[Sto76] L. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3:1–22, 1976.

[TB88] S. Tang and R. Book. Separating polynomial-time turing and truth-table degrees of tally sets. In *Proc. 15th Colloq. on Automata, Languages, and Programming, Lecture Notes in Computer Science*, volume 317, pages 591–599. Springer-Verlag, 1988.

[Wra76] C. Wrathall. Complete sets and the polynomial hierarchy. *Theor. Comput. Sci*, 3:23–33, 1976.

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

[Kö92] J. Köbler. Locating P/poly optimally in the extended low hierarchy. Technical Report Technical Report 92-05, Ulmer Informatik-Berichte, 1992.

[Kad87] J. Kadin. $\mathrm{P}^{\mathrm{NP}[\log n]}$ and sparse Turing-complete sets for NP. In *Proc. Struct. in Complexity Second Annual Conference*, pages 33–40, 1987.

[Kad88] J. Kadin. *Restricted Turing reducibilities and the structure of the polynomial time hierarchy.* PhD thesis, Cornell University, 1988.

[Ko86] K. Ko. Applying techniques of discrete complexity theory to numerical computation. In Ronald V. Book, editor, *Studies in complexity theory*, pages 1–62. Pitman Publishing Limited, 1986.

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

[LT91] A. Lozano and J. Torán. Self-reducible sets of small density. *Mathematical Systems Theory*, 24:83–100, 1991.

[Mah82] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *J. Comput. System Sci.*, 25:130–143, 1982.

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

[Sch83] U. Schöning. A low and a high hierarchy within NP. *J. Comput. System Sci.*, 27:14–28, 1983.

[Sch86a] U. Schöning. Complete sets and closeness to complexity classes. *Mathematical Systems Theory*, 19:29–41, 1986.

[Sch86b] U. Schöning. *Complexity and Structure.* Lecture Notes in Computer Science Vol. 211. Springer-Verlag, 1986.

References

[AH92] E. Allender and L. Hemachandra. Lower bounds for the low hierarchy. *J. ACM*, 39(1):234–251, 1992.

[AKM92] V. Arvind, J. Köbler, and M. Mundhenk. Lowness and the complexity of sparse and tally descriptions. Technical Report Technical Report 92-04, Ulmer Informatik-Berichte, 1992.

[Ang80] D. Angluin. On counting problems and the polynomial-time hierarchy. *Theoretical Computer Science*, 12:161–173, 1980.

[BB86] J. Balcázar and R. Book. Sets with small generalized Kolmogorov complexity. *Acta Informatica*, 23:679–688, 1986.

[BBS86] J. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness, and highness. *SIAM J. Comput.*, 15:739–747, 1986.

[BDG88] J. Balcázar, J. Diaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, 1988.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P =? NP question. *SIAM J. Comput.*, 4(4):431–441, Dec. 1975.

[BK88] R. Book and K. Ko. On sets reducible to sparse sets. *SIAM J. Comput.*, 17:903–919, 1988.

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

[BS] J. Balcázar and U. Schöning. Logarithmic advice classes. Technical Report LSI-88-12, Univ. Politecnica Catalunyu, Spain, to appear in Theoretical Computer Science.

[BS79] T. Baker and A. Selman. A second step toward the polynomial hierarchy. *Theoretical Computer Science*, 8:177–187, 1979.

[BT89] R. Book and Tang. A note on sparse sets and the polynomial-time hierarchy. *Information Processing Letters*, 33(3):141–143, 1989.

Corollary 5.10 *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 $S \notin EL_3^{P,O(f(n))}$.*

Corollary 5.11 *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.*

1. P-close $\not\subseteq EL_3^{P,O(f(n))}$.
2. P_m(SPARSE) $\not\subseteq EL_3^{P,O(f(n))}$.
3. P_{1-tt}(SPARSE) $\not\subseteq EL_3^{P,O(f(n))}$.
4. E_T(SPARSE) $\not\subseteq EL_3^{P,O(f(n))}$.

Acknowledgment: The authors are grateful to the anonymous referees for their many constructive comments and to Johannes Köbler for his suggestions regarding Theorems 5.7 and 5.9.

SAT) does produce a different output in iteration $j+1$ from that in iteration j, then this implies that, during iteration j, some string $< M, x, 0^k >$ was not in $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ and, during iteration $j+1$, $< M, x, 0^k >$ was made a permanent element of $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$. Since M_i is not allowed to make more than $f(n)$ queries to the oracle, taken over all possible oracle sets, M_i on 1^n could query at most $2^{f(n)}$ distinct strings. By the choice of n, $2^{f(n)} < n^c$. Thus, if a string is assigned to F in step (4) during iteration n^c+1, the output of M_i on 1^n relative to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ cannot change from iteration n^c+1 to iteration n^c+2, implying that the process would terminate before reaching step (4) n^c+2 times.

Finally, since at most n^c strings can be permanently assigned to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, $\| N \| \leq n^c \cdot p_i(n)$. Together with condition (c) on the choice of n, this implies that there are sufficiently many strings of length n so that n^c+1 strings can be assigned to F in step (4). This completes the proof of Theorem 5.7. □

We now generalize the definition of extended lowness relative to arbitrary sets of functions as follows: Let F be a set of functions $f : \mathcal{N} \to \mathcal{N}$. For each $k \geq 2$,

$$EL_k^{P,F} = \{A \mid \bigcup_{g \in F} \mathrm{P}^{\Sigma_{k-1}^P(A)[g(n)]} \subseteq \bigcup_{h \in F} \mathrm{P}^{\Sigma_{k-2}^P(A \oplus \mathrm{SAT})[h(n)]}\}.$$

The following corollary follows immediately from Theorem 5.7.

Corollary 5.8 *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 $S \notin EL_3^{P,\{f\}}$.*

Corollary 5.8 is unsatisfying in that the deterministic oracle machines in both $P^{\Sigma_2^P(S)[f(n)]}$ and in $P^{\mathrm{NP}(S \oplus \mathrm{SAT})[f(n)]}$ are allowed at most $f(n)$ queries, eliminating even the possibility of $O(f(n))$ queries. The proof technique used in the proof of Theorem 5.7 can easily be modified to account for $O(f(n))$ queries, if a stronger hypothesis is used.

Theorem 5.9 *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 $\Theta_2^P(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(S \oplus \mathrm{SAT})[O(f(n))]}$.*

computation witnessing that $< M, x, 0^k >$ in $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$. If M_i accepts 1^n relative to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, then Stage i is complete since currently $\| F \|$ is even. (Later iterations of this process may terminate because M_i rejects 1^n relative to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ when $\| F \|$ is currently odd.) If M_i rejects 1^n relative to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, then exactly one string of length n, not reserved for $\overline{S}$, is added to F and the process is repeated. As we will see, this process will be repeated at most $n^c + 1$ times so that $\| F \| \leq n^c + 1$. During this iterative process, N will be the set of strings of length n explicitly reserved for $\overline{S}$. Initially, $N = \emptyset$.

Iterative Process to Construct F:

(1) Run M_i on 1^n using oracle set $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$.

(2) For each query "$< M, x, 0^k > \in K((S_{i-1} \cup F) \oplus \mathrm{SAT})$?" that this computation makes, if $< M, x, 0^k > \in K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ and if membership of $< M, x, 0^k >$ in $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ was not made permanent in an earlier iteration of this process, then make $< M, x, 0^k >$ a permanent element of $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ by

- selecting an accepting computation of M on x relative to $(S_{i-1} \cup F) \oplus \mathrm{SAT}$, and
- if this computation makes a query "$y \in F$?" with $|y| = n$ and $y \notin F \cup N$, then add y to N.

(3) Using oracle set $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, if M_i accepts 1^n and $\| F \|$ is even or if M_i rejects 1^n and $\| F \|$ is odd, then terminate this process with the current F. Otherwise, proceed step (4).

(4) Add to F a string y of length n such that y is not currently in $F \cup N$, and repeat this process.

We will now argue that this process places at most $n^c + 1$ strings into F. Suppose that a string is added to F in step (4) during iteration j of this process. If the computation of M_i on 1^n relative to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$ does not produce a different output in iteration $j + 1$ from that in iteration j, then the process terminates since the size of F changed by 1 from iteration j to iteration $j + 1$. On the other hand, if the computation of M_i on 1^n relative to $K((S_{i-1} \cup F) \oplus$

Theorem 5.7 *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 $\Theta_2^P(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(S \oplus \mathrm{SAT})[f(n)]}$.*

Proof. Following the proof in [Kad87], we construct, for each function $f(n)$ and constant $c > 0$ satisfying the hypothesis, a sparse set S such that $\Theta_2^P(S) \not\subseteq \mathrm{P}^{\mathrm{NP}(S \oplus \mathrm{SAT})[f(n)]}$. Let $L = \{1^n \mid \| S^{=n} \| \text{ is odd } \}$. When S is sparse, $L \in \Theta_2^P(S)$. S will be constructed so that $L \notin \mathrm{P}^{\mathrm{NP}(S \oplus \mathrm{SAT})[f(n)]}$.

Ensuring that $L \notin \mathrm{P}^{\mathrm{NP}(S \oplus \mathrm{SAT})[f(n)]}$ is done by diagonalization. Let $M_1, M_2, \ldots$ be an effective enumeration of deterministic polynomial-time oracle machines with polynomial p_i bounding the running time of M_i for each $i > 0$. S is constructed in stages. Stage i ensures that $L \neq L(M_i, K(S \oplus \mathrm{SAT}))$, when M_i is restricted to at most $f(n)$ queries to the oracle set $K(S \oplus \mathrm{SAT})$. Each stage i determines an integer n_i such that all strings y, with $|y| \leq n_i$, have been assigned to S or $\overline{S}$ by the end of stage i, and such that no strings y, with $|y| > n_i$, have been assigned to S or $\overline{S}$ by the end of stage i. S_i and $\overline{S_i}$ denote the strings assigned to S and $\overline{S}$, respectively, by the end of stage i. Initially, let $S_0 = \{\lambda\}$, let $\overline{S_0} = \emptyset$, and let $n_0 = 0$.

Stage i:

1. Let n be such that:
   (a) $n > n_{i-1}$,
   (b) $f(n) < c \cdot \log n$, and
   (c) $n^c + 1 + n^c \cdot p_i(n) < 2^n$.

2. Find a finite set $F \subseteq \Sigma^{=n}$ such that $\| F \| \leq n^c + 1$ and such that $L \neq L(M_i, K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, when M_i is restricted to at most $f(n)$ queries to $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$. Let $S_i = S_{i-1} \cup F$, let $n_i = p_i(n)$, and let $\overline{S_i} = \overline{S_{i-1}} \cup \{y \mid y \notin F \text{ and } n_{i-1} < |y| \leq n_i\}$.

End of stage i

The set F is found by an iterative process. Initially, $F = \emptyset$. M_i is run on 1^n using oracle set $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$. For each query "$< M, x, 0^k > \in K((S_{i-1} \cup F) \oplus \mathrm{SAT})$?" that this computation makes, if $< M, x, 0^k > \in K((S_{i-1} \cup F) \oplus \mathrm{SAT})$, then strings are reserved for $\overline{S}$ to preserve membership of $< M, x, 0^k >$ in $K((S_{i-1} \cup F) \oplus \mathrm{SAT})$. The newly $\overline{S}$-reserved strings are those of length n not yet resereved for S or $\overline{S}$ that receive a no answer when queried in an accepting

Phase 1:

On input x with $|x| = n$, compute $j = p_B(p_A(p_L(n)))$. Compute $0^m = cf_S(0^j)$ with $O(\log n)$ queries to $cp_S \in \mathrm{NP}(S)$.

Phase 2:

Simulate M_L on input x. Whenever M_L makes the query "$y \in A$?", instead make the query "$y\#0^j\#0^m \in A'$?". Accept iff M_L accepts x.

It is easy to see that $L \in \Theta(\mathrm{NP}(S) \oplus \mathrm{NP}(S \oplus \mathrm{SAT})) \subseteq \Theta(\mathrm{NP}(S \oplus \mathrm{SAT})) = \Theta_2^P(S \oplus \mathrm{SAT})$, and thus $S \in EL_3^{P,\Theta}$. □

Corollary 5.6

1. P-close $\subseteq EL_3^{P,\Theta}$.
2. $\mathrm{P}_m(\mathrm{SPARSE}) \subseteq EL_3^{P,\Theta}$.
3. $\mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq EL_3^{P,\Theta}$.
4. $\mathrm{E}_T(\mathrm{SPARSE}) \subseteq EL_3^{P,\Theta}$.

