COMPLEXITY-RESTRICTED ADVICE FUNCTIONS

JOHANNES KÖBLER* AND THOMAS THIERAUF†

Abstract. We consider uniform subclasses of the nonuniform complexity classes defined by Karp and Lipton [23] via the notion of advice functions. These subclasses are obtained by restricting the complexity of computing correct advice. We also investigate the effect of allowing advice functions of limited complexity to depend on the input rather than on the input's length. Among other results, using the notions described above, we give new characterizations of (a) $\mathrm{NP}^{\mathrm{NP} \cap \mathrm{SPARSE}}$, (b) NP with a restricted access to an NP oracle and (c) the odd levels of the boolean hierarchy.

As a consequence, we show that every set that is nondeterministically truth-table reducible to SAT in the sense of Rich [35] is already deterministically truth-table reducible to SAT. Furthermore, it turns out that the NP reduction classes of bounded versions of this reducibility coincide with the odd levels of the boolean hierarchy.

Key words. nonuniform complexity classes, advice classes, optimization functions, restricted oracle access, sparse NP sets, relativization, boolean hierarchy, truth-table reducibility

AMS(MOS) subject classifications. 68Q15

1. Introduction. In their fundamental paper, Karp and Lipton [23] introduced the notion of advice functions and investigated nonuniform complexity classes which they denoted by $\mathcal{C}/\mathcal{F}$, where $\mathcal{C}$ is a class of sets and $\mathcal{F}$ is a class of (advice) functions. A typical class is P/poly, where poly is the set of polynomially length bounded functions. The interest in P/poly stems from the fact that it consists exactly of the languages that can be computed by polynomially size-bounded circuits [34].

Intuitively, a set A is in $\mathcal{C}/\mathcal{F}$, if A can be solved by a machine of type $\mathcal{C}$ that gets, in addition to the input x, the advice $f(x)$, where f is a function in $\mathcal{F}$ depending only on the length of x. Many researchers have considered nonuniform classes where the function class $\mathcal{F}$ is defined by a quantitative length restriction such as poly and log (see, for example, [3, 5, 23, 36]). Note that for such $\mathcal{F}$ there are nonrecursive functions in $\mathcal{F}$, and therefore $\mathcal{C}/\mathcal{F}$ contains nonrecursive languages.

Here, we consider uniform language classes obtained by imposing complexity bounds on the advice functions. Note that Kämper [22] investigates refinements of the original $\mathcal{C}/\mathcal{F}$ definition by delimiting the complexity of proof sets, i.e., special

* Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany. Supported in part by the DAAD through Acciones Integradas 1991, 313-AI-e-es/zk.

† Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany. Work done in part while visiting the University of Rochester. Supported in part by DFG Postdoctoral Stipend Th 472/1-1, NSF Grant CCR-8957604, and by the DAAD through Acciones Integradas 1991, 313-AI-e-es/zk.

sets of correct advice. In contrast to this, we directly bound the complexity of computing correct advice. With this concept, we are able to show characterizations as well as finer distinctions of several complexity classes. For example, we show that the class $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ coincides with the class $\mathrm{NP}/\mathrm{OptP}[O(\log n)]$, a subclass of $\mathrm{NP}/\log$, where correct advice is computable by an OptP function [29], i.e.,

$$\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}} = \mathrm{NP}/\mathrm{OptP}[O(\log n)]. \tag{1.1}$$

One can interpret equality (1.1) as stating that (exactly) the languages in $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ can be computed in the following way: on input x of length n, at first an $\mathrm{OptP}[O(\log n)]$ precomputation takes place that gets as input only 1^n. The (logarithmically length-bounded) output of this precomputation is then passed along with x to the subsequent NP computation, that decides the membership of x.

Motivated by the relativized separation of P and NP of Baker, Gill, and Solovay [2] (exploiting the fact that an NP oracle machine can ask superpolynomially many queries), Book, Long, and Selman [11] introduced restricted relativizations of NP by bounding the number of oracle queries in various ways. Subsequently, Long [32] investigated the relationship between restricted access of nondeterministic machines to an oracle and full access to a sparse oracle set. Let NP^A_R be the class of all languages whose membership in NP^A is witnessed by an oracle machine such that the number of potential oracle queries in A (asked on some oracle) is polynomially bounded. From this definition, it is clear that $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ is contained in $\mathrm{NP}^{\mathrm{NP}}_R$. Since also coNP is contained in $\mathrm{NP}^{\mathrm{NP}}_R$, $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ and $\mathrm{NP}^{\mathrm{NP}}_R$ are different unless the polynomial hierarchy collapses [21]. By considering the proof of equality (1.1), we will see that if we let the $\mathrm{OptP}[O(\log n)]$ advice function depend not only on the length of the input but on the input itself, we get a characterization of $\mathrm{NP}^{\mathrm{NP}}_R$. This leads us to define the class $\mathcal{C}//\mathcal{F}$, that is defined in the same way as $\mathcal{C}/\mathcal{F}$, but with the advice functions depending on the input. Thus, we obtain the following characterization of $\mathrm{NP}^{\mathrm{NP}}_R$,

$$\mathrm{NP}^{\mathrm{NP}}_R = \mathrm{NP}//\mathrm{OptP}[O(\log n)] \tag{1.2}$$

The characterizations (1.1) and (1.2) give insight into the difference between restricted access to NP oracles and full access to sparse NP sets.

