Locating P/poly Optimally in the Extended Low Hierarchy

J. Köbler
Abteilung für Theoretische Informatik
Universität Ulm
D-89069 Ulm, Germany

Abstract

The low hierarchy within NP and the extended low hierarchy have turned out to be very useful in classifying many interesting language classes. We relocate P/poly from the third Σ-level $\mathrm{EL}_3^{\mathrm{P},\Sigma}$ (Balcázar et al., 1986) to the third Θ-level $\mathrm{EL}_3^{\mathrm{P},\Theta}$ of the extended low hierarchy. The location of P/poly in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ is optimal since, as shown by Allender and Hemachandra (1992), there exist sparse sets that are not contained in the next lower level $\mathrm{EL}_2^{\mathrm{P},\Sigma}$. As a consequence of our result, all NP sets in P/poly are relocated from the third Σ-level $\mathrm{L}_3^{\mathrm{P},\Sigma}$ (Ko and Schöning, 1985) to the third Θ-level $\mathrm{L}_3^{\mathrm{P},\Theta}$ of the low hierarchy.

1 Introduction

Based on ideas from recursive function theory, Schöning [Sc83] introduced the low and high hierarchies inside NP which turned out to be very useful in classifying decision problems in NP not known to be NP-complete or in P. In order to characterize the complexity of language classes not contained in NP, this idea was extended by Balcázar, Book, and Schöning [BBS86] who defined the extended low and high hierarchies (for definitions see Section 2).

Intuitively, the low and high hierarchies classify sets according to how much information they provide for the various levels of the polynomial-time hierarchy, when used as an oracle. The polynomial-time hierarchy, introduced by Meyer and Stockmeyer [MS73, St77], has been recently extended to the Θ-levels by Wagner [Wag90] who demonstrated their robustness and importance by establishing a variety of characterizations for them. The low and high hierarchies, originally introduced only on the base of the Σ-levels, were subsequently refined to include also the levels corresponding to the Δ and Θ-levels of the polynomial-time hierarchy [Sc86a, AH92, LS91]. Very recently, Sheu and Long [SL92] proved that the extended low hierarchy is an infinite hierarchy.

All sets that could be located within the low hierarchies roughly fall into two categories. Either they have *low complexity*, i.e., they are "close" to the class P, as for example the primality [Pr75] and the graph isomorphism [Sc88] problems, or they have *low information content* like sparse sets or sets reducible to (or equivalent to) sparse or tally sets via different kinds of reducibilities. In the past, a variety of language classes has been shown to be included in the low hierarchies (for an overview see for example [Sc86a, AH92, LS91, HOW92, AKM]). Allender and Hemachandra [AH92], and Long and Sheu [LS91] proved the optimality of the location of almost all these classes, at least in some relativized world. However, until now, the exact location of sets having polynomial size circuits remained open.

In research from the early 1980's to the present, there has been considerable interest in the question of whether intractable sets possibly have polynomial size circuits. For example, Karp and Lipton (together with Sipser) [KL80] proved that no NP-complete set has polynomial size circuits unless the polynomial time hierarchy collapses to its second level. In their fundamental work, Karp and Lipton give useful characterizations of various nonuniform classes in terms of advice classes. In this framework, sets having polynomial size circuits are precisely the sets in P/poly, i.e., they are decidable in polynomial time with the help of a polynomial length bounded advice function [Pi79]. Furthermore, the class P/poly can be equivalently characterized as the class of sets that are polynomial-time Turing reducible to some sparse set (due to A. Meyer [BH77]).

Ko and Schöning [KS85] showed that all NP sets in P/poly are contained in the $\mathrm{L}_3^{\mathrm{P},\Sigma}$ level of the low hierarchy. This result is incomparable to the result of Karp, Lipton and Sipser mentioned above showing that all self-reducible NP sets in P/poly are contained in $\mathrm{L}_2^{\mathrm{P},\Sigma}$. Later on, Balcázar, Book, and Schöning [BBS86] located all of P/poly in the $\mathrm{EL}_3^{\mathrm{P},\Sigma}$ level of the extended low hierarchy.

In the present paper, we relocate P/poly from $\mathrm{EL}_3^{\mathrm{P},\Sigma}$ two levels lower in the third Θ-level $\mathrm{EL}_3^{\mathrm{P},\Theta}$. Since there exists a sparse set that is not contained in $\mathrm{EL}_2^{\mathrm{P},\Sigma}$ [AH92], P/poly is not contained in the next lower level of the extended low hierarchy. The $\mathrm{EL}_2^{\mathrm{P},\Sigma}$ lower bound for sparse sets was recently improved by Long and Sheu [LS91] (see Section 5). These lower bounds indicate that the location of P/poly in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ is optimal. Furthermore, our proof of the containment of P/poly in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ actually shows that all sets in the class $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$[1] are in $\mathrm{EL}_3^{\mathrm{P},\Theta}$. As a consequence, all NP sets in $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ are located in the third Θ-level $\mathrm{L}_3^{\mathrm{P},\Theta}$ of the low hierarchy. Hence there exists no $\leq_T^{SN}$-complete set for NP in $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ unless the polynomial-time hierarchy collapses to Θ_3^{P}.

Of course, it is not possible to obtain absolute lower bounds on the location of

[1]Though the $\mathrm{EL}_3^{\mathrm{P},\Sigma}$-lowness of P/poly [BBS86] can be extended to $(\mathrm{NP}/\mathrm{poly}) \cap (\text{co-NP}/\mathrm{poly})$ [He93], it remains open whether this superclass of $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ (cf. [GB91]) can also be relocated in $\mathrm{EL}_3^{\mathrm{P},\Theta}$.

classes in the low hierarchy unless we are able to prove that $\mathrm{P} \neq \mathrm{NP}$. The best known relativized lower bound on the location of $\mathrm{NP} \cap (\mathrm{P/poly})$ in the low hierarchy is due to Long and Sheu who constructed a recursive oracle set relative to which some co-sparse set in NP is not contained in $\mathrm{L}_2^{\mathrm{P},\Delta}$ [LS91].

Our proof showing that P/poly is contained in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ is based on a new upper bound on the complexity of computing small descriptions (that is, small circuits, or more generally, correct advice) for sets in P/poly. Gavaldà and Watanabe [GW91] observed that the techniques in [KS85, Sc86a] show that for every set $A \in \mathrm{P/poly}$ polynomial size circuits can be computed in $\mathrm{F}\Delta_3^{\mathrm{P}}(A)$. Recently, Gavaldà [Ga92b] proved an incomparable upper bound, namely $\mathrm{FP}(\mathrm{NP}(A) \oplus \Sigma_3^{\mathrm{P}})$, thus decreasing the power needed to access the relevant information in A on the cost of increasing the complexity of the unrelativized part of the computation. In his proof, Gavaldà applied Angluin's "majority vote strategy" [An88] to guide the search for correct advice, and he used and extended Sipser's hashing technique [Si83, St85] to estimate the number of advice strings with certain properties. Building on the ideas developed by Gavaldà, we unify the $\mathrm{F}\Delta_3^{\mathrm{P}}(A)$ and the $\mathrm{FP}(\mathrm{NP}(A) \oplus \Sigma_3^{\mathrm{P}})$ upper bounds to $\mathrm{FP}(\mathrm{NP}(A) \oplus \Sigma_2^{\mathrm{P}})$.

By refining the techniques used to obtain our new complexity bound on *computing* correct advice and by applying a census argument, we show in a further step that for every set A in P/poly advice can be *checked* by an $\mathrm{NP}(\mathrm{NP} \oplus A)$ algorithm that gets the advice of a suitable $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$ function. Based on this upper bound, the $\mathrm{EL}_3^{\mathrm{P},\Theta}$-lowness of P/poly is easily obtained by applying an oracle replacement technique.

The paper is organized as follows. Section 2 introduces notation and gives basic definitions. Furthermore, we review some lowness results related to our work.

In Section 3, we prove the new complexity bound for computing correct advice. From this result, we derive also a new upper bound for the complexity of tally set descriptions for languages in P/poly.

In Section 4, we prove as the main result of the paper that P/poly and $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ are located in the $\mathrm{EL}_3^{\mathrm{P},\Theta}$ level of the extended low hierarchy.

Finally, in Section 5, we argue that this placement is optimal by showing that the class of sparse sets is not contained in any reasonable level below $\mathrm{EL}_3^{\mathrm{P},\Theta}$.

2 Preliminaries and notation

All languages are over the binary alphabet $\Sigma = \{0, 1\}$. The *length* of a string $x \in \Sigma^*$ is denoted by $|x|$. For a language A, let $A^{=n}$ $(A^{\leq n})$ denote the set of all strings in A of length n (up to length n, respectively). The *characteristic function* of A is defined as $A(x) = 1$ if $x \in A$, and $A(x) = 0$, otherwise. The cardinality of a finite set A is denoted by $|A|$. For a class $\mathcal{C}$ of sets, co-$\mathcal{C}$ denotes the class $\{\Sigma^* - A \mid A \in \mathcal{C}\}$ of complements of the sets in $\mathcal{C}$.

A set T is called *tally* if $T \subseteq 0^*$. A set S is called *sparse* if the cardinality of

$S^{\le n}$ is bounded above by a polynomial in n. TALLY denotes the class of all tally sets, and SPARSE denotes the class of all sparse sets. The *join* of two sets A and B is $A \oplus B = \{0x \mid x \in A\} \cup \{1x \mid x \in B\}$. The join of language classes is defined analogously. To encode pairs (or tuples) of strings we use a standard polynomial-time computable pairing function denoted by $\langle \cdot, \cdot \rangle$ whose inverses are also computable in polynomial time. $\mathcal{N}$ denotes the set of non-negative integers.

We assume that the reader is familiar with fundamental complexity theoretic concepts such as (oracle) Turing machines and the polynomial-time hierarchy (see for example [BDG, Sc86a]).

The class #P was introduced by Valiant [Va79] and contains for every decision problem in NP the corresponding counting version (i.e. the problem to determine the number of different solutions for a given instance x). Formally, a function f is in #P if there exists a polynomial-time nondeterministic Turing machine M such that M on input x has exactly $f(x)$ many accepting computation paths.