Proof. As in the spirit of Theorem 4.3, we show that if $\{S \mid S \textit{ is sparse}\} \subseteq EL_3^{P,\Theta}$, then $\mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq EL_3^{P,\Theta}$. Applying Theorem 5.5, $\mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq EL_3^{P,\Theta}$. This also suffices to prove 1 and 2 since both P-close and $\mathrm{P}_m(\mathrm{SPARSE})$ are contained in $\mathrm{P}_{1-tt}(\mathrm{SPARSE})$.

Let $A \in \mathrm{P}_{1-tt}(\mathrm{SPARSE})$. By Proposition 2.3, there are sparse sets S_1 and S_2 such that $A \leq_{1-tt}^P S_1 \oplus S_2$ and $S_1 \oplus S_2 \leq_T^P A \oplus SAT$. Then, $\Theta_3^P(A) \subseteq \Theta_3^P(S_1 \oplus S_2) \subseteq \Theta_2^P(S_1 \oplus S_2 \oplus SAT) \subseteq \Theta_2^P(A \oplus SAT)$, and $A \in EL_3^{P,\Theta}$.

Part 4 follows from Theorem 5.5 and the fact that $EL_k^{P,\Theta}$, for $k \geq 2$, is closed under $\equiv_T^P$. □

Under the hypothesis that NP has sparse $\leq_T^P$-complete sets, Kadin proved that the polynomial-time hierarchy collapses to Θ_2^P. He also proved that this collapse is optimal in the following sense: Let $f(n)$ be any function for which there exists a constant $c > 0$ such that $f(n) < c \cdot \log n$ infinitely often. Then, there is a set A so that when letting $B = A \oplus \mathrm{QBF}$, $\mathrm{NP}(B)$ has sparse $\leq_T^P$-complete set and $\mathrm{PH}(B) = \Theta_2^P(B) \not\subseteq \mathrm{P}^{\mathrm{NP(B)}[f(n)]}$. We will use Kadin's proof to show that there is a sparse set S that is essentially no lower than $EL_3^{P,\Theta}$ in the extended low hierarchies.

We begin with a modified version of Kadin's result.

Theorem 5.5 $\{S \mid S \text{ is sparse}\} \subseteq EL_3^{P,\Theta}$.

Proof. The proof of this theorem is just the oracle machine version of the proof of Theorem 3.2. Let S be a sparse set and let p_S be a polynomial such that for all n, $\| S^{\leq n} \| \leq p_S(n)$.

Let $L \in \Theta_3^P(S)$. There exists a deterministic oracle machine M_L, a polynomial p_L that bounds the running time of M_L, and a constant $c > 0$ such that, on input x, M_L makes at most $c \cdot \log(|x|)$ queries to a set $A \in \Sigma_2^P(S)$, and $L = L(M_L, A)$. Similarly, A is accepted by a nondeterministic oracle machine M_A using an oracle set B in $\mathrm{NP}(S)$, which is accepted by a nondeterministic oracle machine M_B using S as the oracle set. Let p_A and p_B bound the running time of M_A and M_B, respectively.

Now consider the following machine M'_B, a modified version of M_B:
on input $y\#code(F)$:

1. Run M_B on input y.
2. For each query "$w \in S$?" that M_B makes, answer YES if and only if $w \in F$.
3. Accept input $y\#code(F)$ if and only if M_B accepts y.

Let $B' = L(M'_B)$. Note that if $F = S^{\leq j}$ for some $j \geq p_B(|y|)$, then $y\#code(F) \in B' \Leftrightarrow y \in B$. Also note that $B' \in \mathrm{NP}$.

Next consider M'_A, a modified version of M_A, that works as follows:
On input $x\#0^j\#0^m$, M'_A operates in two phases:

Phase 1:

Nondeterministically guess m different strings $y_1, \ldots, y_m$, each of length at most j. Using S as the oracle set, verify that each of the guessed strings are in S. If the verification is successful, then let $F = \{y_1, \ldots, y_m\}$ and proceed to Phase 2.

Phase 2:

Run M_A on x. Whenever M_A makes a query "$y \in B$?", M'_A instead makes the query "$y\#code(F) \in B'$?". Accept iff M_A accepts x.

Let $A' = L(M'_A, S \oplus B')$. It is not difficult to see that if $m = \| S^{\leq j} \|$ for some $j \geq p_B(p_A(|x|))$, then $x\#0^j\#0^m \in A' \Leftrightarrow x \in A$. Also, $A' \in \mathrm{NP}(S \oplus B') \subseteq \mathrm{NP}(S \oplus \mathrm{SAT})$.

Finally, consider the following algorithm for L:

- If z is an easy string for M_A, then answer YES iff $z \in A$.
- If z is a hard string for M_A, then $z \in Y \cup N$, and answer YES iff $z \in Y$.

Fact 3: $KM_L \in$ NP.

We are now ready to present the $\mathrm{P}^{(A \oplus SAT)}$ algorithm for L.

input x;
$Y := \emptyset$; $N := \emptyset$;
(1) $(* \; Y \subseteq HQM_L \cap A,\; N \subseteq HQM_L \cap \overline{A} \; *)$
while $x\#code(Y)\#code(N)\#\lambda \in PrefixHardQuery$ do
    $u := \lambda$;
(2) $(*$ u is a prefix of a computation leading to a string $z \in HQM_L - (Y \cup N) \; *)$
    while $(x\#code(Y)\#code(N)\#u0 \in PrefixHardQuery)$ or
        $(x\#code(Y)\#code(N)\#u1 \in PrefixHardQuery)$ do
        if $x\#code(Y)\#code(N)\#u0 \in PrefixHardQuery$ then $u := u0$
        else $u := u1$ end $(*$ if $*)$
    end; $(*$ while $*)$
    extract string z in the query "$z \in A$?" from the last configuration in u;
    if $z \in A$ then $Y := Y \cup \{z\}$ else $N := N \cup \{z\}$ end
end; $(*$ while $*)$
(3) $(* \; Y = HQM_L \cap A,\; N = HQM_L \cap \overline{A} \; *)$
if $x\#code(Y)\#code(N) \in KM_L$ then accept else reject end.

The loop invariants at (1) and (2) imply that condition (3) holds when execution of the algorithm reaches the final if-then-else statement. From this it is easy to see that the algorithm is correct. With respect to running time, the algorithm runs in polynomial time relative to $PrefixHardQuery \oplus KM_L \oplus A$. Since $PrefixHardQuery$ and KM_L are both in NP, the algorithm can be adjusted to run in polynomial time relative to $A \oplus SAT$, and it follows that $L \in \mathrm{P}(A \oplus SAT)$. $\square$

Our next theorem relocates sparse sets from $EL_3^{P,\Delta}$ to $EL_3^{P,\Theta}$. This will relocate all of the remaining $EL_3^{P,\Delta}$ results to $EL_3^{P,\Theta}$ as well.

membership in A. Once the two tables of queries to hard strings for M_A are completed, the oracle A can be replaced by these tables so that the computation of $M_L(x)$ becomes a standard NP computation, and SAT can be consulted one more time to see if the computation of $M_L(x)$ accepts. This proof technique was previously used by Book, Long, and Selman [BLS84].

We now present the details of the proof. First, for each string x, define the set $HQM_L(x) = \{z \mid M_L^A(x)$ queries string z and z is hard for $M_A\}$. Next, define the set $HardQuery$ as follows:
$x\#code(Y)\#code(N)\#C_1\#C_2\#\ldots\ \#C_j \in HardQuery \Leftrightarrow$

1. $Y \cap N = \emptyset$.
2. $j \geq 1$ and each $C_i, 1 \leq i \leq j$, is a configuration of $M_L(x)$.
3. If $C_i, 1 \leq i < j$, is a query configuration making the query "$z \in A$?", then the query is answered according to the following rules:
   - If z is an easy string for M_A, then answer YES iff $z \in A$.
   - If z is a hard string for M_A, then $z \in Y \cup N$, and answer YES iff $z \in Y$
4. C_1 is the initial configuration of $M_L(x)$, and $C_1, C_2, \ldots, C_j$ is a computation of $M_L(x)$ with respect to oracle answers as specified in 3.
5. C_j is an oracle configuration making a query "$z \in A$?", with z being a hard string for M_A and $z \notin Y \cup N$.

Note that when $x\#code(Y)\#code(N)\#C_1\#C_2\#\ldots\ \#C_j \in HardQuery$, configuration C_j is querying a hard string for M_A that is not yet in $Y \cup N$. Also, note that

Fact 1: $HardQuery \in$ P.

Now define the set
$PrefixHardQuery = \{x\#code(Y)\#code(N)\#u \mid (\exists v)\,[x\#code(Y)\#code(N)\#uv \in HardQuery]\}$ and note that

Fact 2: $PrefixHardQuery \in$ NP.

Finally, define the set KM_L such that $x\#code(Y)\#code(N) \in KM_L \Leftrightarrow Y \cap N = \emptyset$ and $M_L(x)$ accepts answering queries "$z \in A$?" as follows:

$$= \mathrm{NP}(\mathrm{P}(\Sigma^P_{k-2}(C \oplus \mathrm{SAT}))) = \mathrm{NP}(\Sigma^P_{k-2}(C \oplus \mathrm{SAT})) = \Sigma^P_{k-1}(C \oplus \mathrm{SAT}),$$

and $C \in EL_k^{P,\Sigma}$. $\square$

Proposition 5.3

1. $EL_1^{P,\Sigma} = EL_2^{P,\Theta} = EL_2^{P,\Delta}$.
2. *For all* $k \geq 3, EL_k^{P,\Theta} \cap \mathrm{NP} = L_k^{P,\Theta}$.

Proof.

1. In [AH92], it is shown that $EL_1^{P,\Sigma} = EL_2^{P,\Delta}$. By the previous lemma, $EL_1^{P,\Sigma} \subseteq EL_2^{P,\Theta} \subseteq EL_2^{P,\Delta}$. Thus, $EL_1^{P,\Sigma} = EL_2^{P,\Theta} = EL_2^{P,\Delta}$.
2. Let $L \in \mathrm{NP}$ and let $k \geq 3$. Then, $L \in EL_k^{P,\Theta} \Leftrightarrow \Theta_k^P(L) \subseteq \Theta_{k-1}^P(L \oplus \mathrm{SAT}) \Leftrightarrow \Theta(\Sigma_{k-1}^P(L)) \subseteq \Theta(\Sigma_{k-2}^P(L \oplus \mathrm{SAT})) \Leftrightarrow \Theta(\Sigma_{k-1}^P(L)) \subseteq \Theta(\Sigma_{k-2}^P(\mathrm{SAT})) \Leftrightarrow \Theta_k^P(L) \subseteq \Theta_k^P \Leftrightarrow L \in L_k^{P,\Theta}$. $\square$

Our next theorem relocates APT from $EL_3^{P,\Delta}$ to $EL_1^{P,\Sigma}$.

Theorem 5.4 $APT \subseteq EL_1^{P,\Sigma}$.

Proof. Let $A \in \mathrm{APT}$. Then there is a machine M_A and a polynomial p_A such that $A = L(M_A)$ and such that for all n,

$$\| \{x \mid |x| \leq n \text{ and } M_A(x) \text{ runs for more than } p_A(|x|) \text{ steps}\} \| \leq p_A(n).$$

We say that a string z *is a hard string for* M_A if and only if the running time of $M_A(z)$ exceeds $p_A(|z|)$ steps, and we say that z *is an easy string for* M_A otherwise.

Now let $L \in \mathrm{NP}^A$, and let M_L be a nondeterministic oracle machine, running in polynomial time p_L, such that $L = L(M_L, A)$. We need to show that $L \in \mathrm{P}^{(A \oplus SAT)}$. We begin with an informal description of how this is done: As M_L computes on an input string x relative to A, M_L can query strings that are both easy and hard for M_A. However, the number of distinct strings that M_L queries that are hard for M_A is bounded above by $p_A(p_L(|x|))$, a polynomial number of distinct hard strings. The $\mathrm{P}^{(A \oplus SAT)}$ algorithm for L uses SAT as as an oracle to deterministically navigate around the computation tree of $M_L(x)$ relative to A, finding each of the at most polynomially many query strings that are hard for M_A. Using A as an oracle, these strings are put into two tables based on

the polynomial-time hierarchy relative to A does not collapse to $\mathrm{P}^{\mathrm{NP}(A)[f(n)]}$. The notation $\mathrm{P}^{\mathrm{NP}(A)[f(n)]}$ means that the deterministic oracle machine is allowed, on inputs of length n, to make at most $f(n)$ queries to an oracle set in NP(A).

