanis and N. Immerman. On complete problems for $\mathrm{NP} \bigcap \mathrm{coNP}$. In *Proceedings of the 12th International Colloquium on Automata, Languages, and Programming*, pages 250–259. Springer-Verlag *Lecture Notes in Computer Science #194*, July 1985.

[HNOS96a] E. Hemaspaandra, A. Naik, M. Ogihara, and A. Selman. P-selective sets and reducing search to decision vs. self-reducibility. *Journal of Computer and System Sciences*, 53(2):194–209, 1996.

[HNOS96b] L. Hemaspaandra, A. Naik, M. Ogihara, and A. Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25(4):697–708, 1996.

[HNP98] L. Hemaspaandra, C. Nasipak, and K. Parkins. A note on linear-nondeterminism, linear-sized, Karp–Lipton advice for the P-selective sets. *Journal of Universal Computer Science*, 4(8):670–674, 1998.

[HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Springer-Verlag, 2002.

[HOW02] L. Hemaspaandra, M. Ogihara, and G. Wechsung. Reducing the number of solutions of NP functions. *Journal of Computer and System Sciences*, 64(2):311–328, 2002.

[HR90] L. Hemachandra and S. Rudich. On the complexity of ranking. *Journal of Computer and System Sciences*, 41(2):251–271, 1990.

[HT96] L. Hemaspaandra and L. Torenvliet. Optimal advice. *Theoretical Computer Science*, 154(2):367–377, 1996.

[HT03] L. Hemaspaandra and L. Torenvliet. *Theory of Semi-Feasible Algorithms*. Springer-Verlag, 2003.

[Huy90] D. Huynh. The complexity of ranking simple languages. *Mathematical Systems Theory*, 23(1):1–20, 1990.

[HY84] J. Hartmanis and Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science*, 34(1–2):17–32, 1984.

[HZZ96] L. Hemaspaandra, M. Zaki, and M. Zimand. Polynomial-time semi-rankable sets. In *Journal of Computing and Information*, 2(1), Special Issue: *Proceedings of the 8th International Conference on Computing and Information*, pages 50–67, 1996. CD-ROM ISSN 1201-8511/V2/#1.

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

[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. ACM Press, April 1980. An extended version has also appeared as: Turing machines that take advice, *L'Enseignement Mathématique*, 2nd series, 28:191–209, 1982.

[Ko82] K. Ko. The maximum value problem and NP real numbers. *Journal of Computer and System Sciences*, 24(1):15–35, 1982.

[Ko83] K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26(2):209–221, 1983.

[Lan53] H. Landau. On dominance relations and the structure of animal societies, III: The condition for score structure. *Bulletin of Mathematical Biophysics*, 15(2):143–148, 1953.

[Nic01] A. Nickelsen. Polynomial-time partial information classes. Wissenschaft und Technik Verlag, 2001. Also Ph.D. thesis, Technische Universität Berlin, Berlin, Germany, 1999.

[NRRS98] A. Naik, J. Rogers, J. Royer, and A. Selman. A hierarchy based on output multiplicity. *Theoretical Computer Science*, 207(1):131–157, 1998.

[NT03] A. Nickelsen and T. Tantau. Partial information classes. *SIGACT News*, 34(1), 2003.

[Ogi96] M. Ogihara. Functions computable with limited access to NP. *Information Processing Letters*, 58(1):35–38, 1996.

[Sel79] A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory*, 13(1):55–65, 1979.

[Sel81] A. Selman. Some observations on NP real numbers and P-selective sets. *Journal of Computer and System Sciences*, 23(3):326–332, 1981.

[Sel82a] A. Selman. Analogues of semirecursive sets and effective reducibilities to the study of NP complexity. *Information and Control*, 52(1):36–51, 1982.

[Sel82b] A. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19(3):287–304, 1982.

[Sip82] M. Sipser. On relativization and the existence of complete sets. In *Proceedings of the 9th International Colloquium on Automata, Languages, and Programming*, pages 523–531. Springer-Verlag *Lecture Notes in Computer Science #140*, July 1982.

[SSW03] J. Shen, L. Sheng, and J. Wu. Searching for sorted sequences in tournaments. *SIAM Journal on Computing*, 32(5):1201–1209, 2003.

[Tan01] T. Tantau. A note on the complexity of the reachability problem for tournaments. Technical Report TR01-092, Electronic Colloquium on Computational Complexity, http://www.eccc.uni-trier.de/eccc/, November 2001.

[Tod91] S. Toda. On polynomial-time truth-table reducibilities of intractable sets to P-selective sets. *Mathematical Systems Theory*, 24(2):69–82, 1991.

[Wes96] D. West. *Introduction to Graph Theory*. Prentice-Hall, 1996.