It seems that the notion of $\mathcal{C}//\mathcal{F}$ is an appropriate concept for studying different kinds of truth-table reducibilities. Let χ^{SAT}_k be the k-ary characteristic function of SAT. Then, $\mathrm{P}//\chi^{\mathrm{SAT}}_k\circ\mathrm{FP}$ is the class of sets that are k-truth-table reducible to some NP set. It is known that these classes are interleaved with levels of the boolean hierarchy: $\mathrm{NP}(k) \subseteq \mathrm{P}//\chi^{\mathrm{SAT}}_k\circ\mathrm{FP} \subseteq \mathrm{NP}(k+1)$ for all $k \geq 1$ [28]. Since $\mathrm{P}//\chi^{\mathrm{SAT}}_k\circ\mathrm{FP}$ is closed under complementation, these classes are all different unless the boolean hierarchy collapses.

$\mathrm{NP}//\chi^{\mathrm{SAT}}_k\circ\mathrm{FP}$ is the class of sets that are k-truth-table reducible to some NP set, where the evaluator is an NP machine. These classes turn out to coincide with the odd levels of the boolean hierarchy, giving for the first time a charcterization of the levels of the boolean hierarchy in terms of reduction classes,

$$\mathrm{NP}(2k+1) = \mathrm{NP}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}.$$

Furthermore, we show that $\mathrm{NP}(2k+1) = \mathrm{NP}^{\mathrm{NP}}_{k\text{-}tt}$, where $\mathrm{NP}^{\mathrm{NP}}_{k\text{-}tt}$ is the class of sets that are nondeterministically k-truth-table reducible to a set in NP in the sense of [35][1], whereas in the unbounded case all sets nondeterministically truth-table reducible to SAT are already deterministically truth-table reducible to SAT, i.e., $\mathrm{NP}^{\mathrm{NP}}_{tt} = \mathrm{P}^{\mathrm{NP}}_{tt}$. The latter result also holds for the strong nondeterministic truth-table reducibility $\leq^{\mathrm{SN}}_{tt}$ introduced by Long [31], i.e., we show that $\{A \mid A \leq^{\mathrm{SN}}_{tt} \mathrm{SAT}\} = \{A \mid A \leq^{\mathrm{P}}_{tt} \mathrm{SAT}\}$.

The paper is organized as follows. Section 2 introduces notation and gives basic definitions. In Section 3, we prove the above mentioned characterizations of $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ and $\mathrm{NP}^{\mathrm{NP}}_{R}$ and we show that changing from $\mathrm{OptP}[O(\log n)]$ to the larger function class FewOptP (containing all functions whose membership in OptP is witnessed by an NP transducer that generates only polynomially many different outputs) does not increase the power of $\mathrm{NP}/\mathrm{OptP}[O(\log n)]$ and $\mathrm{NP}//\mathrm{OptP}[O(\log n)]$.

In Section 4, we separate some of these complexity classes in relativized world; the main result is a separation of $\mathrm{P}/\mathrm{OptP}[O(\log n)]$ and $\mathrm{P}^{\mathrm{NP}\cap\mathrm{SPARSE}[O(\log n)]}$.

In Section 5, we give several characterizations of certain levels of the boolean hierarchy in terms of various complexity restricted advice function classes.

2. Preliminaries and Notation. All languages considered here are over the alphabet $\Sigma = \{0,1\}$. For a string $x \in \Sigma^*$, $|x|$ denotes its length. We assume the existence of a pairing function $\langle\cdot,\cdot\rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$ that is computable in polynomial time and has inverses also computable in polynomial time. $\langle\cdot,\cdot\rangle$ can be extended to encode finite sequences $(x_1,\ldots,x_k)$ of strings into a string $\langle x_1,\ldots,x_k\rangle \in \Sigma^*$. For a set A, $|A|$ denotes its cardinality. The complement $\Sigma^* - A$ of A is denoted by $\overline{A}$. $A^{\leq n}$ is the set of all strings in A of length less than or equal to n.

A languages S is *sparse*, if there is a polynomial p such that for all n, the number of words in S up to length n is at most $p(n)$. Let SPARSE be the class of all sparse languages. A set T is *tally*, if T is a subset of 1^*. Let TALLY be the class of all tally sets.

We assume that the reader is familiar with (nondeterministic, polynomial-time bounded, oracle) Turing machines and complexity classes (see [4, 36]). FP is the class of functions computable by a deterministic polynomial-time bounded Turing transducer. An NP *transducer* is a nondeterministic polynomial-time bounded Turing machine T that on every branch either accepts and writes a binary number

[1] By requiring the NP generator to be single valued, Rich [35] has modified the nondeterministic truth-table reducibility originally defined by Ladner, Lynch, and Selman [30].

on its output tape or rejects. The set of outputs generated by T on input x is denoted by $out_T(x)$.