In Theorem 5.9, we use the oracle construction from Kadin to sharpen the $EL_2^{P,\Sigma}$ lower bound for our $EL_3^{P,\Theta}$ results. Letting f be a function such that for all constants $c > 0$, $f(n) < c \cdot \log n$ infinitely often, we construct a sparse set S such that $S \notin EL_3^{P,O(f)}$.[9] Thus, any future improvement in the $EL_3^{P,\Theta}$ results will require an extremely fine grain view of the extended low hierarchies.

We begin with the definition of the extended low hierarchy relative to the Σ, Δ, and Θ-levels of the polynomial-time hierarchy. This is followed by some basic properties of the hierarchy.

Definition 5.1

1. *[BBS86] For each $k \geq 1$, $EL_k^{P,\Sigma} = \{A \mid \Sigma_k^P(A) \subseteq \Sigma_{k-1}^P(A \oplus \mathrm{SAT})\}$.*
2. *[AH92] For each $k \geq 2$, $EL_k^{P,\Delta} = \{A \mid \Delta_k^P(A) \subseteq \Delta_{k-1}^P(A \oplus \mathrm{SAT})\}$.*
3. *For each $k \geq 2$, $EL_k^{P,\Theta} = \{A \mid \Theta_k^P(A) \subseteq \Theta_{k-1}^P(A \oplus \mathrm{SAT})\}$.*

Lemma 5.2 *For each $k \geq 2$, $EL_{k-1}^{P,\Sigma} \subseteq EL_k^{P,\Theta} \subseteq EL_k^{P,\Delta} \subseteq EL_k^{P,\Sigma}$.*

Proof. Let A be in $EL_{k-1}^{P,\Sigma}$. By definition, $\Sigma_{k-1}^P(\mathrm{A}) \subseteq \Sigma_{k-2}^P(A \oplus \mathrm{SAT})$. Thus we have

$$\Theta_k^P(A) = \Theta(\Sigma_{k-1}^P(A)) \subseteq \Theta(\Sigma_{k-2}^P(A \oplus \mathrm{SAT})) = \Theta_{k-1}^P(A \oplus \mathrm{SAT}),$$

and $A \in EL_k^{P,\Theta}$.

Let B be in $EL_k^{P,\Theta}$ so that $\Theta_k^P(B) \subseteq \Theta_{k-1}^P(B \oplus \mathrm{SAT})$. We have

$$\Delta_k^P(B) = \mathrm{P}(\Sigma_{k-1}^P(B)) \subseteq \mathrm{P}(\Theta_k^P(B)) \subseteq \mathrm{P}(\Theta_{k-1}^P(B \oplus \mathrm{SAT}))$$

$$= \mathrm{P}(\Theta(\Sigma_{k-2}^P(B \oplus \mathrm{SAT}))) = \mathrm{P}(\Sigma_{k-2}^P(B \oplus \mathrm{SAT})) = \Delta_{k-1}^P(B \oplus \mathrm{SAT}),$$

and $B \in EL_k^{P,\Delta}$.

Let C be in $EL_k^{P,\Delta}$ so that $\Delta_k^P(C) \subseteq \Delta_{k-1}^P(C \oplus \mathrm{SAT})$. We have

$$\Sigma_k^P(C) = \mathrm{NP}(\Sigma_{k-1}^P(C)) \subseteq \mathrm{NP}(\Delta_k^P(C)) \subseteq \mathrm{NP}(\Delta_{k-1}^P(C \oplus \mathrm{SAT}))$$

[9] The parameter $O(f)$ in the notation $EL_3^{P,O(f)}$ denotes that at most $O(f(n))$ queries can be made to the oracle set.

5 A Refinement of the Extended Low Hierarchies

By definition, the low hierarchies are restricted to sets in NP. To extend the notion of lowness to sets outside of NP, Balcázar, Book, and Schöning [BBS86] defined the extended low hierarchy relative to the Σ-classes of the polynomial-time hierarchy. They then proved, for example, that sparse sets and sets in P/*poly* are in $EL_3^{P,\Sigma}$, the third level of the extended low hierarchy. Recently, Allender and Hemachandra [AH92] introduced a refinement of the extended low hierarchy based on the Δ-levels of the polynomial-time hierarchy, and located the following classes in the bottom two levels of the refined hierarchy:

$EL_2^{P,\Delta}$: Tally sets, standard and general left cuts, and sets that are $\leq_m^P$-equivalent or $\leq_T^P$-equivalent to some tally set.

$EL_3^{P,\Delta}$: Sparse and co-sparse sets, APT, P-close, sets that are $\leq_m^P$-reducible or $\leq_{1-tt}^P$-reducible to some sparse set, and sets that are $\leq_T^P$-equivalent to some sparse set.

In this section we present a further refinement of the extended low hierarchy based on the Θ-levels of the polynomial-time hierarchy. We show that each of the classes just listed for $EL_3^{P,\Delta}$ are actually in $EL_3^{P,\Theta}$, level three of the newly refined extended low hierarchy. In addition, we prove a stronger result for APT, showing that APT is actually in $EL_1^{P,\Sigma}$. Since it is the case that $EL_1^{P,\Sigma} = EL_2^{P,\Theta} = EL_2^{P,\Delta}$, no further improvement is possible for the $EL_2^{P,\Delta}$ results.

Allender and Hemachandra also considered the optimality of their lowness results. Within the framework of an extended low hierarchy restricted to only Σ and Δ-classes, they proved their $EL_3^{P,\Delta}$ results optimal by constructing a sparse set S such that S is not in $EL_2^{P,\Sigma}$. When the Θ-classes of the extended low hierarchy are also considered, our results show that $EL_3^{P,\Delta}$ is not optimal for sparse sets. However, with attention restricted to Σ, Δ, and Θ-levels, $EL_3^{P,\Theta}$ is optimal for sparse sets by the Allender/Hemachandra construction.

When Kadin proved that the existence of a sparse $\leq_T^P$-complete set for NP collapses the polynomial-time hierarchy to Θ_2^P, he also constructed an oracle set A relative to which this collapse is essentially optimal. Specifically, he proved that if f is any function such that $f(n) < \log n$ infinitely often, then there is an oracle set A such that NP(A) has a sparse $\leq_T^P$-complete set, and such that

construct an oracle set relative to which NP $\cap$ P_{1-tt}(SPARSE) is not in $L_2^{P,\Sigma}$ without also establishing that co-sparse NP sets are not in $L_2^{P,\Sigma}$. We do not know if Theorem 4.3 can be extended to include NP $\cap$ E_T(SPARSE) as well.

Theorem 4.3 $\{S \in \mathrm{NP} \mid S$ *is co-sparse* $\} \subseteq L_2^{P,\Sigma} \Leftrightarrow \mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq L_2^{P,\Sigma}$.

Proof. Since $\{S \in \mathrm{NP} \mid S$ is co-sparse $\} \subseteq \mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE})$, if NP $\cap$ $\mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq L_2^{P,\Sigma}$, then $\{S \in \mathrm{NP} \mid S$ is co-sparse $\} \subseteq L_2^{P,\Sigma}$.

Now assume that $\{S \in \mathrm{NP} \mid S$ is co-sparse $\} \subseteq L_2^{P,\Sigma}$ and let $A \in \mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE})$. We need to show that $\Sigma_2^P(A) \subseteq \Sigma_2^P$. By Proposition 2.3, there are sparse sets S_1 and S_2 such that $A \leq_{1-tt}^P S_1 \oplus S_2$, $S_1 \in \mathrm{NP}$, and $S_2 \in$ co-NP. $A \leq_{1-tt}^P S_1 \oplus S_2$ implies that $\Sigma_2^P(A) \subseteq \Sigma_2^P(S_1 \oplus S_2)$. Using the fact that $S_1 \in \mathrm{NP}$, the proof techniques from Theorem 3.2 can be applied to show that $\Sigma_2^P(S_1 \oplus S_2) \subseteq \Sigma_2^P(S_2)$. $\overline{S_2}$ is a co-sparse set in NP and $\overline{S_2} \in L_2^{P,\Sigma}$ by hypothesis, so it follows that $\Sigma_2^P(\overline{S_2}) \subseteq \Sigma_2^P$ and that $\Sigma_2^P(S_2) \subseteq \Sigma_2^P$, since $\Sigma_2^P(S_2) = \Sigma_2^P(\overline{S_2})$. Therefore, $\Sigma_2^P(A) \subseteq \Sigma_2^P$ and $A \in L_2^{P,\Sigma}$. $\square$

Since $(p_i(n+k+2)^2) < 2^n$, no new strings of length n are in $\overline{S}$ so that $0^n\#0^{k+1} \notin cp_{\overline{S}}$ and also $M_i(0^n\#0^{k+1})$ accepts relative to $K(A_{i+1})$. By assigning all strings y, $n_i < |y| \leq n_{i+1}$, not yet reserved for A_{i+1} or $\overline{A_{i+1}}$, to $\overline{A_{i+1}}$, requirement R_i is satisfied.

Case 2: M_i rejects $(0^n\#0^{k+1})$ relative to $K(A_{i+1})$.

As noted above, at most $(p_i(n+k+2)^2)$ new strings were committed to A_{i+1} or $\overline{A_{i+1}}$. Since $(p_i(n+k+2)^2) < 2^n$, there is some string x, $|x| = n$, such that no strings in the set $W(x) = \{xy \mid |y| = n\}$ have been assigned to A_{i+1} or $\overline{A_{i+1}}$. Assign all of $W(x)$ to A_{i+1} and assign all other strings y, $n_i < |y| \leq n_{i+1}$, not yet reserved for A_{i+1} or $\overline{A_{i+1}}$, to $\overline{A_{i+1}}$. Then, exactly one string of length n has been added to $\overline{S}$, $0^n\#0^{k+1} \in cp_{\overline{S}}$, and $M_i(0^n\#0^{k+1})$ rejects relative to $K(A_{i+1})$. Thus, requirement R_i is satisfied. □

Corollary 4.2 *There exists a recursive set A such that*

1. $\mathrm{NP}^A \cap \text{P-close} \not\subseteq L_2^{P,\Delta}(A)$.
2. $\mathrm{NP}^A \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE}) \not\subseteq L_2^{P,\Delta}(A)$.
3. $\mathrm{NP}^A \cap \mathrm{E}_{1-tt}(\mathrm{SPARSE}) \not\subseteq L_2^{P,\Delta}(A)$.
4. $\mathrm{NP}^A \cap \mathrm{E}_{T}(\mathrm{SPARSE}) \not\subseteq L_2^{P,\Delta}(A)$.

For the classes considered in Corollary 3.5, we have the following sequence of containments:
$\{S \in \mathrm{NP} \mid S \text{ is co-sparse }\} \subseteq \mathrm{NP} \cap \text{P-close} \subseteq \mathrm{NP} \cap \mathrm{E}_{1-tt}(\mathrm{SPARSE}) \subseteq \mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE})$. Theorem 4.1 presents an oracle set relative to which the first of these classes is not in $L_2^{P,\Delta}$. Further, we have stated that we do not know how to construct an oracle set relative to which the first of these classes is not in $L_2^{P,\Sigma}$. This does not precisely address possible oracle constructions for the remaining classes such as $\mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE})$. For example, might Baker/Selman techniques suffice to construct an oracle set relative to which $\mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE})$ is not in $L_2^{P,\Sigma}$?

Our next theorem shows that each of these classes will belong to $L_2^{P,\Sigma}$ if and only if $\{S \in \mathrm{NP} \mid S \text{ is co-sparse }\}$ belongs to $L_2^{P,\Sigma}$. Thus, it is not possible to

$cp_{\overline{S}} \notin \Delta_2^P(A)$ by satisfying each requirement

$$R_i : \exists n, j \in \mathcal{N}[(0^n \# 0^j) \in cp_{\overline{S}} \Leftrightarrow M_i^{K(A)}(0^n \# 0^j) accepts].$$

Requirement R_i is satisfied during stage i of the construction. Prior to stage i, an integer n_i is determined such that all strings of length at most n_i have already been decided for A or $\overline{A}$, and no strings of length exceeding n_i have been decided for A or $\overline{A}$. A_i and $\overline{A_i}$ denote the strings placed into A and $\overline{A}$, respectively, prior to stage i. Also, let $\overline{S_i}$ denote the set $\overline{S}$ relative to A_i; that is, the set of strings in $\overline{S}$ by virtue of (5) when A_i is used in place of A. To start the construction, $n_1 = 0$, $A_1 = \{\lambda\}$, and $\overline{A_1} = \emptyset$.