Karp and Lipton [KL80] introduced the notion of advice functions in order to provide a general framework to characterize nonuniform complexity classes. For a class $\mathcal{C}$ of sets and a class $\mathcal{F}$ of functions from 0^* to Σ^*, let $\mathcal{C}/\mathcal{F}$ be the class of sets A such that there is a set $B \in \mathcal{C}$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$,

$$x \in A \Leftrightarrow \langle x, h(0^{|x|}) \rangle \in B.$$

The function h is called an *advice function* for A, and B is the corresponding *interpreter set*. Note that the advice string $h(0^{|x|})$ which is given to B along with x depends only on the length of x. Intuitively, $\mathcal{C}/\mathcal{F}$ is the class of sets that can be recognized given the power captured by the complexity class $\mathcal{C}$ plus an amount of advice provided by some function in $\mathcal{F}$. One of the best studied nonuniform complexity classes is P/poly where poly contains all functions h such that $|h(0^n)| \le p(n)$ for some fixed polynomial p. By imposing a complexity bound on the advice function class $\mathcal{F}$, the $\mathcal{C}/\mathcal{F}$ notation can also be used to characterize uniform complexity classes (see, e.g., [Kä91, KT94]).

The reducibilities discussed in this paper are the standard polynomial-time reducibilities defined by Ladner, Lynch, and Selman [LLS75], and the following one, denoted by $\le_T^{SN}$. We write $A \le_T^{SN} B$ if $A \in \mathrm{NP}(B) \cap \text{co-NP}(B)$. Relation $\le_T^{SN}$ is called strong nondeterministic polynomial-time Turing reducibility and was introduced by Long [Lo82].

For an oracle set A, $\mathrm{P}(A)$ (resp., $\mathrm{NP}(A)$) is the class of all languages $L(M, A)$ accepted by some deterministic (resp., nondeterministic) polynomial-time oracle machine M relative to A. We write $L \in \mathrm{NP}_1(A)$ if M asks at most one query on each nondeterministic computation path. If additionally M accepts if and only if the answer is 'yes' (resp., 'no') then we write $L \in \mathrm{NP}_{\mathrm{m}}(A)$ (resp., $L \in \mathrm{NP}_{\overline{\mathrm{m}}}(A)$).

An oracle machine M is called *nonadaptive* if M computes a list of all queries before actually asking any query to the oracle; otherwise M is *adaptive*. The class of languages (respectively, functions) computed by some nonadaptive deterministic polynomial-time oracle machine with an oracle from some class $\mathcal{C}$ is denoted by $\mathrm{P}_{tt}(\mathcal{C})$ (respectively, $\mathrm{FP}_{tt}(\mathcal{C})$).

A deterministic polynomial-time oracle machine (resp., oracle transducer) that asks on inputs of length n at most $O(\log n)$ adaptive queries to its oracle is called a Θ-machine (resp., Θ-transducer). For a set A, the class $\Theta(A)$ [LS91] contains all languages $L(M, A)$ accepted by some Θ-machine M using oracle A. The Σ, Δ and Θ-levels of the (relativized) polynomial-time hierarchy [MS73, St77, LS91, Wag90] are defined as follows: $\Sigma_0(A) = \Delta_0(A) = \Theta_0(A) = \mathrm{P}(A)$, and for $k \geq 1$, $\Sigma_k(A) = \mathrm{NP}(\Sigma_{k-1}(A))$, $\Delta_k(A) = \mathrm{P}((\Sigma_{k-1}(A))$, $\Theta_k^{\mathrm{P}}(A) = \Theta(\Sigma_{k-1}^{\mathrm{P}}(A))$. Similarly, for $k \geq 1$, $\mathrm{F}\Delta_k^{\mathrm{P}}(A) = \mathrm{FP}(\Sigma_{k-1}^{\mathrm{P}}(A))$ and $\mathrm{F}\Theta_k^{\mathrm{P}}(A) = \mathrm{F}\Theta(\Sigma_{k-1}^{\mathrm{P}}(A))$, where $\mathrm{FP}(A)$ (resp., $\mathrm{F}\Theta(A)$) is the class containing all functions computable by some polynomial-time transducer (resp., Θ-transducer) under oracle A.

Schöning [Sc83, Sc86a] defined the Σ and Δ-levels[2] of the low hierarchy inside NP. Recently, the low hierarchy was refined by Long and Sheu [LS91] based on the Θ-levels of the polynomial-time hierarchy.

Definition 2.1 **[Sc83, Sc86a, LS91]** *The Σ, Δ, and Θ-levels of the low hierarchy (denoted $\mathrm{L}_k^{\mathrm{P},\Sigma}$, $\mathrm{L}_k^{\mathrm{P},\Delta}$, and $\mathrm{L}_k^{\mathrm{P},\Theta}$, respectively), and of the high hierarchy (denoted $\mathrm{H}_k^{\mathrm{P},\Sigma}$, $\mathrm{H}_k^{\mathrm{P},\Delta}$, and $\mathrm{H}_k^{\mathrm{P},\Theta}$, respectively) are defined as follows,*

a) $\mathrm{L}_k^{\mathrm{P},\Sigma} = \{A \in \mathrm{NP} \mid \Sigma_k^{\mathrm{P}}(A) \subseteq \Sigma_k^{\mathrm{P}}\}$, $\mathrm{H}_k^{\mathrm{P},\Sigma} = \{A \in \mathrm{NP} \mid \Sigma_{k+1}^{\mathrm{P}} \subseteq \Sigma_k^{\mathrm{P}}(A)\}$, $k \geq 0$,

b) $\mathrm{L}_k^{\mathrm{P},\Delta} = \{A \in \mathrm{NP} \mid \Delta_k^{\mathrm{P}}(A) \subseteq \Delta_k^{\mathrm{P}}\}$, $\mathrm{H}_k^{\mathrm{P},\Delta} = \{A \in \mathrm{NP} \mid \Delta_{k+1}^{\mathrm{P}} \subseteq \Delta_k^{\mathrm{P}}(A)\}$, $k \geq 1$,

c) $\mathrm{L}_k^{\mathrm{P},\Theta} = \{A \in \mathrm{NP} \mid \Theta_k^{\mathrm{P}}(A) \subseteq \Theta_k^{\mathrm{P}}\}$, $\mathrm{H}_k^{\mathrm{P},\Theta} = \{A \in \mathrm{NP} \mid \Theta_{k+1}^{\mathrm{P}} \subseteq \Theta_k^{\mathrm{P}}(A)\}$, $k \geq 1$.

Thus, a set $A \in \mathrm{NP}$ is in the low hierarchy if A, when used as an oracle, does not provide any additional power to one of the levels of the polynomial-time hierarchy. In contrast, if A provides some level of the polynomial-time hierarchy with the whole power of the class NP, then A belongs to the high hierarchy. The most important fact about sets located in the low hierarchy is that they cannot be at the same time in any level of the high hierarchy (and therefore they cannot be NP complete under any reasonable reducibility) unless the polynomial-time hierarchy collapses. The following proposition lists some basic properties of the low and high hierarchies inside NP.

Proposition 2.2 **[Sc83, Sc86a, LS91]**

i) $\mathrm{L}_0^{\mathrm{P},\Sigma} = \mathrm{L}_1^{\mathrm{P},\Theta} = \mathrm{L}_1^{\mathrm{P},\Delta} = \mathrm{P}$,

ii) $\mathrm{L}_1^{\mathrm{P},\Sigma} = \mathrm{NP} \cap \text{co-NP}$,

[2]Following Long and Sheu, we use the notation $\mathrm{L}_k^{\mathrm{P},\Sigma}$, $\mathrm{L}_k^{\mathrm{P},\Delta}$, $\mathrm{H}_k^{\mathrm{P},\Sigma}$, $\mathrm{H}_k^{\mathrm{P},\Delta}$, $\mathrm{EL}_k^{\mathrm{P},\Sigma}$, and $\mathrm{EL}_k^{\mathrm{P},\Delta}$ instead of the original notation [Sc83, Sc86a, BBS86, AH92] $\mathrm{L}_k^{\mathrm{P}}$, $\widehat{\mathrm{L}}_k^{\mathrm{P}}$, $\mathrm{H}_k^{\mathrm{P}}$, $\widehat{\mathrm{H}}_k^{\mathrm{P}}$, EL_k, and $\widehat{\mathrm{EL}}_k$ for the Σ and Δ-levels of the low, high, and extended low hierarchies, respectively.

iii) *For each* $k \geq 1$, $\mathrm{L}_{k-1}^{\mathrm{P},\Sigma} \subseteq \mathrm{L}_k^{\mathrm{P},\Theta} \subseteq \mathrm{L}_k^{\mathrm{P},\Delta} \subseteq \mathrm{L}_k^{\mathrm{P},\Sigma}$,

iv) $\mathrm{H}_0^{\mathrm{P},\Sigma} = \mathrm{H}_1^{\mathrm{P},\Theta} = \mathrm{H}_1^{\mathrm{P},\Delta} = \{A \mid A \textit{ is } \leq_T^P\textit{-complete for } \mathrm{NP}\}$,

v) $\mathrm{H}_1^{\mathrm{P},\Sigma} = \{A \mid A \textit{ is } \leq_T^{SN}\textit{-complete for } \mathrm{NP}\}$,

vi) *For each* $k \geq 1$, $\mathrm{H}_{k-1}^{\mathrm{P},\Sigma} \subseteq \mathrm{H}_k^{\mathrm{P},\Theta} \subseteq \mathrm{H}_k^{\mathrm{P},\Delta} \subseteq \mathrm{H}_k^{\mathrm{P},\Sigma}$.