Krentel [29] defines an NP *metric Turing machine* to be an NP transducer that accepts on every branch. For an NP metric Turing machine T and an input $x \in \Sigma^*$ let $\max_T(x)$ $[\min_T(x)]$ be the maximum [minimum] output generated by T on input x on any accepting computation of T. The class OptP [29] of optimization functions is defined as

$$\mathrm{OptP} = \{ \max_T, \min_T \mid T \text{ is an NP metric Turing machine } \}.$$

For a class $\mathcal{R}$ of functions on the natural numbers (called restricting functions), we define the subclass

$$\mathrm{OptP}[\mathcal{R}] = \{f \in \mathrm{OptP} \mid \exists\, r \in \mathcal{R}\, \forall\, x \in \Sigma^* : |f(x)| \le r(|x|)\}$$

containing all optimization functions $f \in \mathrm{OptP}$ such that the length of $f(x)$ in binary is bounded in $|x|$ by a function from $\mathcal{R}$.

$\mathrm{P}^{\mathrm{NP}[\mathcal{R}]}$ denotes the class of sets whose membership in P^{NP} can be witnessed by an oracle machine M making for some $r \in \mathcal{R}$ at most $r(n)$ many queries on inputs of length n. In the case that $\mathcal{R}$ is a singleton set $\{r\}$ we simply write $\mathrm{OptP}[r]$ and $\mathrm{P}^{\mathrm{NP}[r]}$, respectively. Throughout the paper we assume that for every restricting function r the function $x \mapsto r(|x|)$ is computable in polynomial time.

Karp and Lipton [23] introduced the notion of advice functions in order to define nonuniform complexity classes. For a class $\mathcal{C}$ of sets and a class $\mathcal{F}$ of functions from Σ^* 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 $x \in \Sigma^*$

$$x \in A \quad \Leftrightarrow \quad \langle x, h(1^{|x|})\rangle \in B.$$

Note that the advice function h depends only on the length of x. By canceling this restriction we obtain the class $\mathcal{C}//\mathcal{F}$ of all sets A such that there is a set $B \in \mathcal{C}$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$

$$x \in A \quad \Leftrightarrow \quad \langle x, h(x)\rangle \in B.$$

By definition, $\mathcal{C}/\mathcal{F}$ is a subset of $\mathcal{C}//\mathcal{F}$ for each class of sets $\mathcal{C}$ and each class of functions $\mathcal{F}$ which fulfills the condition that if $h \in \mathcal{F}$, then also $x \mapsto h(1^{|x|}) \in \mathcal{F}$. Special advice function classes considered in the literature are $\mathrm{poly} = \{h : \Sigma^* \to \Sigma^* \mid$ there exists a polynomial p such that for all x, $|h(x)| \le p(|x|)\}$ and $\log = \{h : \Sigma^* \to \Sigma^* \mid |h(x)| = O(\log(|x|))\}$.

3. $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ **versus** $\mathrm{NP}_R^{\mathrm{NP}}$**.** In this section we show that $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ can be characterized as the class $\mathrm{NP}/\mathrm{OptP}[O(\log n)]$, i.e., the class of sets that are accepted by an NP machine with advice of a logarithmically length bounded OptP function. Further, it turns out that the related class $\mathrm{NP}_R^{\mathrm{NP}}$ (see definition below) coincides with $\mathrm{NP}//\mathrm{OptP}[O(\log n)]$. For the latter two classes we can show that they are also equal to $\mathrm{P}//\mathrm{OptP}[O(\log n)]$ which, by a result of Krentel [29], is identical to $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$.

Definition 3.1. [11] *For any oracle Turing machine M and any string $x \in \Sigma^*$ let $Q(M, A, x)$ be the set of all oracle queries that M may ask on input x using oracle A, i.e., the set of all strings $y \in \Sigma^*$ such that in some computation of M on input x under oracle A the oracle is queried about y. $Q(M, x)$ is the set of all oracle queries of M on input x using any oracle, i.e., $Q(M, x) = \bigcup_{A \subseteq \Sigma^*} Q(M, A, x)$.*

For any set A, $\mathrm{NP}_R(A)$ is the class of sets $L \in \mathrm{NP}(A)$ whose membership is witnessed by a machine M such that the number of potential oracle queries in A is polynomially bounded, i.e., there exists a polynomial p such that $|Q(M, x) \cap A| \leq p(|x|)$ for all x.

Our first theorem states that if a language L is accepted by an NP oracle machine M using an NP oracle A in such a way that the number of potential oracle queries that are in A is polynomially bounded, then L is in $\mathrm{NP}//\mathrm{OptP}[O(\log n)]$, i.e., membership to L can be tested by an NP machine which gets along with the input the precomputed value of an $\mathrm{OptP}[O(\log n)]$ function. In the special case that A is sparse this containment can be strengthened to $\mathrm{NP}/\mathrm{OptP}[O(\log n)]$, i.e., for all inputs of the same length the advice function yields the same result. The proof is by a census argument similar to that used by Hemachandra [18] and Kadin [21].

Theorem 3.2.

i) $\mathrm{NP}_R^{\mathrm{NP}} \subseteq \mathrm{NP}//\mathrm{OptP}[O(\log n)]$,

ii) $\mathrm{NP}^{\mathrm{NP} \cap \mathrm{SPARSE}} \subseteq \mathrm{NP}/\mathrm{OptP}[O(\log n)]$.

Proof. Let $L = L(M, A)$ for an NP machine M and an oracle A in NP, and let p be a polynomial that bounds the running time of M.