Stage i:

Let $k = \| \overline{S_i} \|$. Let n be the smallest integer such that $n > n_i$ and such that $(p_i(n + k + 2)^2) < 2^n$. Since the construction ensures that $k \leq n_i$, such an n will exist. Let $n_{i+1} = (p_i(n + k + 2))^2$. Initially let $A_{i+1} = A_i$ and let $\overline{A_{i+1}} = \overline{A_i}$.

Run machine M_i on input $(0^n \# 0^{k+1})$ using oracle set $K(A_{i+1})$. Each time M_i enters the QUERY state making a query of the form "$< M, x, 0^t > \in K(A_{i+1})$?", perform the following process.

Process Q:

> If it is possible to add strings to A_{i+1}, obtaining set $\hat{A}$, so that $< M, x, 0^t > \in K(\hat{A})$, then do so. Then, take the leftmost accepting computation of $M(x)$ relative to $\hat{A}$ and put into A_{i+1} and $\overline{A_{i+1}}$ any strings queried, in this computation, that are in $\hat{A}$ and not in $\hat{A}$, respectively. In this way, $< M, x, 0^t > \in K(A)$.
>
> If it is not possible to add strings to A_{i+1} so that $M(x)$ accepts relative to an adjusted A_{i+1}, then it will be the case that $< M, x, 0^t > \notin K(A)$.

The original query "$< M, x, 0^t > \in K(A_{i+1})$?" is then answered consistently with the current A_{i+1}, as modified by Process Q.

Case 1: M_i accepts $(0^n \# 0^{k+1})$ relative to $K(A_{i+1})$.

> In this case, Process Q was invoked at most $p_i(n + k + 2)$ times and each invocation committed at most $p_i(n + k + 2)$ additional strings to A_{i+1}.

not necessarily any lower in the polynomial-time hierarchy. In turn, the cost of computing $cf_{\overline{S}}$ is Θ_3^P when $cp_{\overline{S}}$ is in Σ_2^P. This is the essential difficulty in trying to apply Kadin's proof techniques to obtain a sharper lowness result for co-sparse sets in NP.

When considering question (4), and in particular, when considering whether Kadin's proof techniques might be applicable to (4), it is reasonable to consider relativizations. Specifically, given that Kadin's proof techniques relativize, it is appropriate to ask if there is an oracle set relative to which (4) is not true. We have not been able to construct such an oracle, which involves constructing a set A and a sparse set S such that $S \in \text{co-NP}(A)$ and such that $\Sigma_2^P(A \oplus S) \not\subseteq \Sigma_2^P(A)$. Construction of such an A and S, separating the polynomial-time hierarchy at level two, does not seem to be possible by techniques in Baker and Selman [BS79] and in Angluin [Ang80], and remains an open problem. However, we do present, in Theorem 4.1, an oracle set relative to which co-sparse NP sets are not contained in $L_2^{P,\Delta}$, the low class just below $L_2^{P,\Sigma}$. As a corollary, there are oracle sets relative to which the results of Corollary 3.5 cannot be improved to $L_2^{P,\Delta}$ results.

Theorem 4.1 *There exists a recursive set A and a co-sparse set $S \in \text{NP}(A)$ such that $S \notin L_2^{P,\Delta}(A)$.*

Proof. A and S are constructed so that $cp_{\overline{S}}$ is not in $\Delta_2^P(A)$. Since $cp_{\overline{S}} \in \text{NP}(S)$, when $\overline{S}$ is sparse, and since $\text{NP}(S) \subseteq \text{NP}(A \oplus S) \subseteq \Delta_2^P(A \oplus S)$, it follows that $\Delta_2^P(A \oplus S) \not\subseteq \Delta_2^P(A)$. With this construction, notice also that the cost of computing $cf_{\overline{S}}$ cannot possibly be $\Delta_2^P(A)$ since otherwise, $cp_{\overline{S}}$ would be in $\Delta_2^P(A)$. Hence, this oracle construction simply exploits the difficulty that occurs when attempting to apply Kadin's proof techniques to obtain a sharper lowness result for co-sparse NP sets.

The construction of A is done by a standard initial segments argument in the spirit of Baker, Gill, and Solovay [BGS75]. The set S is defined so that

(5) $(\forall x \in \Sigma^*)[x \in \overline{S} \Leftrightarrow (\forall y \in \Sigma^*)[|y| = |x| \Rightarrow xy \in A]]$.

Clearly, $\overline{S} \in \text{co-NP}(A)$, and the construction will ensure that for all n, $\| \overline{S}^{=n} \| \leq 1$, so that $\overline{S}$ is sparse. Diagonalization is used to ensure that $cp_{\overline{S}} \notin \Delta_2^P(A)$. Let M_1, M_2, $M_3, \ldots$ be an effective enumeration of deterministic oracle machines with machine M_i running in polynomial time p_i. The construction guarantees that

4 Optimality of Lowness Results and Relativizations

An interesting open question is whether the lowness results from the last section are optimal. To be specific, consider just the two results:

(1) Sparse NP sets are in $L_2^{P,\Theta}$.
(2) Co-sparse NP sets are in $L_3^{P,\Theta}$.

The next low class below $L_2^{P,\Theta}$ is $L_1^{P,\Sigma}$, and below $L_3^{P,\Theta}$ is $L_2^{P,\Sigma}$. Corresponding to (1) and (2) we have the questions:

(3) Are sparse NP sets in $L_1^{P,\Sigma}$?
(4) Are co-sparse NP sets in $L_2^{P,\Sigma}$?

Allender and Hemachandra[AH92] also considered questions (3) and (4). For (3), they noted that there is an oracle set A relative to which there are sparse sets in NP^A that are not in $L_1^{P,\Sigma}(A)$. However, we believe that question (3) is more appropriately considered in its unrelativized form. Recall that $L_1^{P,\Sigma} = \mathrm{NP} \cap \text{co-NP}$. Thus, question (3) asks if every sparse set in NP is also in co-NP. Using standard proof techniques, as presented in Balcázar, Díaz, and Gabarró [BDG88] for example, it is easy to show that every sparse set in NP is also in co-NP if and only if the complexity class NEXT = NTIME($2^{O(n)}$) is closed under complementation. The closure of NEXT under complementation is an important open question about nondeterministic time classes, so any stronger lowness result for sparse NP sets would require a major breakthrough in complexity theory. Similarly, for each of the classes considered in Corollary 3.3, an improvement to $L_1^{P,\Sigma}$ is possible if and only if NEXT = co-NEXT.

For question (4), the situation is not quite so clear. On the one hand, when S is a co-sparse set in NP, the proof of Corollary 3.5 did not actually take advantage of the fact that $\overline{S}$ is a sparse set in co-NP. Instead, it was simply noted that $\overline{S} \in$ co-NP implies that $\overline{S} \in \Sigma_2^P$ and then Theorem 3.2 was applied. No use was made of the fact that $\overline{S}$ is in co-NP in addition to Σ_2^P.

On the other hand, the proof of Theorem 3.2 is based on Kadin's proof techniques, and recall that when S is a sparse set in NP, then cp_S is also in NP. This is important in Kadin's proof techniques since cp_S is then used to compute cf_S at a cost of Θ_2^P. However, when S is a co-sparse set in NP, then $cp_{\overline{S}}$ is in Σ_2^P and

Corollary 3.8 *[Kad88] If there exists a co-sparse $\leq_T^{SN}$-complete set for NP, then* PH $\subseteq \Theta_3^P$.

Proof. By Corollary 3.5, co-sparse sets in NP are in $L_3^{P,\Theta}$ and by Theorem 3.6, the polynomial-time hierarchy collapses to Θ_3^P. $\square$

$\cup$ co-NP). In this case, Castro and Seara [CS92] have shown that A belongs to $L_2^{P,\Theta}$.

The situation for general left cuts and standard left cuts of real numbers is similar.[8] Proofs that NP sets, that are either general or standard left cuts of some real number x, are in $L_2^{P,\Delta}$are based on the fact that standard and general left cuts are in E_T(TALLY). Thus, the analysis for NP $\cap$ E_T(TALLY) applies as well to NP sets that are standard or general left cuts.

The literature includes many results showing that the polynomial-time hierarchy collapses under the assumption that NP has sparse or co-sparse complete sets with respect to various reducibilities. Some of these results can be explained by noting that sparse or co-sparse sets in NP are in the low hierarchy, while complete sets in NP are in the high hierarchy. Thus, if a sparse or co-sparse set in NP is a complete set for NP with respect to some reduction, then by Proposition 2.11, Theorem 2.13, and Corollaries 3.3 and 3.5, the polynomial-time hierarchy collapses.

The following theorem summarizes these facts.

Theorem 3.6

1. *For* $k \geq 1$, $[A \in L_k^{P,\Sigma}$ *and* co-NP $\subseteq \mathrm{NP}^A \Rightarrow \mathrm{PH} = \Sigma_k^P]$.
2. *For* $k \geq 2$, $[A \in L_k^{P,\Delta}$ *and* co-NP $\subseteq \mathrm{NP}^A \Rightarrow \mathrm{PH} = \Delta_k^P]$.
3. *For* $k \geq 2$, $[A \in L_k^{P,\Theta}$ *and* co-NP $\subseteq \mathrm{NP}^A \Rightarrow \mathrm{PH} = \Theta_k^P]$.

Proof. If A is in the low hierarchy, then A is in NP. Also, co-NP $\subseteq \mathrm{NP}^A$ implies A is $\leq_T^{SN}$-hard for NP. Thus, A is $\leq_T^{SN}$-complete for NP; that is, $A \in H_1^{P,\Sigma}$. The collapses of the polynomial-time hierarchy follow immediately from Proposition 2.11 and Theorem 2.13. $\square$

Corollary 3.7 *[Kad87] If there exists a sparse* $\leq_T^{SN}$*-complete set for NP, then* PH $\subseteq \Theta_2^P$.

Proof. By Corollary 3.3, sparse sets in NP are in $L_2^{P,\Theta}$ and by Theorem 3.6, the polynomial-time hierarchy collapses to Θ_2^P. $\square$

[8]The definitions of standard left cut and general left cut can be found in [Ko86].

Theorem 3.4 *[BS] If $A \in \mathrm{NP} \cap (\text{co-NP}/log)$, then there exists a sparse set $S \in \mathrm{NP}$ such that $A \in \text{co-NP}(S)$.*

Letting $A \in \mathrm{NP} \cap (\text{co-NP}/log)$, by Theorem 3.4, $A \in \text{co-NP}(S)$ for some sparse set $S \in \mathrm{NP}$. Since $A \in \mathrm{NP}$, we know that $A \in \mathrm{NP}(S) \cap \text{co-NP}(S)$, i.e., $A \leq_T^{SN} S$. By 1, $S \in L_2^{P,\Theta}$ and by Corollary 2.17, $A \in L_2^{P,\Theta}$. $\square$

We should point out that when Balcázar and Schöning proved that $\mathrm{NP} \cap (\text{co-NP}/log) \subseteq L_2^{P,\Delta}$, they remarked that Kadin's proof techniques could be used to obtain a sharper lowness result based on the Θ-classes of the polynomial-time hierarchy. Part 6 of Corollary 3.3 merely provides a formal statement of their remark.

Corollary 3.5

1. $\{S \in \mathrm{NP} \mid S$ *is co-sparse* $\} \subseteq L_3^{P,\Theta}$.
2. $\mathrm{NP} \cap \text{P-close} \subseteq L_3^{P,\Theta}$.
3. $\mathrm{NP} \cap \mathrm{P}_{1-tt}(\mathrm{SPARSE}) \subseteq L_3^{P,\Theta}$.
4. $\mathrm{NP} \cap \mathrm{E}_{1-tt}(\mathrm{SPARSE}) \subseteq L_3^{P,\Theta}$.
5. $\mathrm{NP} \cap \mathrm{E}_T(\mathrm{SPARSE}) \subseteq L_3^{P,\Theta}$.

Proof. Considering 1, let $S \in \mathrm{NP}$ be a co-sparse set. Then, $\overline{S}$ is a sparse set in Σ_2^P and 1 follows directly from Theorem 3.2. Considering 2, let $A \in \mathrm{NP} \cap$ P-close. By Proposition 2.7, there are sparse sets S_1 and S_2 in Σ_2^P such that $A \leq_{1-tt}^P S_1 \oplus S_2$. Applying Theorem 3.2 and Corollary 2.18, $A \in L_3^{P,\Theta}$. 3 and 4 follow similarly using Proposition 2.3, Corollary 2.18, and Theorem 3.2.