As shown in [Ka89, LS91], all tally and all sparse sets in NP are contained in $\mathrm{L}_2^{\mathrm{P},\Theta}$. Since $\mathrm{L}_1^{\mathrm{P},\Sigma} = \mathrm{NP} \cap \text{co-NP}$, the question whether all tally sets in NP are already contained in the next lower level $\mathrm{L}_1^{\mathrm{P},\Sigma}$ is equivalent to the open problem whether the class NE ($= \mathrm{NTIME}(2^{O(n)})$) is closed under complementation [Bo74]. By the recent result of Buhrman, Longpré, and Spaan that every sparse set conjunctively reduces to a tally set [BLS93], it easily follows that there exist tally sets in NP$-$co-NP if and only if there exist sparse sets in NP $-$ co-NP (see also [HY84]). As a consequence, the existence of sparse sets in $\mathrm{L}_2^{\mathrm{P},\Theta} - \mathrm{L}_1^{\mathrm{P},\Sigma}$ is equivalent to the existence of tally sets in $\mathrm{L}_2^{\mathrm{P},\Theta} - \mathrm{L}_1^{\mathrm{P},\Sigma}$ [LS91]. On the other hand, the class of co-sparse NP sets is only known to be located in $\mathrm{L}_3^{\mathrm{P},\Theta}$, and there exists an oracle relative to which some co-sparse NP set is not contained in $\mathrm{L}_2^{\mathrm{P},\Delta}$ [LS91]. The question whether there exists an oracle relative to which some NP set in P/poly is not contained in $\mathrm{L}_2^{\mathrm{P},\Sigma}$, is open.

In order to classify language classes not contained in NP (e.g. sets reducible to sparse or tally sets), the low and high hierarchies have been extended by Balcázar, Book, and Schöning [BBS86] who defined the Σ-levels of the extended low and high hierarchies. Later on, Allender and Hemachandra [AH92], and Long and Sheu [LS91] refined the extended low and high hierarchies by introducing intermediate levels based on the Δ and Θ-levels of the polynomial-time hierarchy, respectively. Very recently, Sheu and Long [SL92] proved that the extended low hierarchy is an infinite hierarchy (see part *ii)* of Proposition 2.4 below).

Definition 2.3 [BBS86, AH92, LS91] *The* Σ, Δ, *and* Θ*-levels of the extended low hierarchy (denoted* $\mathrm{EL}_k^{\mathrm{P},\Sigma}$, $\mathrm{EL}_k^{\mathrm{P},\Delta}$, *and* $\mathrm{EL}_k^{\mathrm{P},\Theta}$, *respectively) are defined as below.*

a) $\mathrm{EL}_k^{\mathrm{P},\Sigma} = \{A \mid \Sigma_k^{\mathrm{P}}(A) \subseteq \Sigma_{k-1}^{\mathrm{P}}(\mathrm{SAT} \oplus A)\}$, $k \geq 1$,

b) $\mathrm{EL}_k^{\mathrm{P},\Delta} = \{A \mid \Delta_k^{\mathrm{P}}(A) \subseteq \Delta_{k-1}^{\mathrm{P}}(\mathrm{SAT} \oplus A)\}$, $k \geq 2$,

c) $\mathrm{EL}_k^{\mathrm{P},\Theta} = \{A \mid \Theta_k^{\mathrm{P}}(A) \subseteq \Theta_{k-1}^{\mathrm{P}}(\mathrm{SAT} \oplus A)\}$, $k \geq 2$.

Intuitively, the extended low hierarchy classifies sets according to how easily the information encoded in the set can be extracted out of it. More precisely, a set A is, for example, in the k-th Σ-level of the extended low hierarchy if the information that can be extracted out of oracle A by the $\Sigma_k^{\mathrm{P}}(\cdot)$ operator is already accessible by the $\Sigma_{k-1}^{\mathrm{P}}(\cdot)$ operator, provided that besides A the $\Sigma_{k-1}^{\mathrm{P}}(\cdot)$ operator is additionally given access to the NP-complete set SAT in order to compensate for the loss in unrelativized power (w.r.t. oracle A). The next proposition gives a complete picture of the inclusional structure of the levels of the extended low hierarchy, and relates them with the corresponding levels of the low hierarchy inside NP.

Proposition 2.4 [BBS86, AH92, LS91, SL92]

i) $\mathrm{EL}_1^{\mathrm{P},\Sigma} = \mathrm{EL}_2^{\mathrm{P},\Theta} = \mathrm{EL}_2^{\mathrm{P},\Delta}$,

ii) For each $k \geq 2$, $\mathrm{EL}_k^{\mathrm{P},\Delta} \subsetneq \mathrm{EL}_k^{\mathrm{P},\Sigma} \subsetneq \mathrm{EL}_{k+1}^{\mathrm{P},\Theta} \subsetneq \mathrm{EL}_{k+1}^{\mathrm{P},\Delta}$,

iii) All levels of the extended low hierarchy are closed under Turing equivalence,

iv) For each $k \geq 2$, $\mathrm{NP} \cap \mathrm{EL}_k^{\mathrm{P},\Delta} = \mathrm{L}_k^{\mathrm{P},\Delta}$ *and* $\mathrm{NP} \cap \mathrm{EL}_k^{\mathrm{P},\Sigma} = \mathrm{L}_k^{\mathrm{P},\Sigma}$,

v) For each $k \geq 3$, $\mathrm{NP} \cap \mathrm{EL}_k^{\mathrm{P},\Theta} = \mathrm{L}_k^{\mathrm{P},\Theta}$.

Note that because of *i)* in the above proposition, $\mathrm{NP} \cap \mathrm{EL}_1^{\mathrm{P},\Sigma} = \mathrm{NP} \cap \mathrm{EL}_2^{\mathrm{P},\Theta} = \mathrm{NP} \cap \mathrm{EL}_2^{\mathrm{P},\Delta} = \mathrm{L}_2^{\mathrm{P},\Delta}$.

A wide variety of classes has been shown to be contained in the extended low hierarchy. We mention the following examples of classes located at the bottom three levels (APT stands for "almost polynomial time" [MP79], IC[log,poly] is the class of sets for which all input strings have low instance complexity [OKSW94], and the classes of P-selective, P-close, and P-cheatable sets have been introduced in [Se79, Sc86b, Be91], respectively):

$\mathrm{EL}_2^{\mathrm{P},\Delta}$**:** Tally sets [BB86], APT [LS91], IC[log,poly] [AKM],

$\mathrm{EL}_2^{\mathrm{P},\Sigma}$**:** BPP [Sc86a], P-selective and P-cheatable sets [ABG90],

$\mathrm{EL}_3^{\mathrm{P},\Theta}$**:** Sparse sets, P-close sets, and sets that are 1-truth-table reducible to some sparse set [LS91], sets that are bounded truth-table, conjunctively, or disjunctively reducible to some sparse set [AKM].

3 Bounding the complexity of computing advice

All problems in P/poly have low nonuniform complexity in the sense that for every n there is a "program" (e.g. a Turing machine or a circuit) of polynomial size that solves all instances of length up to n in polynomial time. Note that for a set A being in P/poly no restriction is imposed on the complexity of the advice function, i.e., the complexity of computing a program for a given input length can be arbitrarily high. Consequently, there is no limit on the uniform complexity of A, and in fact, A might be nonrecursive. For this reason, in order to investigate the complexity of computing advice functions for arbitrary sets A in P/poly, we have to consider upper (and lower) complexity bounds *relative to* A. The main result of this section states that for every set $A \in \mathrm{P/poly}$, correct advice can be computed in $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$. The proof is obtained by refining the techniques used by Gavaldà to derive the following upper bound.

Theorem 3.1 [Ga92b] *Every set A in the class* P/poly *has an advice function in* $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_3^{\mathrm{P}})$.

Gavaldà's theorem is incomparable to the following previously known upper bound which is due to Ko and Schöning.

Theorem 3.2 [KS85, Sc86a] *Every set A in* P/poly *has an advice function in* $\mathrm{FP}(\Sigma_2^{\mathrm{P}}(A))$.

On the other hand, the following lower bounds are known for the complexity of computing correct advice. Gavaldà and Watanabe constructed a set A in P/poly that is reducible to no sparse set in $\mathrm{NP}(A) \cap \text{co-NP}(A)$, and hence has no advice function in $\mathrm{FP}(\mathrm{NP}(A) \cap \text{co-NP}(A))$ [GW91]. Further, they showed that there is a set B (resp., C) in P/poly that has no advice function in $\mathrm{FP}(\mathrm{NP}_{\mathrm{m}}(B))$ (resp., $\mathrm{FP}(\mathrm{NP}_{\overline{\mathrm{m}}}(C))$) [WG92]. By Theorem 3.1, improving these lower bounds to $\mathrm{FP}(\mathrm{NP}_1(\cdot))$ is at least as hard as showing that the polynomial-time hierarchy does not collapse to Δ_2^{P}.

As a side remark, we note that there are close relationships between the complexity of computing correct advice for sets in P/poly and other research topics as, for example, polynomial-time query learnability and identity mapping networks (we refer the interested reader to [Wat92, Ga92a, Ga92b, WG92]).

In the following we recall the definitions and properties of hashing that we need. Sipser [Si83] used universal hashing, originally invented by Carter and Wegman [CW79], to decide (probabilistically) whether a finite set X is large or small. A linear hash function h from Σ^p to Σ^m is given by a Boolean (m,p)-matrix (a_{ij}) and maps any string $x = x_1 \dots x_p \in \Sigma^p$ to some string $y = y_1 \dots y_m$ where $y_i = \bigoplus_{j=1}^{p}(a_{ij} \wedge x_j)$. We say that a family $H = \{h_1, \dots, h_s\}$ of linear hash functions from Σ^p to Σ^m *hashes* a set $X \subseteq \Sigma^p$ if every $x \in X$ can be mapped to Σ^m by some hash function $h_k \in H$ which avoids any collision between x and some other string in X:

$$\forall\, x \in X\, \exists\, k\, (1 \le k \le s) \forall\, y \in X : x \neq y \Rightarrow h_k(x) \neq h_k(y).$$

Note that the predicate "H hashes X" is in co-NP provided that membership in the set X can be tested in NP. More formally, the set $\{\langle v, H\rangle \mid H \text{ hashes the set } X_v\}$ is in co-NP, if the sets X_v are succinctly represented in such a way that the language $\{\langle y, v\rangle \mid y \in X_v\}$ is in NP.

We denote the set of all families $H = \{h_1, \dots, h_s\}$ of s linear hash functions from Σ^p to Σ^m by $\mathcal{H}(s,p,m)$, and we say that a set $X \subseteq \Sigma^p$ is hashable in $\mathcal{H}(s,p,m)$ if there exists a family $H \in \mathcal{H}(s,p,m)$ that hashes X. In the case that the size s of the hash families coincides with the dimension m of their range we simply write $\mathcal{H}(m,p)$ instead of $\mathcal{H}(m,p,m)$. It is easy to see that the set $\{\langle v, 0^s, 0^p, 0^m\rangle \mid X_v \text{ is hashable in } \mathcal{H}(s,p,m)\}$ can be decided in Σ_2^{P} provided that membership in the set X_v is decidable in NP. The following theorem is proved by a pigeon-hole argument; it provides an upper bound on the cardinality of hashable sets.