To show *i)* let r be a polynomial such that $|Q(M, x) \cap A| \leq r(|x|)$ for all x. An NP machine knowing the size of the set $Q(M, x) \cap A$ can guess this set (note that the problem to decide for given strings x and y whether y is in $Q(M, x) \cap A$ is in NP). Define the function

$$h(x) = |Q(M, x) \cap A|$$

and the set

$$B = \{\langle x, k\rangle \mid \exists X \subseteq Q(M, x) \cap A : |X| = k \text{ and } x \in L(M, X)\}.$$

Then $B \in \mathrm{NP}$ and $h \in \mathrm{OptP}[O(\log |x|)$, since $h(x)$ is the maximum output of the following algorithm.

> On input x guess $k \leq r(|x|)$ and $x_1 < \ldots < x_k \in \Sigma^{\leq p(|x|)}$;
> if $x_1, \ldots, x_k \in Q(M, x) \cap A$, then output k, else output 0.

Now, it holds for all $x \in \Sigma^*$ that $x \in L$ if and only if $\langle x, h(x)\rangle \in B$. Therefore, L is in NP//OptP[$O(\log n)$].

For *ii*) let A be sparse and r be a polynomial such that $|A^{\leq p(n)}| \leq r(n)$, for all n. Define the function

$$h(x) = |A^{\leq p(|x|)}|$$

and the set

$$B = \{\langle x, k\rangle \mid \exists X \subseteq A^{\leq p(|x|)} : |X| = k \text{ and } x \in L(M, X)\}.$$

By a similar argument as in the proof of *i*), $x \in L$ if and only if $\langle x, h(1^{|x|})\rangle \in B$. This shows that L is in NP/OptP[$O(\log n)$]. □

Combining Theorem 3.2 *ii*) with the result of Balcázar and Schöning [5] that $\mathrm{NP/log} \cap \mathrm{coNP} \subseteq \mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}}$ (see also [3]), it follows that for every coNP set in NP/log correct advice can already be computed by an OptP function.

Corollary 3.3. $\mathrm{NP/log} \cap \mathrm{coNP} = \mathrm{NP/OptP}[O(\log n)] \cap \mathrm{coNP}$.

To show the reverse containments of Theorem 3.2, we make use of the following lemma. It states that every OptP function h can be computed by a deterministic polynomial-time oracle machine by asking $|h(x)|$ many queries to an NP oracle.

Lemma 3.4. [29] $\mathrm{OptP}[r] \subseteq \mathrm{FP}^{\mathrm{NP}[r]}$ *for any restricting function* r.

Corollary 3.5.

i) $\mathrm{NP}_R^{\mathrm{NP}} = \mathrm{NP//OptP}[O(\log n)]$,

ii) $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}} = \mathrm{NP}^{\mathrm{NP}\cap\mathrm{TALLY}} = \mathrm{NP/OptP}[O(\log n)]$.

Proof. By Theorem 3.2, it only remains to show the inclusions from right to left.

To show *i*), let L be in NP//OptP[$O(\log n)$] via an NP machine N and an optimization function h. Then L can be accepted by an NP machine M that computes deterministically by binary search the value of h according to Lemma 3.4 asking $O(\log n)$ many queries to an NP oracle, and then simulates N without asking further oracle queries. Since $Q(M, x)$ is polynomially bounded, it follows that L is in $\mathrm{NP}_R^{\mathrm{NP}}$.

If h is a function that depends only on the length of its argument, then $h(x)$ can be computed by binary search using the tally NP set $T = \{1^{\langle n,k\rangle} \mid k \leq h(1^n)\}$. This proves *ii*). □

Note that the above proof shows that every language in NP//OptP[$O(\log n)$] (and thus in $\mathrm{NP}_R^{\mathrm{NP}}$) can in fact be accepted by an NP oracle machine M such that $Q(M, x)$ is polynomially bounded.

In the next lemma, we show that an NP computation getting along with the input the result $h(x)$ of an OptP precomputation can be transformed into a P computation by precomputing one additional bit. Note that this bit actually depends on x even if $h(x)$ only depends on the length of x.

Lemma 3.6. $\mathrm{NP//OptP}[r] \subseteq \mathrm{P//OptP}[r + 1]$, *for any function* r.

Proof. Let L be in $\mathrm{NP}//\mathrm{OptP}[r]$, witnessed by an NP set B and an $\mathrm{OptP}[r]$ function $h = \max_T$ for some NP metric machine T. Define the $\mathrm{OptP}[r+1]$ function

$$h'(x) = \begin{cases} h(x)1, & \text{if } \langle x, h(x)\rangle \in B, \\ h(x)0, & \text{otherwise.} \end{cases}$$

Then it holds for all x that $\langle x, h(x)\rangle \in B \quad \Leftrightarrow \quad \langle x, h'(x)\rangle \in B'$, where the set $B' = \{\langle x, k\rangle \mid k \text{ is odd}\}$ is in P. The case that $h = \min_T$ can be proved analogously. □

Combining Corollary 3.5 *i)* and Lemma 3.6 we obtain a further characterization of the class $\mathrm{NP}^{\mathrm{NP}}_R$ and its closure under complementation. Note that $\mathrm{P}//\mathrm{OptP}[O(\log n)] = \mathrm{P}^{\mathrm{NP}[O(\log n)]}$ [29].