Finally, let $A \in \mathrm{NP} \cap \mathrm{E}_T(\mathrm{SPARSE})$. Then, there is a sparse set $S \in \Delta_2^P$ such that $A \equiv_T^P S$. $S \in \Delta_2^P$ implies $S \in \Sigma_2^P$ and applying Corollary 2.18 and Theorem 3.2, $A \in L_3^{P,\Theta}$. $\square$

Recall that Allender and Hemachandra proved that $\mathrm{NP} \cap \mathrm{E}_T(\mathrm{TALLY})$ is in $L_2^{P,\Delta}$, but that we do not know if $\mathrm{NP} \cap \mathrm{E}_T(\mathrm{TALLY})$ is in $L_2^{P,\Theta}$. To better understand this problem, let $A \in \mathrm{NP} \cap \mathrm{E}_T(\mathrm{TALLY})$. If there is a tally set $T \in \mathrm{NP}$ such that $A \equiv_T^P T$, then $A \in L_2^{P,\Theta}$. On the other hand, if every tally set T such that $A \equiv_T^P T$ is in Δ_2^P - Θ_2^P, then $A \notin L_2^{P,\Theta}$. Finally, if neither of these two cases occur, then for every tally set T such that $A \equiv_T^P T$, $T \in \Theta_2^P$ - (NP

Now define the set B_3 as follows:

$$y\#0^j\#0^m \in B_3 \Leftrightarrow \exists code(F)[0^j\#0^m\#code(F) \in \hat{S} \text{ and } y\#code(F) \in B_2].$$

Set B_3 corresponds to the idea of nondeterministically enumerating an initial segment of S given a census value. It is not difficult to show, using standard techniques, that $B_3 \in \Sigma_k^P$. Combining (5) and (6), we have that

(7) If $j \geq p_1(|y|+kq_1(|y|)+k)$ and if $m = \| S^{\leq j} \|$, then $y\#0^j\#0^m \in B_3 \Leftrightarrow y \in B_1$.

Finally, let M_L' be a modified version of M_L that works as follows:
On input x with $|x| = n$, M_L' operates in two phases:

Phase 1:

Letting $j = p_1(p_L(|x|) + kq_1(p_L(|x|)) + k)$, M_L' makes $O(\log n)$ queries to $cp_S \in \Sigma_k^P$ to compute $0^m = cf_S(0^j)$.

Phase 2:

M_L' begins to run M_L on input x. Whenever M_L would make the query "$y \in B_1$?", M_L' instead makes the query "$y\#0^j\#0^m \in B_3$?".

Considering property (7) and phase 1 of the computation of M_L', it is clear that $L = L(M_L', cp_S \oplus B_3)$ and that $L \in \Theta_{k+1}^P$.[6] $\square$

Corollary 3.3

1. $\{S \in \text{NP} \mid S \text{ is sparse } \} \subseteq L_2^{P,\Theta}$.
2. $\{T \in \text{NP} \mid T \text{ is a tally set } \} \subseteq L_2^{P,\Theta}$.
3. $\text{NP} \cap \text{APT} \subseteq L_2^{P,\Theta}$.
4. $\text{NP} \cap \text{P}_m(\text{SPARSE}) \subseteq L_2^{P,\Theta}$.
5. $\text{NP} \cap \text{P}_m(\text{TALLY}) \subseteq L_2^{P,\Theta}$.
6. $\text{NP} \cap (\text{co-NP}/log) \subseteq L_2^{P,\Theta}$.

Proof. 1 and 2 follow immediately from Theorem 3.2. 3, 4, and 5 follow from Propositions 2.2 and 2.6, Corollary 2.18, and Theorem 3.2. The proof of 6 is based on the following interesting theorem from Balcázar and Schöning [BS] [7].

[6] Another line of argument for this proof is to first show that $\Sigma_k^P(S) \subseteq \Theta_{k+1}^P$ when S is a sparse set in Σ_k^P. This would involve essentially the same proof as was just presented and is the direct generalization of (3) from Kadin's proof. Now applying Castro and Seara's [CS92] new result, we have $\Theta_{k+1}^P = \Theta(\Sigma_k^P(S)) \subseteq \Theta(\Theta_{k+1}^P) = \Theta_{k+1}^P$.

[7] A set A is in co-NP/*log* if and only if $(\exists B \in \text{co-NP})\ (\exists f : \{0\}^* \to \Sigma^*)\ (\exists c > 0)\ (\forall x \in \Sigma^*)\ [|f(0^{|x|})| \leq c \cdot \log(|x|) \text{ and } x \in A \Leftrightarrow x\#f(0^{|x|}) \in B]$.

We now present the detailed proof using the quantifier characterization of the polynomial-time hierarchy. Let M_L be a deterministic polynomial time oracle machine, running in time p_L, and making at most $O(\log n)$ queries. Let B_1 be an oracle set in $\Sigma_k^P(S)$ such that $L = L(M_L, B_1)$. Since $B_1 \in \Sigma_k^P(S)$, there is a polynomial q_1 and a deterministic polynomial time oracle machine M_1 running in time p_1 such that

$$(\forall y)[y \in B_1 \Leftrightarrow \exists_{q_1} y_1 \forall_{q_1} y_2 \ldots Q_{q_1} y_k [M_1^S(y\#y_1\#\ldots\#y_k)\ accepts]].$$

Let M_2 be a deterministic machine that behaves as follows:

on input $(y\#code(F)\#y_1\#\ldots\#y_k)$:
if one or more of $y_1, y_3, \ldots$ has length $> q_1(|y|)$ then
    reject
elsif one or more of $y_2, y_4, \ldots$ has length $> q_1(|y|)$ then
    accept
else
    accept $\Leftrightarrow M_1^F(y\#y_1\#\ldots\#y_k)$ accepts
end.

Now define the set B_2 such that

$$y\#code(F) \in B_2 \Leftrightarrow \exists_{q_1} y_1 \forall_{q_1} y_2 \ldots Q_{q_1} y_k [M_2(y\#code(F)\#y_1\#\ldots\#y_k)\ accepts].$$

Set B_2 corresponds to the idea of using an initial segment of S in place of S. Note that

(5) If $F = S^{\leq j}$ for some $j \geq p_1(|y| + kq_1(|y|) + k)$, then $y\#code(F) \in B_2 \Leftrightarrow y \in B_1$, since $M_1^S(y\#y_1\#\ldots\#y_k)$ and $M_1^F(y\#y_1\#\ldots\#y_k)$ receive exactly the same oracle answers. Also note that $B_2 \in \Sigma_k^P$.

Next we need a variant, $\hat{S}$, of the set S:

$$\hat{S} = \{0^j\#0^m\#code(F) \mid F \subseteq S^{\leq j}\ and\ \|F\| = m\}.$$

Note that

(6) If $m = \|S^{\leq j}\|$, then $0^j\#0^m\#code(F) \in \hat{S} \Leftrightarrow F = S^{\leq j}$. Also note that $\hat{S} \in \Sigma_k^P$.

the same way that Book and Tang's theorem generalized (1). In fact, the proof of Theorem 3.2 appears to have been known to Book and Tang, although they did not make this explicit in the statement of their theorem.

Theorem 3.2 *For all $k \geq 1$, if S is a sparse set in Σ_k^P, then $\Theta_{k+1}^P(S) \subseteq \Theta_{k+1}^P$.*

Proof. Let S be a sparse set in Σ_k^P for some $k \geq 1$. We need to show that $\Theta_{k+1}^P(S) \subseteq \Theta_{k+1}^P$; that is, letting L be an arbitrary language in $\Theta_{k+1}^P(S)$, we must show that L is in Θ_{k+1}^P.

We begin with an informal oracle machine/oracle set description of how the proof works. Intuitively, we can think of L as being accepted by a Θ-machine, M_L, using an oracle set, A_k, from $\Sigma_k^P(S)$. Similarly, A_k is accepted by an oracle machine NM_k using an oracle set A_{k-1} from $\Sigma_{k-1}^P(S)$, and so on. This sequence of oracle sets and oracle machines eventually leads to an oracle set A_2 accepted by an oracle machine NM_2 using an oracle set A_1 in $\mathrm{NP}(S)$, accepted by machine NM_1.

The Θ_{k+1}^P-algorithm for L starts with a modified version, M_L', of the Θ-machine M_L. On an input x, M_L' first makes $O(\log |x|)$ queries to cp_S to compute the census of an appropriate initial segment of S. Then, M_L' begins to operate just as M_L, except that M_L' passes this census information to a machine NM_k' whenever M_L would make a query to A_k. The machine NM_k' is a modified version of NM_k that operates in two phases. Given an input string and a census value, NM_k' enters an initial phase in which it attempts to nondeterministically enumerate an initial segment of S of size equal to the census value. This enumeration uses the Σ_k^P algorithm for S. If successful, NM_k' then begins to operate just like NM_k, except that whenever NM_k' would query A_{k-1}, NM_k' passes the initial segment of S to a machine NM_{k-1}'. In turn, each of the oracle machines continue to pass the initial segment to the next lower machine until finally reaching NM_1'. NM_1' uses the initial segment of S in place of S. The entire process results in a Θ_{k+1}^P-algorithm for L.

Essentially, this algorithm removes the oracle set S by computing the census function for S and then nondeterministically enumerating an initial segment of S. The cost of computing the census function is Θ_{k+1}^P: $O(\log n)$ queries to the census predicate cp_S which is in Σ_k^P. The cost of nondeterministically enumerating an initial segment of S is Σ_k^P.

Kadin later strengthened Mahaney's result showing that if NP has a sparse $\leq_T^{SN}$-complete set, then the polynomial-time hierarchy collapses to Θ_2^P.[4] As in Mahaney's proof, Kadin's proof breaks into two separate steps.

(3) If $S \in$ NP and S is sparse, then NP$(S) \subseteq \Theta_2^P$.

(4) If S is $\leq_T^{SN}$-complete for NP, then NP$(S) = \Sigma_2^P$.

Once again, (3) addresses the lowness of sparse sets and strongly suggests that sparse NP sets may actually belong to $L_2^{P,\Theta}$. We note that an $L_2^{P,\Theta}$ result does not follow quite as easily as did the $L_2^{P,\Delta}$ result: writing the conclusion of (3) as

$$\Sigma_1^P(S) \subseteq \Theta_2^P$$

and applying $\Theta(\ )$ to both sides of the containment gives

$$\Theta_2^P(S) = \Theta(\Sigma_1^P(S)) \subseteq \Theta(\Theta_2^P).$$

To complete the argument for the Θ-lowness of sparse sets, we use the fact that Θ_2^P consists of exactly those languages that are polynomial time truth-table reducible to languages in NP. It follows that $\Theta(\Theta_2^P) \subseteq \Theta_2^P$ and that sparse sets are in $L_2^{P,\Theta}$.[5]

We improve almost all of the lowness results for Δ-classes of the low hierarchy, obtaining instead corresponding Θ-lowness results:

$L_2^{P,\Theta}$: Sparse and tally NP sets, NP $\cap$ (co-NP/*log*), NP $\cap$ APT, NP sets that are $\leq_m^P$-reducible to some sparse or some tally set.

$L_3^{P,\Theta}$: co-sparse NP sets, NP $\cap$ P-close, NP sets that are $\leq_{1-tt}^P$-reducible to some sparse set, and NP sets that are $\leq_{1-tt}^P$-equivalent or that are $\leq_T^P$-equivalent to some sparse set.

Note that the only classes of sets from $L_2^{P,\Delta}$ that we do not know to be in $L_2^{P,\Theta}$ are NP sets that are $\leq_T^P$-equivalent to some tally set and NP sets that are either standard or general left cuts for some real number x.

Each of these lowness results follow from our next theorem. The proof of Theorem 3.2 just uses Kadin's proof techniques to generalize (3) from above, in

[4]Kadin used the hypothesis that $S \in$ NP, S is sparse, and co-NP $\subseteq$ NP(S). This is equivalent to assuming that S is sparse and $\leq_T^{SN}$-complete for NP.

[5]Castro and Seara [CS92] have recently shown the more general result that for $k \geq 2$, $\Theta(\Theta_k^P) \subseteq \Theta_k^P$.

3 Classes in the Θ-levels of the Low Hierarchy

Schöning[Sch86b], Ko and Schöning[KS85], Balcázar and Book[BB86], Balcázar and Schöning[BS], and Allender and Hemachandra[AH92] have located the following classes of sets in the Δ-levels of the low hierarchy.

$L_2^{P,\Delta}$: Sparse and tally NP sets, NP ∩ (co-NP/*log*), NP ∩ APT, NP sets that are standard or general left cuts for some real number x, NP sets that are $\leq_m^P$-reducible to some sparse or some tally set, and NP sets that are $\leq_T^P$-equivalent to some tally set.