Theorem 3.3 [Si83] *If a set $X \subseteq \Sigma^p$ is hashable in $\mathcal{H}(s,p,m)$ then $|X| \le s \cdot 2^m$.*

On the other hand, we get from the next theorem (called Coding Lemma in [Si83]) a lower bound on the cardinality of sets that are not hashable.

Theorem 3.4 [Si83] *Let $X \subseteq \Sigma^p$ be a set of cardinality at most 2^{m-1}. Then the probability that a uniformly at random chosen hash family $H \in \mathcal{H}(m,p)$ hashes X is at least $1/2$.*

By combining the above two theorems, one can already obtain a rough estimation for the cardinality of a given set $X \subseteq \Sigma^p$: let m be the minimum hash family size such that X is hashable in $\mathcal{H}(m,p)$; then $2^{m-2} < |X| \le m2^m$. As shown in the next theorem and in the subsequent lemma, a much sharper estimation can be obtained by estimating the size of the set X^r (for some large enough integer $r > 0$). This trick was for the first time used by Stockmeyer to prove that any #P function can be approximated in polynomial time with the help of an oracle from Σ_2^{P} [St85] (see also [Pi88, KST89, Kö89]).

Theorem 3.5 [St85] *For every #P function $f > 0$ and every polynomial $q > 0$, there is a $\mathrm{F}\Delta_3^{\mathrm{P}}$ function g such that for all strings x,*

$$0 \le \frac{g(x) - f(x)}{f(x)} < 1/q(|x|).$$

The following lemma shows how one can "approximately" find out whether a set $X \subseteq \Sigma^p$ is of size smaller or larger than a given number α: choose a suitable integer r and test whether the set X^r can be hashed by hash families of suitable size. If the answer is "no", then $|X|$ is definitely larger than α, otherwise, though we cannot conclude that $|X| \le \alpha$, it follows that $|X|$ can be only "slightly larger" than α (how much larger depends on the choice of r).

Lemma 3.6 *Let s, p, r be positive integers such that $r \ge 8ps^2$. Then the following implications hold for every set $X \subseteq \Sigma^p$ and every α, $1 \le \alpha \le 2^p$,*

$$\begin{aligned} |X| \le \alpha \quad &\Rightarrow \quad X^r \textit{ is hashable in } \mathcal{H}(\lceil r \log \alpha \rceil + 1, p \cdot r) \\ &\Rightarrow \quad |X| \le \alpha + \alpha/s. \end{aligned}$$

Proof The first implication is an immediate consequence of Theorem 3.4 and holds independently of the choice of r. To see the second implication observe that it follows by Theorem 3.3 that X^r is only hashable in $\mathcal{H}(\lceil r \log \alpha \rceil + 1, p \cdot r)$ if

$$|X^r| \quad \le \quad (\lceil r \log \alpha \rceil + 1) \cdot 2^{\lceil r \log \alpha \rceil + 1}.$$

Since $(\lceil r \log \alpha \rceil + 1) \cdot 2^{\lceil r \log \alpha \rceil + 1} < 4(2 + r \log \alpha) \cdot \alpha^r$, and since the inequalities $r \ge 8ps^2$ and $\alpha \le 2^p$ imply that $4(2 + r \log \alpha) < (1 + 1/s)^r$, the lemma follows. ∎

Gavaldà extended Sipser's Coding Lemma (Theorem 3.4) to the case of a collection $\mathcal{X}$ of exponentially many sets. He proved that any collection of 2^n many sets of suitable size can be simultaneously hashed by a hash family consisting of $(n+1)$ times as many functions (as compared to the case of a single set, see Theorem 3.4).

Theorem 3.7 [Ga92b] *Let $\mathcal{X}$ be a collection of 2^n sets $X_1, \ldots, X_{2^n} \subseteq \Sigma^p$, each of cardinality at most 2^{m-1}. Then the probability that a uniformly at random chosen hash family $H \in \mathcal{H}(m(n+1), p, m)$ hashes every X_i, $1 \leq i \leq 2^n$, is at least $1/2$.*

We now turn our attention back to the computation of advice functions for sets in P/poly. Let A be in P/poly, witnessed by an interpreter set $B \in \mathrm{P}$ and an advice function $h \in$ poly. By a padding argument, we can assume that there is a polynomial p such that for all n, $|h(0^n)| = p(n)$. We use the following notation which is adopted from [Ga92b].

- A string w is called a *correct advice* for $A^{=n}$ w.r.t. B, if for all $x \in \Sigma^n$, $x \in A \Leftrightarrow \langle x, w\rangle \in B$.
- A *sample* of $A^{=n}$ is (the encoding of) a set of pairs of the form $\langle x_i, A(x_i)\rangle$, where $x_i \in \Sigma^n$.
- For any sample $S = \{\langle x_1, b_1\rangle, \ldots, \langle x_k, b_k\rangle\}$ of $A^{=n}$, let

$$\mathit{Consistent}(S) = \{w \in \Sigma^{p(n)} \mid \forall i (1 \leq i \leq k) : B(\langle x_i, w\rangle) = b_i\}$$

  be the set of all advice strings w of length $p(n)$ that are correct on S. The cardinality of the set $\mathit{Consistent}(S)$ is denoted by $c(S)$.
- For any sample S of $A^{=n}$ and any string $x \in \Sigma^n$, let $\mathit{Accept}(x, S)$ (resp., $\mathit{Reject}(x, S)$) be the set of all consistent advice strings that *accept* x (resp., *reject* x). That is, $\mathit{Accept}(x, S) = \{w \in \mathit{Consistent}(S) \mid B(\langle x, w\rangle) = 1\}$ and $\mathit{Reject}(x, S) = \{w \in \mathit{Consistent}(S) \mid B(\langle x, w\rangle) = 0\}$. Note that $\mathit{Accept}(x, S)$ and $\mathit{Reject}(x, S)$ form a partition of the set $\mathit{Consistent}(S)$.

Theorem 3.8 *Every set A in* P/poly *has an advice function in* $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$.

Proof Let A be in P/poly, witnessed by an interpreter set $B \in \mathrm{P}$ and an advice function $h \in$ poly. As before, we can assume that $|h(0^n)| = p(n)$ for some fixed polynomial $p > 0$. We show that the function g defined as

$$g(0^n) = \min\{w \in \Sigma^{p(n)} \mid w \text{ is a correct advice for } A^{=n} \text{ w.r.t. } B\}$$

can be computed in polynomial time by an oracle transducer M using an oracle set in $\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}}$. The intuitive idea behind the algorithm performed by M is the following: M on input 0^n first computes a sample S of $A^{=n}$ which has the property that for every string x of length n the majority of all advice strings in *Consistent*(S) decides x correctly, i.e.

$$x \in A \iff |Accept(x,S)| > |Reject(x,S)|.$$

Moreover, the construction of S guarantees that the two sets $Accept(x,S)$ and $Reject(x,S)$ are very different in size, i.e., for every $x \in \Sigma^n$ a large majority of strings in $Consistent(S)$ is in the set

$$Correct(x,S) = \{w \in Consistent(S) \mid B(\langle x,w \rangle) = A(x)\}$$

of consistent advice strings that decide x correctly, and only a small minority is in the complementary set

$$Incorrect(x,S) = \{w \in Consistent(S) \mid B(\langle x,w \rangle) \neq A(x)\}.$$

In addition to S, machine M constructs a hash family H that hashes simultaneously all sets $Incorrect(x,S)$, $x \in \Sigma^n$, but none of the sets $Correct(x,S)$, $x \in \Sigma^n$. Having constructed S and H, M uses the Σ_2^{P} oracle set

$$\begin{aligned} U = \{\langle 0^n, H, u\rangle \mid & \text{ there exists a } v \text{ such that } |uv| = p(n) \text{ and for all } x \in \Sigma^n \text{ it} \\ & \text{holds that either } B(\langle x, uv\rangle) = 1 \text{ and } H \text{ hashes } Reject(x,S) \\ & \text{or } B(\langle x, uv\rangle) = 0 \text{ and } H \text{ hashes } Accept(x,S)\} \end{aligned}$$

to determine by prefix search the lexicographically first correct advice string w_{min} of length $p(n)$.

A formal description of M is given in Figure 1 (the polynomial r and the functions l, m are defined below). Starting with the empty sample, M constructs in the loop a sample S and a hash family $H \in \mathcal{H}((n+1)\cdot m(S), p(n), m(S))$ that hashes simultaneously all sets $Incorrect(x,S)$, $x \in \Sigma^n$. By Theorem 3.7, the existence of such a hash family is guaranteed provided that for all $x \in \Sigma^n$, $|Incorrect(x,S)| \le 2^{m(S)-1}$. Hence M extends S as long as M finds (inside the loop) an instance $x \in \Sigma^n$ for which $|Incorrect(x,S)|$ is possibly larger than $2^{m(S)-1}$. This procedure is similar to the so-called "majority vote strategy" due to D. Angluin [An88].