Corollary 3.7. $\mathrm{NP}^{\mathrm{NP}}_R = \mathrm{P}//\mathrm{OptP}[O(\log n)]$.

Corollary 3.8. $\mathrm{NP}^{\mathrm{NP}}_R$ *is closed under complementation.*

Remark 3.9. *The results stated in Corollary 3.5 can be extended to the classes of the polynomial-time hierarchy [37]. In order to do so, we define restricted relativizations of the Σ-levels of the polynomial hierarchy. $\Sigma^{\mathcal{C}}_{k,R}$ is the class of all sets L accepted by a k-alternating polynomial-time Turing machine [16] using an oracle A from $\mathcal{C}$ such that $|Q(M,x) \cap A|$ is polynomially bounded. Then, the results stated in Corollary 3.5 can be extended to*

$$\begin{aligned} \Sigma_k^{\Sigma_k \cap \mathrm{SPARSE}} &= \Sigma_k/\mathrm{Opt}\,\Sigma_{k-1}[O(\log n)], \\ \Sigma_{k,R}^{\Sigma_k} &= \Sigma_k//\mathrm{Opt}\,\Sigma_{k-1}[O(\log n)] \;=\; \mathrm{P}^{\Sigma_k[O(\log n)]}, \end{aligned}$$

where $\mathrm{Opt}\,\mathcal{C}$ *is the class of optimization functions computable by an* NP *transducer using some oracle in the class $\mathcal{C}$. Since $\Sigma_k/\mathrm{Opt}\,\Sigma_{k-1}[O(\log n)]$ is included in $\mathrm{P}^{\Sigma_k[O(\log n)]}$, this sharpens the recent result in [13] that $\Sigma_k^{\Sigma_k \cap \mathrm{SPARSE}} \subseteq \mathrm{P}^{\Sigma_k[O(\log n)]}$.*

Remark 3.10. *The advice (even depending on the input) provided by an* $\mathrm{OptP}[O(\log n)]$ *function does not increase the power of the probabilistic class* PP*:* $\mathrm{PP}//\mathrm{OptP}[O(\log n)] = \mathrm{PP}$*. This follows from the result by Toda [40] that* $\mathrm{PP}^{\mathrm{NP}}_R = \mathrm{PP}$*, since* $\mathrm{PP}//\mathrm{OptP}[O(\log n)]$ *coincides with the class* $\mathrm{PP}//\mathrm{FP}^{\mathrm{NP}[O(\log n)]}$ *(see Lemma 3.4) that is clearly contained in* $\mathrm{PP}^{\mathrm{NP}}_R$.

Next, we consider uniform subclasses of P/log and P/poly. Whereas the proof of Corollary 3.5 *ii)* also yields the inclusion of $\mathrm{P}/\mathrm{OptP}[O(\log n)]$ in $\mathrm{P}^{\mathrm{NP}\cap\mathrm{SPARSE}[O(\log n)]}$, the census technique of Theorem 3.2 cannot be applied to obtain the reverse containment. The next theorem is proved by constructing (long enough initial segments of) a sparse NP set by an OptP computation. The underlying technique was used by Mahaney [33] to show that $\mathrm{NP}^{\mathrm{NP}\cap\mathrm{SPARSE}} \subseteq \mathrm{P}^{\mathrm{NP}}$.

Theorem 3.11. $\mathrm{P}^{\mathrm{NP}\cap\mathrm{SPARSE}} \subseteq \mathrm{P}/\mathrm{OptP}$.

Proof. Let $L = L(M,S)$, for a P machine M and a sparse NP set S. Let p and r be polynomials such that p bounds the running time of M and $|S^{\leq n}| \leq r(n)$.

Define

$$h(x) = \langle S^{\leq p(|x|)} \rangle.$$

Then, $h \in$ OptP, since $h(x)$ is the maximum output of the following algorithm.

> On input 1^n guess $k \leq r(p(n))$ and $x_1 < \ldots < x_k \in \Sigma^{\leq p(|x|)}$;
> if $x_1, \ldots, x_k \in S$, then output $\langle x_1, \ldots, x_k \rangle$, else output 0.

Now, the computation of M using oracle S on input x can be simulated by a P machine answering oracle questions according to the set $h(1^{|x|})$. □

Let FewOptP be the class of functions $f \in$ OptP computed by an NP transducer that produces at most a polynomial number of different outputs. Clearly, OptP$[O(\log n)] \subseteq$ FewOptP, and obviously, this is a proper inclusion.

However, as shown by the next theorem, the classes NP/OptP$[O(\log n)]$ and NP//OptP$[O(\log n)]$ remain unchanged when the function class OptP$[O(\log n)]$ is replaced by the larger class FewOptP.

THEOREM 3.12.

i) NP//FewOptP = P//FewOptP = P//OptP$[O(\log n)]$,

ii) NP/FewOptP = NP/OptP$[O(\log n)]$.