$L_3^{P,\Delta}$: co-sparse NP sets, NP ∩ P-close, NP sets that are $\leq_{1-tt}^P$-reducible to some sparse set, and NP sets that are $\leq_{1-tt}^P$-equivalent or that are $\leq_T^P$-equivalent to some sparse set.

The basic proof technique for establishing the Δ-lowness of these sets comes from a theorem by Mahaney in which he proved that if NP has a sparse $\leq_T^P$-complete set, then the polynomial-time hierarchy collapses to Δ_2^P. Mahaney's proof breaks into two separate parts:

(1) If $S \in$ NP and S is sparse, then NP(S) $\subseteq \Delta_2^P$.

(2) If S is $\leq_T^P$-complete for NP, then NP(S) = Σ_2^P.

Part (1) is actually the portion of Mahaney's proof that addresses the lowness of sparse NP sets. To see this clearly, consider (1) and write its conclusion as

$$\Sigma_1^P(S) \subseteq \Delta_2^P.$$

Now apply P( ) to both sides of this containment to get

$$\Delta_2^P(S) = \mathrm{P}(\Sigma_1^P(S)) \subseteq \mathrm{P}(\Delta_2^P) = \Delta_2^P$$

and immediately, $S \in L_2^{P,\Delta}$ and sparse NP sets are in $L_2^{P,\Delta}$.

This type of result has been generalized to arbitrary levels of the polynomial-time hierarchy in the following theorem by Book and Tang. Each of the lowness results stated above follow easily from this theorem and its proof techniques.

Theorem 3.1 *[BT89] For $k \geq 1$, if S is a sparse set in Σ_k^P, then $\Delta_{k+1}^P(S) \subseteq \Delta_{k+1}^P$.*

The proof of (2) is by induction on k, with (1) being the base of the induction. Proceeding to the inductive step, assume (2) true for some arbitrary $k \geq 1$ and consider $k+1$. We have

$$\Sigma_{k+1}^P(A) = \mathrm{NP}(\Sigma_k^P(A)) \subseteq \mathrm{NP}(\Sigma_k^P(B)) = \Sigma_{k+1}^P(B),$$

proving (2).

Now let $k \geq 2$ and consider $\Theta_k^P(A)$.

$$\Theta_k^P(A) = \Theta(\Sigma_{k-1}^P(A)) \subseteq \Theta(\Sigma_{k-1}^P(B)) = \Theta_k^P(B).$$

□

Corollary 2.16 *For $k \geq 1$, if $A \leq_T^P B$, then $\Theta_k^P(A) \subseteq \Theta_k^P(B)$.*

Corollary 2.17 *For $k \geq 2$, $L_k^{P,\Theta}$ is closed downward under $\leq_T^{SN}$-reductions.*

Corollary 2.18 *For $k \geq 1$, $L_k^{P,\Theta}$ is closed downward under $\leq_T^P$-reductions.*

2. Let A be in $H_{k-1}^{P,\Sigma}$. By definition, $\Sigma_k^P \subseteq \Sigma_{k-1}^P(\mathrm{A})$. Thus we have

$$\Theta_{k+1}^P = \Theta(\Sigma_k^P) \subseteq \Theta(\Sigma_{k-1}^P(A)) = \Theta_k^P(A),$$

and $A \in H_k^{P,\Theta}$.

Let B be in $H_k^{P,\Theta}$ so that $\Theta_{k+1}^P \subseteq \Theta_k^P(B)$. We have

$$\Delta_{k+1}^P = \mathrm{P}(\Sigma_k^P) \subseteq \mathrm{P}(\Theta_{k+1}^P) \subseteq \mathrm{P}(\Theta_k^P(B))$$

$$= \mathrm{P}(\Theta(\Sigma_{k-1}^P(B))) = \mathrm{P}(\Sigma_{k-1}^P(B)) = \Delta_k^P(B),$$

and $B \in H_k^{P,\Delta}$.

3. Suppose $L_k^{P,\Theta} \cap H_k^{P,\Theta} \neq \emptyset$. By definition, there is a set A in NP such that

$$\Theta_{k+1}^P \subseteq \Theta_k^P(A) \subseteq \Theta_k^P,$$

so that $\Theta_{k+1}^P = \Theta_k^P$. Thus, $\mathrm{PH} = \Theta_k^P$.

4. For $k = 1$, $\mathrm{PH} = \Theta_1^P = \mathrm{P}$. Therefore, $L_k^{P,\Theta} = H_k^{P,\Theta} = \mathrm{NP} = \mathrm{P}$.

For $k \geq 2$, let A be in NP. Recall that $\Theta_k^P \subseteq \Theta_k^P(A) \subseteq \Theta_{k+1}^P$. From the assumption that $\mathrm{PH} = \Theta_k^P$, it follows that $\Theta_k^P = \Theta_k^P(A) = \Theta_{k+1}^P$, so that $A \in L_k^{P,\Theta}$ and $A \in H_k^{P,\Theta}$. □

Corollary 2.14

1. $L_1^{P,\Theta} = \mathrm{P}$.
2. $H_1^{P,\Theta} = \{A \in \mathrm{NP} \mid A \textit{ is } \leq_T^P\textit{-complete } \}$.

Proof. Follows immediately from Proposition 2.10 and Theorem 2.13. □

We conclude this section with some technical lemmas that will be useful later.

Lemma 2.15 *For $k \geq 2$, if $A \leq_T^{SN} B$, then $\Theta_k^P(A) \subseteq \Theta_k^P(B)$.*

Proof. Let $A \leq_T^{SN} B$. Recall that

(1) If $A \leq_T^{SN} B$, then $\mathrm{NP}(A) \subseteq \mathrm{NP}(B)$.

We first note that (1) can be generalized to

(2) For $k \geq 1$, if $A \leq_T^{SN} B$, then $\Sigma_k^P(A) \subseteq \Sigma_k^P(B)$.

Proposition 2.11 *[Sch83, Sch86b]*

1. *For* $k \geq 0$, *if* $\mathrm{PH} \neq \Sigma_k^P$, *then* $L_k^{P,\Sigma} \cap H_k^{P,\Sigma} = \emptyset$.
2. *For* $k \geq 0$, *if* $\mathrm{PH} = \Sigma_k^P$, *then* $L_k^{P,\Sigma} = H_k^{P,\Sigma} = \mathrm{NP}$.
3. *For* $k \geq 1$, *if* $\mathrm{PH} \neq \Delta_k^P$, *then* $L_k^{P,\Delta} \cap H_k^{P,\Delta} = \emptyset$.
4. *For* $k \geq 1$, *if* $\mathrm{PH} = \Delta_k^P$, *then* $L_k^{P,\Delta} = H_k^{P,\Delta} = \mathrm{NP}$.

We now introduce a further refinement of the low and high hierarchies based on the Θ-levels of the polynomial-time hierarchy. First note that for every set $A \in \mathrm{NP}$ and $k \geq 2$,

$$\Theta_k^P \subseteq \Theta_k^P(A) \subseteq \Theta_{k+1}^P.$$

Definition 2.12 *Relative to the* Θ*-classes of the polynomial-time hierarchy, the* Θ*-classes of the low hierarchy are*

$$\{L_k^{P,\Theta} \mid k \geq 1\}, \text{ where } L_k^{P,\Theta} = \{A \in \mathrm{NP} \mid \Theta_k^P(A) \subseteq \Theta_k^P\},$$

and the Θ*-classes of the high hierarchy are*

$$\{H_k^{P,\Theta} \mid k \geq 1\}, \text{ where } H_k^{P,\Theta} = \{A \in \mathrm{NP} \mid \Theta_{k+1}^P \subseteq \Theta_k^P(A)\}.$$

The following theorem gives the basic relationships between the Θ-classes of the low and high hierarchies and the Σ and Δ-classes of the low and high hierarchies.

Theorem 2.13 *For each* $k \geq 1$*:*

1. $L_{k-1}^{P,\Sigma} \subseteq L_k^{P,\Theta} \subseteq L_k^{P,\Delta} \subseteq L_k^{P,\Sigma}$.
2. $H_{k-1}^{P,\Sigma} \subseteq H_k^{P,\Theta} \subseteq H_k^{P,\Delta} \subseteq H_k^{P,\Sigma}$.
3. $\mathrm{PH} \neq \Theta_k^P$ *implies* $L_k^{P,\Theta} \cap H_k^{P,\Theta} = \emptyset$.
4. $\mathrm{PH} = \Theta_k^P$ *implies* $L_k^{P,\Theta} = H_k^{P,\Theta} = \mathrm{NP}$.

Proof.

1. By Proposition 2.10, we only need to verify that $L_{k-1}^{P,\Sigma} \subseteq L_k^{P,\Theta} \subseteq L_k^{P,\Delta}$. Let A be in $L_{k-1}^{P,\Sigma}$. By definition, $\Sigma_{k-1}^P(\mathrm{A}) \subseteq \Sigma_{k-1}^P$. Thus we have

   $$\Theta_k^P(A) = \Theta(\Sigma_{k-1}^P(A)) \subseteq \Theta(\Sigma_{k-1}^P) = \Theta_k^P,$$

   and $A \in L_k^{P,\Theta}$.

   Let B be in $L_k^{P,\Theta}$ so that $\Theta_k^P(B) \subseteq \Theta_k^P$. We have

   $$\Delta_k^P(B) = \mathrm{P}(\Sigma_{k-1}^P(B)) \subseteq \mathrm{P}(\Theta_k^P(B)) \subseteq \mathrm{P}(\Theta_k^P) = \mathrm{P}(\Theta(\Sigma_{k-1}^P)) = \mathrm{P}(\Sigma_{k-1}^P) = \Delta_k^P,$$

   and $B \in L_k^{P,\Delta}$.

where Q is $\exists$ if k is odd and $\forall$ otherwise, and $\exists_q y_1$ ($\forall_q y_2$) denotes $\exists y_1 : |y_1| \leq q(|x|)$ ($\forall y_2 : |y_2| \leq q(|x|)$, respectively).

For each set A in NP and $k \geq 1$, the following containments hold:

$$\Sigma_k^P \subseteq \Sigma_k^P(A) \subseteq \Sigma_{k+1}^P \text{ and } \Delta_k^P \subseteq \Delta_k^P(A) \subseteq \Delta_{k+1}^P.$$

If $A \in$ P, then the left containments become equality. If A is an NP-complete set, then the right containments become equality. These containments provide the basis for the definitions of the low and high hierarchies within NP, relative to the Σ and Δ-classes of the polynomial-time hierarchy.

Definition 2.9

1. *[Sch83] Relative to the Σ-classes of the polynomial-time hierarchy, the Σ-classes of the low hierarchy are*

$$\{L_k^{P,\Sigma} \mid k \geq 0\}, \text{ where } L_k^{P,\Sigma} = \{A \in \text{NP} \mid \Sigma_k^P(A) \subseteq \Sigma_k^P\},$$

and the Σ-classes of the high hierarchy are

$$\{H_k^{P,\Sigma} \mid k \geq 0\}, \text{ where } H_k^{P,\Sigma} = \{A \in \text{NP} \mid \Sigma_{k+1}^P \subseteq \Sigma_k^P(A)\}.$$

2. *[Sch86b] Relative to the Δ-classes of the polynomial-time hierarchy, the Δ-classes of the low hierarchy are*

$$\{L_k^{P,\Delta} \mid k \geq 1\}, \text{ where } L_k^{P,\Delta} = \{A \in \text{NP} \mid \Delta_k^P(A) \subseteq \Delta_k^P\},$$

and the Δ-classes of the high hierarchy are

$$\{H_k^{P,\Delta} \mid k \geq 1\}, \text{ where } H_k^{P,\Delta} = \{A \in \text{NP} \mid \Delta_{k+1}^P \subseteq \Delta_k^P(A)\}.$$

Propositions 2.10 and 2.11 list some of the basic properties of the low and high hierarchies. Proposition 2.10 indicates that the high classes correspond to classes of complete sets, while Proposition 2.11 states that the low and high hierarchies are disjoint if and only if the polynomial-time hierarchy is properly infinite.