During each execution of the loop, M first looks for an instance $x \in \Sigma^n$ such that both of the two sets $Accept(x,S)$ and $Reject(x,S)$ (and hence $Incorrect(x,S)$) are reasonably large (the exact meaning of the if-condition will become clear after Claim 2 below). If such an x exists, then M expands S by the pair $\langle x, A(x)\rangle$ and repeats the loop. Otherwise, M constructs a hash family H in $\mathcal{H}((n+1)\cdot m(S), p(n), m(S))$ that hashes for every $x \in \Sigma^n$ *exactly* one of the two sets $Accept(x,S)$ and $Reject(x,S)$ (by Claim 1 below, H cannot hash both of them). Then M checks whether for some $x \in \Sigma^n$, the set $Incorrect(x,S)$ is not hashed by H. In that case, H must hash $Correct(x,S)$ and thus $|Incorrect(x,S)| > c(S)/2$ by Claim 1 below. Hence M expands S by the pair $\langle x, A(x)\rangle$ and repeats the loop. Otherwise, H is the desired hash family and M determines the lexicographically smallest correct advice string w_{min}.

```
input 0^n
S := ∅
loop
  if there exists an x ∈ Σ^n such that
     there exist H_1, H_2 ∈ H(l(S), r(n) · p(n)) such that
     H_1 hashes Accept(x, S)^{r(n)} and H_2 hashes Reject(x, S)^{r(n)}
  then
    construct by prefix search the lexicographically first such x
    S := S ∪ {⟨x, A(x)⟩}
  else
    construct by prefix search the lexicographically first hash family
    H ∈ H((n + 1) · m(S), p(n), m(S)) such that for all x ∈ Σ^n,
    H hashes either Accept(x, S) or Reject(x, S)
    if there exists an x ∈ Σ^n such that H does not hash Incorrect(x, S)
    then
      construct by prefix search the lexicographically first such x
      S := S ∪ {⟨x, A(x)⟩}
    else exit(loop) end
  end
end loop
compute the smallest correct advice string w_min using oracle U
output w_min
```

Figure 1: An $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$ algorithm for computing correct advice.

We now analyze the algorithm in more detail. Observe that the function $c(S) = |\mathit{Consistent}(S)|$ is in #P. Thus, by Theorem 3.5, it can be approximated by an $\mathrm{F}\Delta_3^{\mathrm{P}}$ function within an arbitrarily small relative error, i.e., for every polynomial $q > 0$ (which will be fixed later) there is a $\mathrm{F}\Delta_3^{\mathrm{P}}$ function c'' such that

$$0 \leq \frac{c''(S) - c(S)}{c(S)} < 1/q(n).$$

Note that c'' never underestimates c, i.e., c'' actually is an upper bound for c. To obtain a lower bound for c we define the function

$$c'(S) = \frac{q(n)}{q(n)+1} c''(S). \tag{1}$$

By a padding argument, we can assume that the number of correct advice strings for $A^{=n}$ in $\Sigma^{p(n)}$ is at least $4(n+1)p(n)$. Hence the following inequalities can be assumed to hold for every sample S of $A^{=n}$:

$$4(n+1)p(n) \leq c'(S) \leq c(S) \leq c''(S) \leq 2^{p(n)}. \tag{2}$$

Next we choose the function m just small enough to guarantee that no hash family $H \in \mathcal{H}((n+1)\cdot m(S), p(n), m(S))$ is able to hash for some $x \in \Sigma^n$ both of $\mathit{Accept}(x, S)$ and $\mathit{Reject}(x, S)$,

$$m(S) = \left\lfloor \log \frac{c'(S)}{2(n+1)p(n)} \right\rfloor. \tag{3}$$

Observe that as a consequence of (2), $m(S) \geq 1$ for every sample S.

Claim 1 *For every sample S and every $x \in \Sigma^n$, only sets $X \subseteq \Sigma^{p(n)}$ of size smaller than $c(S)/2$ are hashable in $\mathcal{H}((n+1)\cdot m(S), p(n), m(S))$.*

Proof of Claim 1. By Theorem 3.3, any family $H \in \mathcal{H}((n+1)\cdot m(S), p(n), m(S))$ can only hash sets $X \subseteq \Sigma^{p(n)}$ of cardinality at most

$$\begin{aligned}(n+1)\cdot m(S)\cdot 2^{m(S)} &< \frac{c'(S)}{2p(n)} \log c'(S), \text{ by (3)} \\ &\leq \frac{c(S)}{2}, \text{ by (2).}\end{aligned}$$

□ *Proof of Claim 1.*

Now we consider the test performed by M at the beginning of the loop to find out whether there exists an instance $x \in \Sigma^n$ such that $\mathit{Accept}(x, S)$ and $\mathit{Reject}(x, S)$ are both reasonably large. In order to check this by a Σ_2^{P} predicate, we use the fact that $\mathit{Accept}(x, S)$ and $\mathit{Reject}(x, S)$ partition $\mathit{Consistent}(S)$ and let M make the equivalent test whether neither of the two sets is close in size to $\mathit{Consistent}(S)$. To increase the accuracy of the test, M checks for a suitable polynomial r whether the sets $\mathit{Accept}(x, S)^{r(n)}$ and $\mathit{Reject}(x, S)^{r(n)}$ are hashable by hash families of suitable size.

Since $c(S \cup \langle x, A(x)\rangle) = c(S) - |\mathit{Incorrect}(x, S)|$, the number of loop iterations is polynomially bounded provided that M expands S only by strings x such that for some fixed polynomial t,

$$|\mathit{Incorrect}(x, S)| \geq \frac{c(S)}{t(n)}.$$

Hence, it suffices to choose the polynomials q and r and the parameter $l(S)$ in such a way that the following two implications hold:

$|\mathit{Accept}(x, S)| > 2^{m(S)-1}$ and $|\mathit{Reject}(x, S)| > 2^{m(S)-1}$
$\Rightarrow$ $\mathit{Accept}(x, S)^{r(n)}$ and $\mathit{Reject}(x, S)^{r(n)}$ are hashable in $\mathcal{H}(l(S), r(n)\cdot p(n))$
$\Rightarrow$ $|\mathit{Accept}(x, S)| \geq c(S)/t(n)$ and $|\mathit{Reject}(x, S)| \geq c(S)/t(n)$.

Using the fact that $\mathit{Accept}(x, S)$ and $\mathit{Reject}(x, S)$ partition $\mathit{Consistent}(S)$ it is easy to derive the validity of the above implications (for $t = 2q$) from the next claim. Define $l(S) = \lceil r(n) \log(c''(S) - 2^{m(S)-1}) \rceil + 1$.

Claim 2 *The polynomials q and r can be chosen such that for every set $X \subseteq \Sigma^{p(n)}$,*

$$\begin{aligned} |X| \leq c(S) - 2^{m(S)-1} \quad &\Rightarrow \quad X^{r(n)} \text{ is hashable in } \mathcal{H}(l(S), r(n) \cdot p(n)) \\ &\Rightarrow \quad |X| \leq c(S) - c(S)/(2q(n)). \end{aligned}$$

Proof of Claim 2. The first implication holds independently of the choice of the polynomials q and r since if $|X| \leq c(S) - 2^{m(S)-1}$, then we have by the choice of $l(S)$,

$$|X^{r(n)}| \leq \left(c(S) - 2^{m(S)-1}\right)^{r(n)} \leq \left(c''(S) - 2^{m(S)-1}\right)^{r(n)} \leq 2^{l(S)-1}.$$

Hence, the hashability of $X^{r(n)}$ in $\mathcal{H}(l(S), r(n) \cdot p(n))$ follows by Theorem 3.4. To derive the second implication, let $q(n) = 16(n+1)p(n)$. By the choice of $m(S)$, see (3) above, we have

$$m(S) - 1 \geq \log \frac{c'(S)}{2(n+1)p(n)} - 2 = \log \frac{2c'(S)}{q(n)}.$$

From (1) it follows that

$$c''(S) - 2^{m(S)-1} \leq \frac{q(n)+1}{q(n)} c'(S) - \frac{2c'(S)}{q(n)} = \frac{q(n)-1}{q(n)} c'(S). \tag{4}$$

By Lemma 3.6, we can choose for r a positive polynomial such that for every set $X \subseteq \Sigma^{p(n)}$, $X^{r(n)}$ is only hashable in $\mathcal{H}(l(S), r(n) \cdot p(n))$ if

$$|X| \leq \frac{2q(n)+1}{2q(n)} \left(c''(S) - 2^{m(S)-1}\right). \tag{5}$$

Putting (4) and (5) together we get

$$|X| \leq \frac{2q(n)+1}{2q(n)} \cdot \frac{q(n)-1}{q(n)} c'(S) \leq \frac{2q(n)-1}{2q(n)} c(S).$$

This completes the proof of Claim 2. □ *Proof of Claim 2.*

To complete the proof of the theorem it remains to show that the computation of M can be performed in $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$. First observe that as argued above, the number of loop iterations (and therefore the size of the sample S computed by M) is polynomially bounded since every extension of S cancels an inverse polynomial fraction of strings from *Consistent*(S). Since $|S|$ is polynomially bounded, $c''(S)$ (and therefore also $c'(S)$, $l(S)$ and $m(S)$) can be computed in polynomial time by asking a Σ_2^{P} oracle. The evaluation of the conditions of the two if-statements (and the prefix searches in the positive case) can be done in polynomial time by asking an oracle in Σ_2^{P} and $\mathrm{NP}_1(A)$, respectively. Similarly, the prefix search for the hash family H can be performed under a Σ_2^{P} oracle, and as argued at the beginning of the proof, w_{min} can be determined in polynomial time by asking the Σ_2^{P} oracle set U. ∎

The previous proof shows more generally that every set A in $\mathcal{C}$/poly has an advice function in $\mathrm{FP}(\mathrm{NP}(A \oplus \mathcal{C}) \oplus \Sigma_2^{\mathrm{P}}(\mathcal{C}))$, where the $\mathrm{NP}(A \oplus \mathcal{C})$ oracle can be decided by an NP oracle machine that asks on each computation path at most one query to A. Thus, using the fact that an NP $\cap$ co-NP oracle does not provide the class NP (and consequently none of the higher levels of the polynomial-time hierarchy) with any additional power, i.e. $\mathrm{NP}(\mathrm{NP} \cap \text{co-NP}) = \mathrm{NP}$ [Sc83], we easily get the following extension of Theorem 3.8.

Theorem 3.9 *Every set A in* $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ *has an advice function computable in* $\mathrm{FP}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$.

Observe that it follows already from Theorems 3.8 and 3.9 that the classes P/poly and (NP $\cap$ co-NP)/poly are contained in the third Δ-level $\mathrm{EL}_3^{\mathrm{P},\Delta}$ of the extended low hierarchy. To show that P/poly is contained in $\mathrm{EL}_3^{\mathrm{P},\Theta}$, it would suffice to improve the upper bound for computing correct advice to $\mathrm{F}\Theta(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$ (also $\mathrm{F}\Theta(\mathrm{NP}(A \oplus \mathrm{NP}))$ would suffice). However, such an improvement is not possible since it would imply the containment of P/poly in P/log (see e.g. [OKSW94] for a proof of $\mathrm{P/log} \subsetneq \mathrm{P/poly}$). Rather, the containment of P/poly in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ follows from our main result in the next section that on input 0^n, an $\mathrm{F}\Theta(\mathrm{NP}(A \oplus \mathrm{NP}))$ transducer can compute a certain kind of census information that enables an $\mathrm{NP}(A \oplus \mathrm{NP})$ algorithm to decide whether a given advice string is correct for $A^{=n}$.