Proof. Let L be a set in NP//FewOptP via $A \in$ NP and $f \in$ FewOptP. Let T be an NP metric machine for f, i.e., $f = \max_T$ (the proof for $f = \min_T$ is similar), and the number of different outputs of T is polynomially bounded. Define the function

$$h(x) = |out_T(x)|$$

and the set

$$B = \{\langle x, m\rangle \mid \exists\, z_1 < \ldots < z_m \in out_M(x) : \ \langle x, z_m\rangle \in A\}.$$

It is easy to see that $h \in$ OptP$[O(\log n)]$ and $B \in$ NP. Now, x is in L if and only if $\langle x, h(x)\rangle$ is in B, and therefore, L is in NP//OptP$[O(\log n)]$ = P//OptP$[O(\log n)]$. The latter equality follows from Corollaries 3.5, part *i*), and 3.7. The proof of *ii*) is analogous, we only have to replace $out_T(x)$ by $out_T(1^{|x|})$. □

The technique used in the previous proof cannot be applied to show that the classes P/OptP$[O(\log n)]$ and P/FewOptP are equal. However, the proof of P/OptP$[O(\log n)] \subseteq \mathrm{P}^{\mathrm{NP}\cap\mathrm{SPARSE}[O(\log n)]}$ (using binary search, see the proof of Corollary 3.5 *ii*)) can be refined to show the following theorem. It states that a set in P/FewOptP can be decided in polynomial time by querying a sparse NP oracle (polynomially often).

THEOREM 3.13. P/FewOptP $\subseteq \mathrm{P}^{\mathrm{NP}\cap\mathrm{SPARSE}}$.

Proof. Let f be in FewOptP and let T be an NP transducer computing f. Using the sparse NP set

$$S = \{\langle 1^n, m, i, z\rangle \mid \exists\, z_1 < \ldots < z_m \in out_T(1^n) \ \exists\, z' : \ \ z\,z' = z_i\}$$

as oracle, $f(x)$ can be computed in polynomial time by determining first $|out_T(1^n)|$ and then applying a prefix search to find the optimum value in $out_T(1^n)$. $\square$

The known relationships of the language classes considered in this section are summarized in the diagram of Figure 3.1.

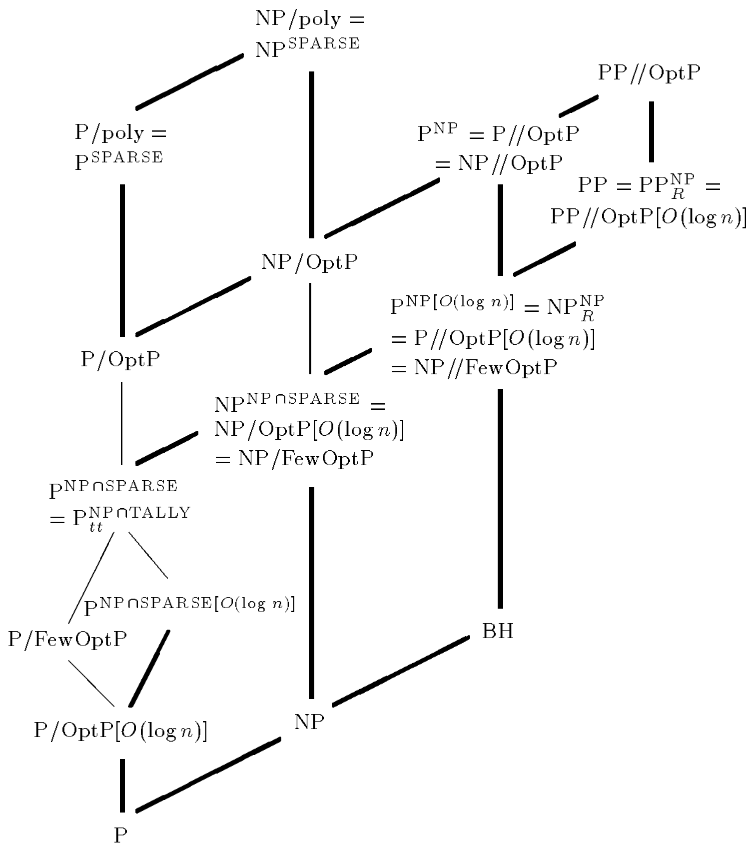

FIG. 3.1. *Inclusion structure of some considered complexity classes; thick lines indicate that there are relativized separations (see Section 4).*

4. Relativized Separations. Since Baker, Gill, and Solovay [2] separated P from NP relative to some oracle, relativizations have been an important subject in complexity theory. In this section, we discuss which of the inclusions in Figure 3.1 are strict, at least in some relativized world.

Since there are nonrecursive sets in P/poly and in NP/poly, these two classes are clearly different from all other (recursive) classes considered here. Whether there are any other strict inclusions in the diagram of Figure 3.1 is not known. For some of the inclusions, however, the question whether they are proper can be linked to central open problems in complexity theory.

For example, by the result of Karp, Lipton, and Sipser (see [23]) that NP $\subseteq$ P/poly implies the collapse of the polynomial hierarchy to its second level, it follows that if PH $\neq \Sigma_2$, then NP is not contained in any of the classes on the left column of Figure 3.1. Since this holds in all relativized worlds, and since there exists an oracle separating PH from Σ_2 [24], it follows that relative to this oracle all the inclusions between the first and the second column are proper.