Proposition 2.10 *[Sch83, Sch86b]*

1. *For each $k \geq 1$, $L_{k-1}^{P,\Sigma} \subseteq L_k^{P,\Delta} \subseteq L_k^{P,\Sigma}$ and $H_{k-1}^{P,\Sigma} \subseteq H_k^{P,\Delta} \subseteq H_k^{P,\Sigma}$.*
2. *$L_0^{P,\Sigma} =$ P, $L_1^{P,\Sigma} =$ NP $\cap$ co-NP, and $L_1^{P,\Delta} =$ P.*
3. *$H_0^{P,\Sigma} = H_1^{P,\Delta} = \{A \mid A$ is $\leq_T^P$-complete for NP$\}$ and $H_1^{P,\Sigma} = \{A \mid A$ is $\leq_T^{SN}$-complete for NP$\}$.*

Definition 2.8 *For an arbitrary set A, language L is an element of the class of languages $\Theta(A) \Leftrightarrow (\exists$ a deterministic polynomial time oracle Turing machine $M)(\exists c > 0)[L = L(M, A)$ and, on each input x, M makes at most $c \cdot \log |x|$ queries$]$. For an arbitrary class $\mathcal{C}$ of languages, $\Theta(\mathcal{C}) = \bigcup_{A \in \mathcal{C}} \Theta(A)$.*

The following useful fact should not be overlooked: for every set A, $\mathrm{P}(\Theta(A)) = \mathrm{P}(A) = \Theta(\mathrm{P}(A))$.

The classes of the polynomial-time hierarchy (PH), including the Θ-classes, are $\{\Sigma_k^P, \Pi_k^P, \Delta_k^P, \Theta_k^P \mid k \geq 0\}$, where

$$\begin{aligned}
&\Sigma_0^P = \Pi_0^P = \Delta_0^P = \Theta_0^P = \mathrm{P}, \text{ and for } k \geq 0, \\
&\Sigma_{k+1}^P = \mathrm{NP}(\Sigma_k^P), \\
&\Pi_{k+1}^P = \text{co-}\Sigma_{k+1}^P, \\
&\Delta_{k+1}^P = \mathrm{P}(\Sigma_k^P), \\
&\Theta_{k+1}^P = \Theta(\Sigma_k^P).
\end{aligned}$$

For every set A, the classes of the polynomial-time hierarchy relative to A ($\mathrm{PH}(A)$) are $\{\Sigma_k^P(A), \Pi_k^P(A), \Delta_k^P(A), \Theta_k^P(A)) \mid k \geq 0\}$, where

$$\begin{aligned}
&\Sigma_0^P(A) = \Pi_0^P(A) = \Delta_0^P(A) = \Theta_0^P(A) = \mathrm{P}(A), \text{ and for } k \geq 0 \\
&\Sigma_{k+1}^P(A) = \mathrm{NP}(\Sigma_k^P(A)), \\
&\Pi_{k+1}^P(A) = \text{co-}\Sigma_{k+1}^P(A), \\
&\Delta_{k+1}^P(A) = \mathrm{P}(\Sigma_k^P(A)), \\
&\Theta_{k+1}^P(A) = \Theta(\Sigma_k^P(A)) \ .
\end{aligned}$$

Note that for every set A and for all $k \geq 0$,

$$\Sigma_k^P(A) \cup \Pi_k^P(A) \subseteq \Theta_{k+1}^P(A) \subseteq \Delta_{k+1}^P(A) \subseteq \Sigma_{k+1}^P(A) \cap \Pi_{k+1}^P(A).$$

Sets in the polynomial-time hierarchy relative to A have also been characterized in terms of polynomial bounded quantification over predicates in $\mathrm{P}(A)$ [Sto76, Wra76]:
$L \in \Sigma_k^P(A) \Leftrightarrow$ *there is a polynomial q and a deterministic polynomial time oracle machine M such that*

$$(\forall x)[x \in L \Leftrightarrow \exists_q y_1 \forall_q y_2 \ldots Q_q y_k [M^A(x \# y_1 \# y_2 \# \ldots \# y_k) \ accepts]],$$

S_1 is essentially the image of A under truth tables of the form "$y \in S$" and S_2 is essentially the image of $\overline{A}$ under truth tables of the form "$y \notin S$". It is easy to see that $A \leq^P_{1-tt} S_1 \oplus S_2$ by a machine $\hat{M}$ that, on input x, first generates the truth table that M generates on input x. If the truth-table is $TRUE$, then $\hat{M}$ accepts, and if it is $FALSE$, then $\hat{M}$ rejects. If the truth table is '$y \in S?$', then $\hat{M}$ accepts iff $0^{|x|}\#y\#^{p_M(|x|)-|y|} \in S_1$. If the truth table is '$y \notin S?$', then $\hat{M}$ accepts iff $0^{|x|}\#y\#^{p_M(|x|)-|y|} \notin S_2$.

The reductions $S_1 \leq^{NP}_m A$, $S_2 \leq^{NP}_m \overline{A}$, and $S_1 \oplus S_2 \leq^P_T A \oplus SAT$ are similar to the reductions $\hat{S} \leq^{NP}_m A$ and $\hat{S} \leq^P_T A \oplus SAT$ from the proof of Proposition 2.2. Also, S_1 and S_2 are sparse. □

The class APT, almost polynomial time, was defined by Meyer and Paterson [MP79], and the class P-close was defined by Schöning[Sch86a].

Definition 2.4 *Set A is in the class* APT *iff there is a deterministic Turing machine M that accepts A, a polynomial p, and a sparse set S such that M runs for at most $p(|x|)$ steps for all $x \in \overline{S}$.*

Definition 2.5 *Set A is an element of* P-close *if and only if there exists a set $B \in$ P such that $A \triangle B$ is a sparse set.*

The proofs of the following two propositions are straightforward.

Proposition 2.6 *Let $A \in APT$ such that $A \neq \emptyset$ and $A \neq \Sigma^*$. Then there is a sparse set S such that $A \equiv^P_m S$.*

Proposition 2.7 *Let $A \in$ P-close such that $A \neq \emptyset$ and $A \neq \Sigma^*$. Then there are sparse sets S_1 and S_2 such that $A \leq^P_{1-tt} S_1 \oplus S_2$, $S_1 \leq^P_m A$, and $S_2 \leq^P_m \overline{A}$.*

It should be noted that not every sparse set is in APT. In fact, it is easy to see that there are nonrecursive tally sets that are not in APT. Also, there are recursive sparse sets that are not $\leq^P_{1-tt}$-reducible to any set in APT.

We next define the Θ-operator and then incorporate it into the definition of the polynomial-time hierarchy.

For an arbitrary reducibility $\leq_r^P$, let $\mathrm{P}_r(\mathrm{SPARSE}) = \{A \mid \exists \text{ sparse set } S\ (A \leq_r^P S)\}$ and let $\mathrm{E}_r(\mathrm{SPARSE}) = \{A \mid \exists \text{ sparse set } \ (A \equiv_r^P S)\}$. $\mathrm{P}_r(\mathrm{TALLY})$ and $\mathrm{E}_r(\mathrm{TALLY})$ are defined similarly. These classes were first introduced in Book and Ko[BK88] and in Tang and Book[TB88].

The next two propositions are probably well known.

Proposition 2.2 *If $A \in \mathrm{P}_m(\mathrm{SPARSE})$, then there is a sparse set S such that $A \leq_m^P S$, $S \leq_m^{NP} A$, and $S \leq_T^P A \oplus \mathrm{SAT}$.*

Proof. Let $A \leq_m^P S$, where S is a sparse set, by a function f computable in polynomial time p_f. Define the set $\hat{S}$ such that

$$\hat{S} = \{0^n \# y \#^{p_f(n)-|y|} \mid \exists x \in A^{\leq n}[f(x) = y]\}.$$

$\hat{S}$ is essentially the image of A under f and it is sparse. It is easy to see that $A \leq_m^P \hat{S}$ by the function $g(x) = 0^{|x|} \# f(x) \#^{p_f(|x|)-|f(x)|}$. It is also easy to see that $\hat{S} \leq_m^{NP} A$ by a nondeterministic machine that, on input $0^n \# y \#^{p_f(n)-|y|}$, guesses an x such that $|x| \leq n$, verifies that $f(x) = y$, and then outputs x. If SAT is available as an oracle set, then this nondeterministic reduction can be modified to first see if there is an x such that $|x| \leq n$ and $f(x) = y$. If not, then reject. If so, use SAT to actually obtain such an x, and then accept iff $x \in A$. Thus, $\hat{S} \leq_T^P A \oplus \mathrm{SAT}$. $\square$

Proposition 2.3 *If $A \in \mathrm{P}_{1-tt}(\mathrm{SPARSE})$, then there are sparse sets S_1 and S_2 such that $A \leq_{1-tt}^P S_1 \oplus S_2$, $S_1 \leq_m^{NP} A$, $S_2 \leq_m^{NP} \overline{A}$, and $S_1 \oplus S_2 \leq_T^P A \oplus \mathrm{SAT}$.*

Proof. Let $A \leq_{1-tt}^P S$, where S is a sparse set, by a machine M that runs in polynomial time p_M. We can assume that M generates only four types of truth-tables: $TRUE$, $FALSE$, "$y \in S?$", and "$y \notin S?$". Define the set S_1 such that

$$S_1 = \{0^n \# y \#^{p_M(n)-|y|} \mid \exists x \in A^{\leq n}[M(x) \text{ generates the truth-table "} y \in S?\text{"}]\}.$$

Similarly, define the set S_2 such that

$$S_2 = \{0^n \# y \#^{p_M(n)-|y|} \mid \exists x \in \overline{A}^{\leq n}[M(x) \text{ generates the truth-table "} y \notin S?\text{"}]\}.$$

2 Notation, Definitions, and Preliminary Results

Σ denotes an arbitrary alphabet of size at least two. For $x \in \Sigma^*$, $|x|$ denotes the length of x, with λ denoting the string of length 0. For a set A over Σ, $\overline{A}$ denotes its complement $\Sigma^* - A$, and for a class of sets $\mathcal{C}$, co-$\mathcal{C}$ denotes the set of complements of the elements of $\mathcal{C}$. For arbitrary sets A and B, $A \oplus B = \{x0 \mid x \in A\} \cup \{x1 \mid x \in B\}$. For an arbitrary finite set $F = \{y_1, y_2, \ldots y_k\}$ over Σ, $code(F) = y_1 \# y_2 \ldots \# y_k$, where $\#$ is a symbol not in Σ and $y_1 < y_2 \ldots < y_k$ with respect to the lexicographic order on Σ^*. A tally set is any set over $\{1\}^*$.

$L(M)$ denotes the set accepted by Turing machine M, and $L(M, A)$ denotes the set accepted by an oracle Turing machine M using oracle set A. The classes P and NP have their standard definitions. $\mathrm{P}(A)$ and $\mathrm{NP}(A)$ are the A-relativized versions of P and NP.

Let $\| A \|$ denote the cardinality of a set A, and let $\| A^{\leq n} \|$ denote the cardinality of $A^{\leq n}$; that is, the cardinality of A restricted to strings of length at most n. A set S is *sparse* if and only if there is a polynomial p such that for all n, $\| S^{\leq n} \| \leq p(n)$. When working with sparse sets, we will be interested in their census predicate and census function as defined in the following:

Definition 2.1 *Let S be a sparse set over Σ.*

1. *The* census predicate *for S is $cp_S = \{0^n \# 0^j \mid j \leq \| S^{\leq n} \|\}$.*
2. *The* census function *for S is $cf_S : \{0\}^* \to \{0\}^*$ such that for all $n \in N$, $cf_S(0^n) = 0^{\| S^{\leq n} \|}$.*

Polynomial time many-one reducibility (denoted by $A \leq_m^P B$), *nondeterministic polynomial time many-one reducibility* (denoted by $A \leq_m^{NP} B$), *polynomial time one truth-table reducibility* (denoted by $A \leq_{1-tt}^P B$), and *polynomial time Turing reducibility* (denoted by $A \leq_T^P B$), have their usual definitions as do *NP-complete* and *polynomial time Turing-complete*. Set A is *strong nondeterministic polynomial time Turing reducible* to B (denoted by $A \leq_T^{SN} B$) if and only if $A \in \mathrm{NP}(B) \cap \text{co-NP}(B)$. A is $\leq_T^{SN}$*-complete* set for NP if $A \in \mathrm{NP}$ and for every set $B \in \mathrm{NP}$, $B \leq_T^{SN} A$.

For each oracle set A, we define the set $K(A)$ as follows: for a nondeterministic oracle machine M, a string x, and an integer k, let $< M, x, 0^k > \in K(A)$ if and only if M, using A as the oracle, accepts x in at most k steps. It is well known that, for every set A, $K(A)$ is $\leq_m^P$-complete for $\mathrm{NP}(A)$.

$\subseteq \Delta_{k-1}^P(A \oplus SAT)$. Allender and Hemachandra then located the following classes of sets in the Δ-levels of the extended low hierarchy:

$EL_2^{P,\Delta}$: Tally sets, standard and general left cuts, and sets that are $\leq_m^P$-equivalent or $\leq_T^P$-equivalent to some tally set.