Closely related to the complexity of computing correct advice is the complexity of tally set descriptions for sets in P/poly (= P(TALLY) [Ha83]). A *tally set description* (c.f. [GW91]) for a set A in P/poly is a tally set T such that $A \in \mathrm{P}(T)$. As an immediate consequence of Theorem 3.8 we get the following corollary which improves the $\Delta_3^{\mathrm{P}}(A)$ upper bound for the complexity of tally set descriptions for sets in P/poly due to Ko and Schöning [KS85, Sc86a], but is incomparable with the recent $\mathrm{NP}(A \oplus \Sigma_2^{\mathrm{P}})$ bound of Gavaldà [Ga92b].

Corollary 3.10 *For every set A in the class* P/poly *there exists a tally set T in* $\mathrm{P}(\mathrm{NP}_1(A) \oplus \Sigma_2^{\mathrm{P}})$ *such that* $A \in \mathrm{P}(T)$.

4 Lowness

In this section, we prove that P/poly is contained in $\mathrm{EL}_3^{\mathrm{P},\Theta}$. In order to obtain the $\mathrm{L}_3^{\mathrm{P},\Sigma}$-lowness of all NP sets in P/poly, Ko and Schöning [KS85] used the fact that the correctness of a given advice string is decidable in co-NP(A). This fact is also the key to the $\mathrm{EL}_3^{\mathrm{P},\Sigma}$-lowness of P/poly (see [BBS86]). We show in Lemma 4.1 below that a set A in P/poly is contained in $\mathrm{EL}_3^{\mathrm{P},\Theta}$ provided that advice strings can be checked for correctness by an $\mathrm{NP}(\mathrm{NP} \oplus A)$ algorithm that takes advice of an $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$ function. In the subsequent Theorem 4.2 we refine the techniques of the previous section and apply a census argument to show that indeed every set A in P/poly fulfills the assumptions of Lemma 4.1.

Lemma 4.1 *Let $A \in \mathrm{P/poly}$, witnessed by an interpreter set $B \in \mathrm{P}$ and an advice function $h \in \mathrm{poly}$. Assume that $|h(0^n)| = p(n)$ for some polynomial p and that the language*

$$C = \{\langle 0^n, w\rangle \mid w \textit{ is a correct advice string of length } p(n) \textit{ for } A^{=n} \textit{ w.r.t. } B\}$$

is in $\mathrm{NP}(\mathrm{NP} \oplus A)/\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$. Then A is in $\mathrm{EL}_3^{\mathrm{P},\Theta}$.

Proof Since the Θ_k^{P} classes are closed under the $\Theta(\cdot)$ operator,[3] i.e., $\Theta(\Theta_k^{\mathrm{P}}) = \Theta_k^{\mathrm{P}}$ (see also [CS92]), it suffices to prove the inclusion of $\Sigma_2^{\mathrm{P}}(A)$ in $\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$. Let L be a set in $\Sigma_2^{\mathrm{P}}(A)$. By the quantifier characterization of the polynomial-time hierarchy [St77, Wr77] there exists a deterministic, polynomial-time oracle machine M such that

$$L = \{x \mid \exists y \forall z : \langle x, y, z\rangle \in L(M, A)\},$$

where the quantifiers range over all strings of length $q(|x|)$ for some fixed polynomial q. Further, there is a polynomial t such that on input $\langle x, y, z\rangle$, machine M asks only queries of length at most $t(|x|)$. Let L' be the language

$$L' = \{\langle x, w_0, \ldots, w_{t(|x|)}\rangle \mid \exists y \forall z : \langle x, y, z, w_0, \ldots, w_{t(|x|)}\rangle \in L(M')\},$$

where on input $\langle x, y, z, w_0, \ldots, w_{t(|x|)}\rangle$, M' simulates M on input $\langle x, y, z\rangle$, answering each oracle query u according to whether $\langle u, w_{|u|}\rangle \in B$. Then $L' \in \Sigma_2^{\mathrm{P}}$, and we have that x is in L if and only if

$$\exists w_0, \ldots, w_{t(|x|)} : \langle x, w_0, \ldots, w_{t(|x|)}\rangle \in L' \text{ and } \forall i (0 \leq i \leq t(|x|)) : \langle 0^i, w_i\rangle \in C.$$

Thus, L is in $\mathrm{NP}(\mathrm{NP} \oplus A)/\mathrm{FP}_{tt}(\mathrm{NP}(\mathrm{NP} \oplus A)) \subseteq \mathrm{P}_{tt}(\mathrm{NP}(\mathrm{NP} \oplus A)) = \Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$. ■

Now we prove our main result that for every set A in P/poly the correctness of a given advice string can be checked by an $\mathrm{NP}(\mathrm{NP} \oplus A)$ algorithm that takes advice of an $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$ function.

Theorem 4.2 *Let $A \in \mathrm{P/poly}$, witnessed by an interpreter set $B \in \mathrm{P}$ and an advice function $h \in \mathrm{poly}$, where $|h(0^n)| = p(n)$ for some polynomial p. Then the language*

$$C = \{\langle 0^n, w\rangle \mid w \textit{ is a correct advice string of length } p(n) \textit{ for } A^{=n} \textit{ w.r.t. } B\}$$

is in $\mathrm{NP}(\mathrm{NP} \oplus A)/\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$.

[3] Since for every oracle class $\mathcal{C}$, $\Theta(\mathcal{C})$ is contained in $\mathrm{P}_{tt}(\mathcal{C})$, and since $\mathrm{P}_{tt}(\mathrm{NP})$ is contained in $\Theta(\mathrm{NP})$, as shown in [He87, BH91] via a proof that relativizes, it follows that $\Theta(\Theta(\mathrm{NP}(\mathcal{C}))) \subseteq \mathrm{P}_{tt}(\mathrm{P}_{tt}(\mathrm{NP}(\mathcal{C}))) \subseteq \mathrm{P}_{tt}(\mathrm{NP}(\mathcal{C})) \subseteq \Theta(\mathrm{NP}(\mathcal{C}))$. Hence, for $\mathcal{C} = \Sigma_k^P$ we have $\Theta(\Theta_{k+2}^{\mathrm{P}}) = \Theta_{k+2}^{\mathrm{P}}$.

input $\langle 0^n, w, k\rangle$
($\star$ k is assumed to be the advice provided by function g on 0^n $\star$)
guess $x_1, \ldots, x_k \in \Sigma^n$
$S := \{\langle x_1, A(x_1)\rangle, \ldots, \langle x_k, A(x_k)\rangle\}$
guess $H \in \mathcal{H}(l(n,k), r(n)\cdot p(n))$
if H hashes $\mathit{Consistent}(S)^{r(n)}$ **then**
  guess $H' \in \mathcal{H}((n+1)m(n,k), p(n), m(n,k))$
  if for all $x \in \Sigma^n$ either $B(\langle x, w\rangle) = 1$ and H' hashes $\mathit{Reject}(x,S)$,
    or $B(\langle x, w\rangle) = 0$ and H' hashes $\mathit{Accept}(x,S)$
  then *accept* **else** *reject* **end**
else *reject* **end**

Figure 2: An NP(NP $\oplus$ A) algorithm for checking correct advice.

Proof We show that C can be decided by a nondeterministic machine M' that uses an NP $\oplus$ A oracle and gets advice from a function g in $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$. A formal description of M' is given in Figure 2 (the polynomial r and the FP functions l and m are defined below).

As in the proof of Theorem 3.8, M' determines a sample S of $A^{=n}$ such that for all instances $x \in \Sigma^n$, the set $\mathit{Incorrect}(x,S)$ is small compared to $\mathit{Correct}(x,S)$. However, S is now guessed by M' rather than computed deterministically. As asserted by Claim 3 below, there exists such a sample of size $k = g(0^n)$. Moreover, M' can verify that a guessed sample S of size k indeed has the desired property by guessing a hash family H of appropriate size and checking whether H hashes the set $\mathit{Consistent}(S)^{r(n)}$.

Claim 3 *For every polynomial $t > 0$ there are a polynomial r, functions d, l in* FP *and an* $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$ *function g such that for all n,*

i) there exists a sample S of $A^{=n}$ of size $g(0^n)$ such that the set $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n, g(0^n)), r(n)\cdot p(n))$, and

ii) for every sample S as in i), $c(S) \geq d\big(n, g(0^n)+1\big)$ and $|\mathit{Incorrect}(x,S)| \leq d\big(n, g(0^n)+1\big)\big/ t(n)$ for all $x \in \Sigma^n$.

The basic idea behind the proof of Claim 3 is the following: Let $c^*(n,j) = \min_{|S|=j} c(S)$. Then for any sample S and for all instances $x \in \Sigma^n$ we have that $|\mathit{Incorrect}(x,S)| \leq c(S) - c^*(n, |S|+1)$. The polynomial r and the functions l, d are chosen in such a way that the hashability of the set $\mathit{Consistent}(S)^{r(n)}$ in $\mathcal{H}(l(n,j), r(n)\cdot p(n))$ guarantees that $c(S)$ can only be slightly larger than $d(n,j)$. On the other hand, the value $g(0^n)$ has the property that $d(n, g(0^n))$ is very close to $c^*(n, g(0^n)+1)$. Hence, any of the sets $\mathit{Incorrect}(x,S)$, $x \in \Sigma^n$, must be small, if $|S| = g(0^n)$ and $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n, g(0^n)), r(n)\cdot p(n))$.

We first finish the description of M', before we give a formal proof for Claim 3. After guessing a sample S of size k and verifying that $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n,k), r(n)\cdot p(n))$, M' guesses a hash family H' and checks whether for all $x \in \Sigma^n$,

- H' hashes either $\mathit{Accept}(x,S)$ or $\mathit{Reject}(x,S)$, and
- H' hashes $\mathit{Reject}(x,S)$ if and only if $B(\langle x,w\rangle) = 1$.