Similarly, using the result of Kadin [21] that coNP $\subseteq \mathrm{NP}^{\mathrm{NP} \cap \mathrm{SPARSE}}$ implies $\mathrm{PH} = \mathrm{P}^{\mathrm{NP}[O(\log n)]}$, it follows that if $\mathrm{PH} \neq \mathrm{P}^{\mathrm{NP}[O(\log n)]}$, then $\mathrm{NP}^{\mathrm{NP} \cap \mathrm{SPARSE}} \neq \mathrm{P}^{\mathrm{NP}[O(\log n)]}$. Since, as it is easily seen, the inclusion coNP $\subseteq$ NP/OptP implies $\mathrm{PH} = \mathrm{P}^{\mathrm{NP}}$, we can state the following theorem.

THEOREM 4.1. $\mathrm{PH} \neq \mathrm{P}^{\mathrm{NP}} \;\Rightarrow\; \mathrm{NP/OptP} \neq \mathrm{P//OptP}$.

Furthermore, by the recent result of Toda [39] that $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$, it follows that $\mathrm{P}^{\mathrm{NP}[O(\log n)]} \neq \mathrm{PP}$ and $\mathrm{P}^{\mathrm{NP}} \neq \mathrm{PP//OptP}$ unless $\mathrm{PH} = \mathrm{P}^{\mathrm{NP}}$.

Beigel [7] constructed an oracle A such that $\mathrm{P}^{\mathrm{NP}^A} - \mathrm{PP}^A \neq \emptyset$. Since $\mathrm{P}^{\mathrm{NP}[O(\log n)]} \subseteq \mathrm{PP}$ [9], oracle A also separates $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ and P^{NP} (for a direct proof see [14]).

Cai et al. [15] showed the existence of an oracle A such that relative to A the boolean hierarchy is infinite, i.e., $\forall\, k : \mathrm{NP}^A(k) \neq \mathrm{coNP}^A(k)$. In fact, Cai et al. construct the oracle A in such a way that, for all k, some *tally* test language $L_k(A)$ is in $\mathrm{coNP}^A(k) - \mathrm{NP}^A(k)$. Because it holds for every oracle set B that

$$\mathrm{NP}^B(2^k - 1) \cup \mathrm{coNP}^B(2^k - 1) \;\subseteq\; \mathrm{P//OptP}^B[k] \;\subseteq\; \mathrm{NP}^B(2^k)$$

[43, 28, 8], it follows that $L_{2^k-1}(A) \in \mathrm{P//OptP}^A[k] \cap \mathrm{TALLY} \subseteq \mathrm{P/OptP}^A[k]$, i.e.,

$$\exists\, A \;\; \forall\, k \geq 1 : \mathrm{P/OptP}^A[k] - \mathrm{NP}^A(2^k - 1) \neq \emptyset.$$

Since $\mathrm{P/OptP}[k]$ is contained in the 2^k-th level of the boolean hierarchy, this result is optimal.

Clearly, if the boolean hierarchy is proper, it does not have complete sets. Since the class $\mathrm{P/OptP}[O(\log n)]$ has complete sets, it is not contained in BH in any relativized world where the boolean hierarchy is infinite, i.e.,

$$\exists\, A : \mathrm{P/OptP}^A[O(\log n)] - \mathrm{BH}^A \neq \emptyset.$$

The main result in this section is a separation of the classes $\mathrm{P/OptP}[O(\log n)]$ and $\mathrm{P}^{\mathrm{NP} \cap \mathrm{SPARSE}[O(\log n)]}$. In fact, we show that for any fixed polynomial q there is a relativization such that NP contains sparse sets that are not in the nonuniform class P/q (defined as $\mathrm{P}/\{h \mid |h(x)| \leq q(|x|)\}$).

THEOREM 4.2. *For every polynomial q there exists a set A such that*

$$(\mathrm{NP}^A \cap \mathrm{SPARSE}) - \mathrm{P}^A/q \neq \emptyset.$$

Proof. For an arbitrary set A we define a sparse set $L(A) \in \mathrm{NP}^A$ as follows. For a given n and a suitable choosen function $l(n)$, we partition the $2^{l(n)}$ words of length $l(n)$ into $q(n)+1$ intervals (with respect to the lexicographic ordering) $I_1^{l(n)}, \ldots, I_{q(n)+1}^{l(n)}$ such that

$$|I_k^{l(n)}| \geq \left\lfloor \frac{2^{l(n)}}{q(n)+1} \right\rfloor, \text{ for } k = 1, \ldots, q(n)+1.$$

For each interval containing a word in A, we put a word into $L(A)$: let $w_1^n, w_2^n, \ldots$ be an enumeration of Σ^n in lexicographic order and let $l(n) = n + q(n)$. Define the NP^A set

$$L(A) = \{w_k^n \mid n \geq 1,\ \ 1 \leq k \leq q(n)+1 \text{ and } I_k^{l(n)} \cap A \neq \emptyset\}.$$

Clearly, there are at most $q(n)+1$ words of length n in A, i.e., $L(A)$ is sparse. Now we construct a set A in stages such that $L(A) \notin \mathrm{P}/q$. Let $M_1, M_2, \ldots$ be an enumeration of all polynomial-time bounded Turing machines with running times $p_1, p_2, \ldots$, respectively.
Stage 0. $A := \emptyset$; $n_0 := \min\{n \mid \forall\, m \geq n: \ \ q(m) < 2^m\}$.
Stage $s \geq 1$. Choose n_s minimal such that $n_s > \max\{p_i(n_{s-1}) \mid i < s\}$ and $2^{n_s} > 2\, p_s(n_s)\, (q(n_s)+1)^2$.