$EL_3^{P,\Delta}$: Sparse and co-sparse sets, APT, P-close, sets that are $\leq_m^P$-reducible or $\leq_{1-tt}^P$-reducible to some sparse set, and sets that are $\leq_T^P$-equivalent to some sparse set.

In Section 5, we refine the extended low hierarchy by introducing levels corresponding to the Θ-levels of the polynomial-time hierarchy. We then apply Kadin's proof techniques to sharpen each of the known $EL_3^{P,\Delta}$ results to $EL_3^{P,\Theta}$ results, with the exception of APT. [3] For APT we use proof techniques related to those in Book, Long, and Selman[BLS84] to show that APT is actually in $EL_1^{P,\Sigma}$. Since it is the case that $EL_2^{P,\Delta} = EL_2^{P,\Theta} = EL_1^{P,\Sigma}$, improvement of $EL_2^{P,\Delta}$ results is not of interest.

Considering the optimality of the $EL_3^{P,\Theta}$ results, Allender and Hemachandra [AH92] have already shown that all of the classes listed earlier, except for APT, do not belong to $EL_2^{P,\Sigma}$. For example, they proved that there is a sparse set S such that S is not in $EL_2^{P,\Sigma}$; that is, $\Sigma_2^P(S) \not\subseteq \mathrm{NP}^{S \oplus SAT}$. Using an oracle construction technique that is also taken from Kadin[Kad87], we prove a slightly stronger lower bound on the $EL_3^{P,\Theta}$ results, again with the exception of APT. Letting f be a any function such that for all constants $c > 0$, $f(n) < c \cdot log\ n$ infinitely often, we show that there is a sparse set S such that $P^{\Sigma_2^P(S)[O(f(n)]} \not\subseteq \mathrm{P}^{\mathrm{NP(S \oplus SAT)}[O(f(n))]}$. Thus, any further improvement in the $EL_3^{P,\Theta}$ results would require an extremely fine grain refinement of the extended low hierarchy.

[3] [AKM92] and [Kö92] include several new examples of sets in $EL_2^{P,\Theta}$ and $EL_3^{P,\Theta}$, including the result that P/poly is in $EL_3^{P,\Theta}$.

$L_3^{P,\Theta}$: co-sparse NP sets, NP $\cap$ P-close, NP sets that are $\leq^P_{1-tt}$-reducible to some sparse set, and NP sets that are $\leq^P_{1-tt}$-equivalent or that are $\leq^P_T$-equivalent to some sparse set.

Comparing these lists to the earlier ones, the only classes of sets from $L_2^{P,\Delta}$ that we do not know to be in $L_2^{P,\Theta}$ are NP sets that are $\leq^P_T$-equivalent to some tally set and NP sets that are either standard or general left cuts for some real number x. Each of the classes of sets listed in $L_3^{P,\Delta}$ have been relocated in $L_3^{P,\Theta}$.

These improved lowness results follow from Kadin's proof techniques, and many follow directly from a single theorem generalizing ideas embedded in Kadin's proofs. Specifically, we show

Theorem 3.2 *For all $k \geq 1$, if S is a sparse set in Σ^P_k, then $\Theta^P_{k+1}(S) \subseteq \Theta^P_{k+1}$.*

Notice that Theorem 3.2 is merely a sharper version of the theorem by Book and Tang.

In Section 4, we consider the optimality of these new lowness results. Specifically, with respect to low classes of the form $L_k^{P,\Sigma}$, $L_k^{P,\Delta}$, and $L_k^{P,\Theta}$, we consider whether the new $L_2^{P,\Theta}$ and $L_3^{P,\Theta}$ results can be improved. It turns out that each of the new $L_2^{P,\Theta}$ results can be improved if and only if NEXT = co-NEXT. Also, we do not know if the $L_3^{P,\Theta}$ results can be improved to $L_2^{P,\Sigma}$, or better. However, we do construct an oracle set relative to which none of the $L_3^{P,\Theta}$ results can be improved to $L_2^{P,\Delta}$, and we do not know if there is an oracle set relative to which the $L_3^{P,\Theta}$ classes are not in $L_2^{P,\Sigma}$. This oracle construction indicates that Kadin's proof techniques will not suffice for any significant improvement in the $L_3^{P,\Theta}$ results.

In order to apply the concept of lowness to sets outside of NP, Balcázar, Book, and Schöning [BBS86] introduced the extended low hierarchy. Level k of the extended low hierarchy, denoted $EL_k^{P,\Sigma}$, consists of sets A such that $\Sigma^P_k(A) \subseteq \Sigma^P_{k-1}(A \oplus SAT)$. Several classes of sets have been located in the extended low hierarchy; for example, sets in P/poly were shown to be in $EL_3^{P,\Sigma}$ in [BBS86].

Just as Schöning had done for the low hierarchy, Allender and Hemachandra [AH92] introduced a refinement of the extended low hierarchy based on the Δ-levels of the polynomial-time hierarchy: set A is in $EL_k^{P,\Delta}$ if and only if $\Delta^P_k(A)$

For each of these classes, sparse sets play a key role in their being low classes, and the underlying proof technique used to prove their lowness involves deterministic enumeration of sparse sets. Mahaney [Mah82] introduced this enumeration technique when showing that the existence of a sparse polynomial time Turing complete set for NP implies the polynomial-time hierarchy collapses to Δ_2^P. The technique can be applied to arbitrary levels of the polynomial-time hierarchy to obtain the following theorem by Book and Tang [BT89]:

Theorem 3.1 [BT89] *For all $k \geq 1$, if S is a sparse set in Σ_k^P, then $\Delta_{k+1}^P(S) \subseteq \Delta_{k+1}^P$.*

Each of the lowness results just enumerated follow, either directly or indirectly, from this general theorem and its corresponding proof technique.

More recently, Kadin [Kad87] used a new idea to strengthen Mahaney's result, showing that if NP has a sparse strong nondeterministic polynomial time Turing complete set, then the polynomial-time hierarchy collapses to Θ_2^P, the class of languages recognizable in deterministic polynomial time with $O(log\ n)$ queries to NP oracle sets. The key idea in Kadin's proof is to compute the census of sparse sets in advance, an operation that requires only $O(log\ n)$ queries to an appropriate oracle set. A second idea also used by Kadin, but not as crucial to his proof, is the idea of the pseudo-complement of a sparse set, an idea also introduced by Mahaney.

In this paper, we use Kadin's proof techniques to sharpen almost all of the lowness results listed earlier. We begin by introducing, in Section 2, yet another refinement of the high and low hierarchies, this refinement being based on the Θ-classes of the polynomial-time hierarchy. Letting $L_k^{P,\Theta}$ denote level k of the low hierarchy relative to Θ-classes, in Section 3 we locate the following classes in $L_2^{P,\Theta}$ and $L_3^{P,\Theta}$: [2]

$L_2^{P,\Theta}$: Sparse and tally NP sets, NP $\cap$ (co-NP/*log*), NP $\cap$ APT, NP sets that are $\leq_m^P$-reducible to some sparse or some tally set.

[2] In recent work, Arvind, Köbler, and Mundhenk [AKM92] and Köbler[Kö92] have located new classes of sets in $L_2^{P,\Theta}$ and $L_3^{P,\Theta}$. This includes, for example, that NP $\cap$ P/poly and NP $\cap$ (NP $\cap$ co-NP)/poly are in $L_3^{P,\Theta}$. In addition, earlier results of Lozano and Torán [LT91] show that certain classes of sets are in $L_2^{P,\Theta}$, although the formal definition of $L_2^{P,\Theta}$ was not yet in the literature.

1 Introduction

Building on ideas from recursive function theory, Schöning [Sch83] introduced the low and high hierarchies for NP. These hierarchies are derived as follows: Suppose that set $A \in$ NP, and let $k \geq 1$. Then, $\Sigma_k^P \subseteq \Sigma_k^P(A) \subseteq \Sigma_{k+1}^P$. If $\Sigma_k^P = \Sigma_k^P(A)$, then A belongs to $L_k^{P,\Sigma}$, level k of the low hierarchy. On the other hand, if $\Sigma_k^P(A) = \Sigma_{k+1}^P$, then A belongs to $H_k^{P,\Sigma}$, level k of the high hierarchy.[1] Intuitively, when A belongs to $L_k^{P,\Sigma}$, then the information content of A, with respect to the $\Sigma_k^P(\ )$ operator, is so low that $\Sigma_k^P(A)$ has no more power than Σ_k^P, and when A belongs to $H_k^{P,\Sigma}$, then the information content of A, with respect to the $\Sigma_k^P(\ )$ operator, is so high that $\Sigma_k^P(A)$ includes all of Σ_{k+1}^P.

The low and high hierarchies provide a formal framework for analyzing the internal structure of NP. For example, Schöning proved that NP sets with small circuits are in $L_3^{P,\Sigma}$ and that self-reducible NP sets with small circuits are in $L_2^{P,\Sigma}$.

Schöning [Sch86b] also introduced a refinement of the low and high hierarchies based on the Δ-classes of the polynomial-time hierarchy: if $\Delta_k^P = \Delta_k^P(A)$, then A belongs to $L_k^{P,\Delta}$, level k of the low hierarchy relative to Δ-classes, and if $\Delta_k^P(A) = \Delta_{k+1}^P$, then A belongs to $H_k^{P,\Delta}$, level k of the high hierarchy relative to Δ-classes. Schöning[Sch86b], Ko and Schöning[KS85], Balcázar and Book[BB86], Balcázar and Schöning[BS], and Allender and Hemachandra[AH92] have located several classes of sets in this refined low hierarchy:

$L_2^{P,\Delta}$: Sparse and tally NP sets, NP $\cap$ (co-NP/*log*), NP $\cap$ APT, NP sets that are standard or general left cuts for some real number x, NP sets that are $\leq_m^P$-reducible to some sparse or some tally set, and NP sets that are $\leq_T^P$-equivalent to some tally set.

$L_3^{P,\Delta}$: co-sparse NP sets, NP $\cap$ P-close, NP sets that are $\leq_{1-tt}^P$-reducible to some sparse set, and NP sets that are $\leq_{1-tt}^P$-equivalent or that are $\leq_T^P$-equivalent to some sparse set.

[1] Schöning originally used the notation L_k^P and H_k^P for the low and high hierarchies based on the Σ-classes of the polynomial-time hierarchy. We use the more suggestive notation $L_k^{P,\Sigma}$ and $H_k^{P,\Sigma}$ to distinguish between low and high hierarchies based on different classes in the polynomial-time hierarchy. Similarly, we will be using the notation $L_k^{P,\Delta}$, $H_k^{P,\Delta}$, $EL_k^{P,\Sigma}$, and $EL_k^{P,\Delta}$, for the notation $\hat{L}_k^P$, $\hat{H}_k^P$, EL_k, and $\widehat{EL}_k$, respectively.

Abstract

Based on the Σ and Δ-classes of the polynomial-time hierarchy, Schöning [Sch83, Sch86b] introduced low and high hierarchies within NP. Several classes of sets have been located in the bottom few levels of these hierarchies [Sch83, Sch86b, KS85, BB86, BS, AH92]. Most results placing sets in the Δ-levels of the low hierarchy are related to sparse sets, and the proof techniques employed involve deterministic enumeration of sparse sets. Balcázar, Book, and Schöning [BBS86] and Allender and Hemachandra[AH92] introduced extended low hierarchies, involving sets outside of NP, based on the Σ and Δ-classes of the polynomial-time hierarchy. Several classes of sets have been located in the Δ-levels of these hierarchies as well, and once again most such results involve sparse sets.

In this paper we introduce a refinement of the low and high hierarchies and of the extended low hierarchies. Our refinement is based on the Θ-classes of the polynomial-time hierarchy. We show that almost all of the classes of sets that are known to belong to the Δ-levels of the low and extended low hierarchies actually belong to the newly defined Θ-levels of these hierarchies. Our proofs use Kadin's [Kad87] technique of computing the census of a sparse set first, followed by a nondeterministic enumeration of the set. This leads to the sharper lowness results.

We also consider the optimality of these new lowness results. For sets in the Θ-levels of the low hierarchy we have oracle results indicating that substantially stronger results are not possible through use of Kadin's technique. For sets in the Θ-classes of the extended low hierarchy we have tight absolute lower bounds; that is, lower bounds without oracles. These bounds are slightly stronger than similar bounds appearing in [AH92].

A Refinement of the Low and High Hierarchies [1]

Timothy J. Long
Ming-Jye Sheu

Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

[1]This work was supported in part by NSF Grant CCR-8909071