Assuming that $k = g(0^n)$, we know by Claim 3 that the sets $\mathit{Incorrect}(x,S)$ are small compared to $\mathit{Correct}(x,S)$. In the sequel we choose the function m and the polynomial t such that H' is not able to hash any of the sets $\mathit{Correct}(x,S)$.

In order to guarantee that any $H' \in \mathcal{H}((n+1)m(n,k), p(n), m(n,k))$ hashes at most one of the two sets $\mathit{Accept}(x,S)$ and $\mathit{Reject}(x,S)$, if $c(S) \geq d(n,k+1)$, we define the function m as

$$m(n,k) \quad = \quad \left\lfloor \log \frac{d(n,k+1)}{2(n+1)p(n)} \right\rfloor . \tag{6}$$

Without limitation of generality we can assume that the number of correct advice strings for $A^{=n}$ in $\Sigma^{p(n)}$ is not smaller than $4(n+1)p(n)$. Hence, also $d(n,k+1)$ can be assumed to be not smaller than $4(n+1)p(n)$, implying that $m(n,k) \geq 1$. By the definition of m it follows from Theorem 3.3 that H' can only hash sets X of cardinality

$$\begin{aligned} |X| &\leq (n+1)\cdot m(n,k)\cdot 2^{m(n,k)} \\ &< \frac{d\big(n,k+1\big)}{2p(n)} \log d\big(n,k+1\big), \text{ by (6)} \\ &\leq c(S)/2, \text{ since } d\big(n,k+1\big) \leq c(S) \leq 2^{p(n)}. \end{aligned}$$

Next we choose the polynomial t such that for every sample S as in part $i)$ of Claim 3 there exists a hash family $H' \in \mathcal{H}((n+1)m(n,k), p(n), m(n,k))$ that hashes simultaneously all sets $\mathit{Incorrect}(x,S)$, $x \in \Sigma^n$. By Theorem 3.7, such a family exists if $|\mathit{Incorrect}(x,S)| \leq 2^{m(n,k)-1}$ for all $x \in \Sigma^n$. Let $t(n) = 8(n+1)p(n)$, then it follows from part $ii)$ of Claim 3 and from (6) that for every $x \in \Sigma^n$,

$$\begin{aligned} |\mathit{Incorrect}(x,S)| &\leq d\big(n,k+1\big)\big/ 8(n+1)p(n) \\ &\leq 2^{m(n,k)-1}. \end{aligned}$$

To summarize, if S is a sample as in part $i)$ of Claim 3, and if H' is a hash family from $\mathcal{H}((n+1)m(n,k), p(n), m(n,k))$ that hashes for every $x \in \Sigma^n$ one of the sets $\mathit{Accept}(x,S)$ and $\mathit{Reject}(x,S)$, then H' hashes for every $x \in \Sigma^n$ the set $\mathit{Incorrect}(x,S)$ but not the set $\mathit{Correct}(x,S)$. Hence, $x \in A$ if and only if H' hashes the set $\mathit{Reject}(x,S)$, and $x \notin A$ if and only if H' hashes the set $\mathit{Accept}(x,S)$.

This shows that exactly the advice strings $w \in \Sigma^{p(n)}$ that are correct for $A^{=n}$ are accepted by M', provided that the correct value $k = g(0^n)$ is given along with the input.

It remains to prove Claim 3. Let

$$d(n,j) \;=\; 2^{p(n)}\left(1 - \frac{1}{2t(n)}\right)^j.$$

Since $d(n,0) = 2^{p(n)} = c^*(n,0)$, and since $c^*(n,j) \geq 4(n+1)p(n)$ for all $j \geq 0$, there exists a j such that $d(n,j) \geq c^*(n,j)$ and $d(n,j+1) \leq c^*(n,j+1)$, implying that $c^*(n,j) - c^*(n,j+1) \leq d(n,j) - d(n,j+1)$. In the sequel we show how to compute by binary search a value k for which the difference between $c^*(n,k)$ and $c^*(n,k+1)$ is "almost" bounded by $d(n,k) - d(n,k+1)$.

By Lemma 3.6, there is a polynomial $r > 0$ such that for every set $X \subseteq \Sigma^{p(n)}$ and all $j \geq 0$ such that $d(n,j) \geq 1$,

$$|X| \leq d(n,j) \;\Rightarrow\; X^{r(n)} \text{ is hashable in } \mathcal{H}(l(n,j), r(n)\cdot p(n)) \tag{7}$$

$$\Rightarrow\; |X| \leq \frac{4t(n)+1}{4t(n)}\, d(n,j). \tag{8}$$

where $l(n,j) = \lceil r(n) \log d(n,j) \rceil + 1$. The advice function g is defined by the following algorithm which on input 0^n outputs $g(0^n)$.

> **input** 0^n
> $k := 0$
> $k' := \min\{j \in \mathcal{N} \mid \frac{4t(n)+1}{4t(n)}\, d(n,j) < 4(n+1)p(n)\}$
> **repeat**
> $j := \lfloor (k+k')/2 \rfloor$
>   **if** there exists a sample S of $A^{=n}$ of cardinality j such that
> $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n,j), r(n)\cdot p(n))$
>   **then** $k := j$ **else** $k' := j$ **end**
> **until** $k' = k+1$
> **output** k

First observe that the initial value of k' is polynomially bounded in n. Hence, the number of iterations of the repeat-loop is logarithmically bounded. Since the condition of the if-statement can be evaluated in $\mathrm{NP}(\mathrm{NP} \oplus A)$, it follows that g can be computed in $\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP} \oplus A)$.

To see that the first statement of the claim is fulfilled, notice that the predicate "there exists a sample S of $A^{=n}$ of cardinality k such that $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n,k), r(n)\cdot p(n))$" is an invariant of the repeat-loop. The predicate is clearly fulfilled whenever the value of k changes inside the loop, and by (7) above it also holds for the initial value $k = 0$ since $d(n,0) = 2^{p(n)}$.

To prove part ii), let S be an arbitrary sample of size $g(0^n)$ such that $\mathit{Consistent}(S)^{r(n)}$ is hashable in $\mathcal{H}(l(n, g(0^n)), r(n) \cdot p(n))$. By (8), it follows that

$$c(S) \quad \leq \quad \frac{4t(n)+1}{4t(n)}\, d\Big(n, g(0^n)\Big)\,. \tag{9}$$

Furthermore, also the predicate "there does not exist a sample S' of size k' for which the set $\mathit{Consistent}(S')^{r(n)}$ is hashable in $\mathcal{H}(l(n, k'), r(n) \cdot p(n))$" is an invariant of the repeat-loop. Notice that by (8), this is true for the initial value of k', and it clearly remains true afterwards. Therefore, since after the last execution of the repeat-loop we have $k' = k+1 = g(0^n)+1$, it follows from (7) that for every sample S' of size at most $g(0^n)+1$ (and consequently for the sample S),

$$c(S') \quad > \quad d\Big(n, g(0^n)+1\Big)\,. \tag{10}$$

Finally, we have to show that for all $x \in \Sigma^n$, $\mathit{Incorrect}(x, S)$ is of cardinality at most $d(n, g(0^n)+1)/\,t(n)$. By way of a contradiction assume that there exists an $x \in \Sigma^n$ such that $|\mathit{Incorrect}(x, S)| > d(n, g(0^n)+1)/\,t(n)$. Then it follows that

$$\begin{aligned}
c(S \cup \{\langle x, A(x)\rangle\}) \quad &< \quad c(S) - d\Big(n, g(0^n)+1\Big)\Big/\,t(n) \\
&\leq \quad \frac{4t(n)+1}{4t(n)}\, d\Big(n, g(0^n)\Big) - d\Big(n, g(0^n)+1\Big)\Big/\,t(n), \text{ by (9)} \\
&= \quad \left(\frac{4t(n)+1}{4t(n)-2} - \frac{1}{t(n)}\right) d\Big(n, g(0^n)+1\Big) \\
&< \quad d\Big(n, g(0^n)+1\Big)\,, \text{ since } t(n) \geq 8,
\end{aligned}$$

contradicting inequality (10). This completes the proof of the claim and of the theorem. ■

By combining the previous theorem with Lemma 4.1 we get the main result of the paper.

Theorem 4.3 P/poly *is contained in* $\mathrm{EL}_3^{\mathrm{P},\Theta}$.

Observe that the proof of Theorem 4.2 actually shows more generally that for a set A in $\mathcal{C}$/poly correct advice can be recognized in $\mathrm{NP}(\mathrm{NP}(\mathcal{C}) \oplus A)/\mathrm{F}\Theta_2^{\mathrm{P}}(\mathrm{NP}(\mathcal{C}) \oplus A)$. Therefore, we also obtain the following extension of Theorem 4.3.

Theorem 4.4 (NP $\cap$ co-NP)/poly *is contained in* $\mathrm{EL}_3^{\mathrm{P},\Theta}$.

As a direct consequence of Theorems 4.3 and 4.4 we get an improvement of Ko and Schöning's result that NP $\cap$ (P/poly) is contained in $\mathrm{L}_3^{\mathrm{P},\Sigma}$ [KS85].

Corollary 4.5 NP $\cap$ (NP $\cap$ co-NP)/poly *is contained in* $\mathrm{L}_3^{\mathrm{P},\Theta}$.

Since all $\leq_T^{SN}$-complete sets for NP are contained in $\mathrm{H}_1^{\mathrm{P},\Sigma} \subseteq \mathrm{H}_3^{\mathrm{P},\Theta}$, and since a set in $\mathrm{L}_3^{\mathrm{P},\Theta}$ cannot be at the same time in $\mathrm{H}_3^{\mathrm{P},\Theta}$ unless the polynomial-time hierarchy collapses to Θ_3^{P} [LS91], we have the following corollary.

Corollary 4.6 *If there exists a $\leq_T^{SN}$-complete set for* NP *in* $(\mathrm{NP} \cap \text{co-NP})/\text{poly}$, *then the polynomial-time hierarchy collapses to* Θ_3^{P}.

The above corollary improves the result of Ko and Schöning [KS85] that there exists no $\leq_T^{SN}$-complete set for NP in P/poly unless the polynomial-time hierarchy collapses to Σ_3^{P}, and extends the result of Kadin [Ka88] that there exists no co-sparse $\leq_T^{SN}$-complete set for NP unless the polynomial-time hierarchy collapses to Θ_3^{P}.

5 Optimality

Long and Sheu [LS91] refined the notion of extended lowness as follows.[4] For a class $\mathcal{C}$ of languages and a class F of functions from $\mathcal{N}$ to $\mathcal{N}$, let $\mathrm{P}^{\mathcal{C}[F]}$ denote the class of sets decidable by some deterministic polynomial-time oracle machine that for some function $f \in F$ asks on inputs of length n at most $f(n)$ many (adaptive) queries to some oracle $C \in \mathcal{C}$. Then, for each $k \geq 2$,

$$\mathrm{EL}_k^{\mathrm{P},F} = \left\{ A \;\middle|\; \Sigma_{k-1}^{\mathrm{P}}(A) \subseteq \mathrm{P}^{\Sigma_{k-2}^{\mathrm{P}}(\mathrm{SAT}\oplus A)[F]} \right\}.$$

Note that by the discussion at the beginning of the proof of Lemma 4.1, $\mathrm{EL}_k^{\mathrm{P},O(log(n))} = \mathrm{EL}_k^{\mathrm{P},\Theta}$ for $k \geq 2$. Long and Sheu proved the following lower bound results for sparse sets in the refined extended low hierarchy.

Theorem 5.1 [LS91] *Let $f(n) : \mathcal{N} \to \mathcal{N}$ be any function for which there exists a constant $c > 0$ such that $f(n) < c \cdot \log n$ infinitely often. Then, there exists a sparse set S such that $\Theta_2^{\mathrm{P}}(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(\mathrm{SAT}\oplus S)[f(n)]}$, implying that $S \notin \mathrm{EL}_3^{\mathrm{P},\{f\}}$.*

Theorem 5.2 [LS91] *Let $f(n) : \mathcal{N} \to \mathcal{N}$ be any function such that for all constants $c > 0$, $f(n) < c \cdot \log n$ infinitely often. Then, there exists a sparse set S such that $\Theta_2^{\mathrm{P}}(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(\mathrm{SAT}\oplus S)[O(f(n))]}$, implying that $S \notin \mathrm{EL}_3^{\mathrm{P},O(f(n))}$.*

Theorems 5.1 and 5.2 still leave open the possibility that every sparse set is contained in $\bigcup_{f \in O(\log n)} \mathrm{EL}_3^{\mathrm{P},\{f\}}$. However, by slightly refining the proof techniques that lead to Theorem 5.1, a sparse set S can be constructed that for every single $f \in O(\log n)$ is not contained in $\mathrm{EL}_3^{\mathrm{P},\{f\}}$.

[4] Actually, the definition for $\mathrm{EL}_k^{\mathrm{P},F}$ given in [LS91] is slightly more restrictive than the one given here, but their lower bound results for sparse sets (Theorems 5.1 and 5.2) are not affected by our extension.

Theorem 5.3 *There exists a sparse set S such that for every function $f \in O(\log n)$, $\Theta_2^{\mathrm{P}}(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(\mathrm{SAT}\oplus S)[f(n)]}$, implying that $S \notin \bigcup_{f\in O(\log n)} \mathrm{EL}_3^{\mathrm{P},\{f\}}$.*

From these results we can conclude that the lower and upper bounds for the location of P/poly in the extended low hierarchy are as tight as possible.

Acknowledgments

For helpful conversations and suggestions regarding this work the author is very grateful to V. Arvind, R. Beigel, L. Fortnow, R. Gavaldà, L. Hemaspaandra, T. Long, M. Mundhenk, U. Schöning, M.-J. Sheu, and R. Schuler. The author also thanks the two referees for their valuable suggestions.

References

[AH92] E. ALLENDER AND L. HEMACHANDRA. Lower bounds for the low hierarchy. *J. ACM* **39** (1992) 234–250.

[ABG90] A. AMIR, R. BEIGEL, AND W. GASARCH. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Structure in Complexity Theory Conference*, IEEE (1990) 232–243.

[An88] D. ANGLUIN. Queries and concept learning. *Machine Learning* **2** (1988) 319–342.

[AKM] V. ARVIND, J. KÖBLER, M. MUNDHENK. Upper bounds on the complexity of sparse and tally descriptions. To appear in *Mathematical Systems Theory.*

[BBS86] J.L. BALCÁZAR, R. BOOK, AND U. SCHÖNING. Sparse sets, lowness and highness. *SIAM Journal on Computing* **23** (1986) 679–688.

[BB86] J.L. BALCÁZAR AND R. BOOK. Sets with small generalized Kolmogorov complexity. *Acta Informatica* **23** (1986) 679–688.

[BDG] J.L. BALCÁZAR, J. DÍAZ, J. GABARRÓ. *Structural Complexity Theory.* (Springer, Berlin, 1988 and 1990).

[Be91] R. BEIGEL. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science* **84** (1991) 199–223.

[BH77] L. BERMAN, J. HARTMANIS. On isomorphism and density of NP and other complexity classes. *SIAM Journal on Computing* **6** (1977) 305–327.

[Bo74] R. BOOK. Tally languages and complexity classes. *Information and Control* **26** (1974) 186–193.

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

[BH91] S. Buss and L. Hay. On truth-table reducibility to SAT. *Information and Computation* **91** (1991) 86–102.

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences* **18** (1979) 143–154.

[CS92] J. Castro and C. Seara. The θ-operator and the low hierarchy. Technical Report LSI-92-16-R, Universitat Politécnica de Catalunya, 1992.

[Ga92a] R. Gavaldà. Kolmogorov Randomness and its Applications to Structural Complexity Theory. Doctoral Dissertation, Universitat Politècnica de Catalunya, 1992.

[Ga92b] R. Gavaldà. Bounding the complexity of advice functions. In *Proceedings of the 7th Structure in Complexity Theory Conference*, IEEE (1992) 249–254.

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

[GW91] R. Gavaldà and O. Watanabe. On the computational complexity of small descriptions. In *Proceedings of the 6th Structure in Complexity Theory Conference*, IEEE (1991) 89–101; the final version is to appear in *SIAM Journal on Computing.*

[Ha83] J. Hartmanis. On sparse sets in NP $-$ P. *Information Processing letters* **16** (1983) 55–60.

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

[He87] L. Hemachandra. The strong exponential hierarchy collapses. In *Proceedings of the 19th ACM Symposium on Theory of Computing* (1987) 110–122.

[He93] L. Hemachandra. Personal communication, Würzburg, 1993.

[HOW92] L. Hemachandra, M. Ogiwara, and O. Watanabe. How hard are sparse sets. In *Proceedings of the 7th Structure in Complexity Theory Conference*, IEEE (1992) 222–238.

[Ka88] J. Kadin. *Restricted Turing reducibilities and the structure of the polynomial time hierarchy.* Doctoral Dissertation, Cornell University, 1988.

[Ka89] J. Kadin. $\mathrm{P}^{\mathrm{NP}[\log n]}$ and sparse Turing-complete sets for NP. *Journal of Computer and System Sciences* **39** (1989) 282–298.

[Kä91] J. Kämper. Non-uniform proof systems: A new framework to describe non-uniform and probabilistic complexity classes. *Theoretical Computer Science* **85**(2) (1991) 305-331.

[KL80] R.M. Karp and R.J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings 12th ACM Symposium Theory of Computing* (1980) 302–309.

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

[Kö89] J. Köbler. *Strukturelle Komplexität von Anzahlproblemen.* Doctoral Dissertation. University of Stuttgart, 1989.

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

[KT94] J. Köbler and T. Thierauf. Complexity-restricted advice functions. *SIAM Journal on Computing* **23**(2) (1994) 261–275.

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

[Lo82] T.J. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science* **21** (1982) 1–25.

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

[MP79] A. Meyer and M. Paterson. With what frequency are apparently intractable problems difficult? Technical Report MIT/LCS/TM-126, M.I.T., 1979.

[MS73] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential time. In *Proceedings 13th IEEE Symposium on Switching and Automata Theory* (1973) 125–129.

[OKSW94] P. Orponen, K. Ko, U. Schöning, and O. Watanabe. Instance complexity. To appear in *J. ACM.*

[Pi88] M. Piotrów. On complexity of counting. In *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science,* Lecture Notes in Computer Science, Vol. 324 (Springer, Berlin, 1988) 472–482.

[Pi79] N. Pippenger. On simultaneous resource bounds. In *Proceedings 20th Symposium on Foundations of Computer Science*, IEEE (1979) 307–311.

[Pr75] V. Pratt. Every prime has a succinct certificate. *SIAM Journal on Computing* **4** (1975) 214–220.

[Sc83] U. Schöning. A low hierarchy within NP. *Journal of Computer and System Sciences* **27** (1983) 14–28.

[Sc86a] U. Schöning. *Complexity and Structure.* Lecture Notes in Computer Science, Vol. 211 (Springer, Berlin, 1986).

[Sc86b] U. Schöning. Complete sets and closeness to complexity classes. *Mathematical Systems Theory* **19** (1986) 29–41.

[Sc88] U. Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences* **37** (1988) 312–323.

[Se79] A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory* **13** (1979) 55–65.

[SL92] M.-J. Sheu and T.J. Long. The extended low hierarchy is an infinite hierarchy. *Proceedings of the 9th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Vol. 577 (Springer, Berlin, 1992) 187–198.

[Si83] M. Sipser. A complexity theoretic approach to randomness. *Proceedings 15th ACM Symposium Theory of Computing* (1983) 330–335.

[St77] L.J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science* **3** (1977) 1–22.

[St85] L.J. Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing* **14** (1985) 849–861.

[Va79] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science* **8** (1979) 189–201.

[Wag90] K.W. Wagner. Bounded query classes. *SIAM Journal on Computing* **19** (1990) 833–846.

[Wat92] O. Watanabe. On the complexity of small descriptions and related topics. In *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science,* Lecture Notes in Computer Science, Vol. 629 (Springer, Berlin, 1992) 82–94.

[WG92] O. Watanabe and R. Gavaldà. Structural analysis of polynomial time query learnability. Technical Report 92TR-0004, Dept. of Computer Science, Tokyo Institute of Technology (1992).

[Wr77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science* **3** (1977) 23–33.