The algorithm in Figure 4.1 determines the words of length $l(n_s)$ that are included in A. This is done by diagonalizing against machine M_s and all potential advice for M_s on an input of length n_s.

Let M be any P machine. We show that M, taking advice of any q-length bounded function, does not accept $L(A)$. Let s be an index such that $M = M_s$. There are $2^{q(n_s)+1} - 1$ potential words as advice for M_s on inputs of length n_s (that are stored in ADVICE). Each execution of the for-loop diagonalizes against at least half of the possible advice for M_s. Since $\log(2^{q(n_s)+1} - 1) \leq q(n_s) + 1$, ADVICE becomes empty at the end of the algorithm. The construction further guarantees that for every advice a, $|a| \leq q(n_s)$, there exists a $k \leq q(n_s)+1$ such that

$$\langle w_k^{n_s}, a \rangle \in L(M_s, A) \ \ \Leftrightarrow \ \ w_k^{n_s} \notin L(A).$$

Therefore, it suffices to show that the algorithm can always find a $y \in I_k^{l(n_s)} -$ QUERY. In every execution of the for-loop and for every advice no more than $p_s(n_s)$ words are added to the set QUERY, i.e.,

$$|\text{ QUERY }| \leq (q(n_s)+1)\, 2^{q(n_s)+1}\, p_s(n_s).$$

Thus, we have for $1 \leq k \leq q+1$,

$$\begin{aligned} |I_k^{l(n_s)} - \text{ QUERY }| &\geq |I_k^{l(n_s)}| - |\text{ QUERY }| \\ &\geq \left\lfloor \frac{2^{n_s+q(n_s)}}{q(n_s)+1} \right\rfloor - (q(n_s)+1)\, 2^{q(n_s)+1}\, p_s(n_s) \end{aligned}$$

ADVICE := $\Sigma^{\leq q(n_s)}$;
(* ADVICE contains all potential advice against that we have to diagonalize *)
QUERY := $\emptyset$;
(* In QUERY we freeze the oracle queries of M_s during the construction *)
for $k := 1$ **to** $q(n_s)+1$ **do**
  ACC := $\{a \in \text{ADVICE} \mid M_s^A(w_k^{n_s}, a) \text{ accepts } \}$;
  REJ := ADVICE − ACC;
  if |ACC| ≥ |REJ| **then**
    (* $I_k^{l(n_s)} \cap A$ remains empty, i.e., no word in ACC is advice for $w_k^{n_s}$ *)
    ADVICE := REJ;
    QUERY := QUERY $\cup \bigcup_{a \in \text{ACC}} Q(M_s, A, \langle w_k^{n_s}, a\rangle)$;
  else
    (* put a word in $I_k^{l(n_s)} \cap A$, i.e., no word in REJ is advice for $w_k^{n_s}$ *)
    ADVICE := ACC;
    QUERY := QUERY $\cup \bigcup_{a \in \text{REJ}} Q(M_s, A, \langle w_k^{n_s}, a\rangle)$;
    choose a $y \in I_k^{l(n_s)}$−QUERY;
    $A := A \cup \{y\}$
  end (* if *)
end (* for *).

FIG. 4.1. *Algorithm used in the proof of Theorem 4.2.*

$$
\begin{aligned}
&\geq \frac{2^{q(n_s)+1}\left(2^{n_s-1} - (q(n_s)+1)^2\, p_s(n_s)\right)}{q(n_s)+1} - 1 \\
&\geq \frac{2^{q(n_s)+1}}{q(n_s)+1} - 1 \qquad \text{by choice of } n_s \\
&> 0.
\end{aligned}
$$

□

COROLLARY 4.3. $\exists\, A : (\text{NP}^A \cap \text{SPARSE}) - \text{P}^A/\log \neq \emptyset$.

Using a "Kolmogorov-argument", Corollary 4.3 was already shown by Hemachandra [19]. An immediate consequence of Corollary 4.3 is the existence of an oracle separating $\text{P}/\text{OptP}[O(\log n)]$ and $\text{P}^{\text{NP}\cap\text{SPARSE}[O(\log n)]}$.

COROLLARY 4.4. $\exists\, A : \text{P}^A/\text{OptP}^A[O(\log n)] \neq \text{P}^{\text{NP}^A\cap\text{SPARSE}[O(\log n)]}$.

5. Bounded Advice versus the Boolean Hierarchy. The levels of the boolean hierarchy build as their union the boolean closure of NP, i.e. the smallest class of sets that contains NP and is closed under union, intersection, and complementation. In this section, we give several characterizations of the odd levels of the boolean hierarchy. First, we show that NP machines that get as advice the value of the k-ary characteristic function χ_k^{SAT} of SAT, where χ_k^{SAT} is evaluated on a k-tuple that is computed from the input by an FP function, accept exactly the languages in the $(2k+1)$-th level of the boolean hierarchy. The same is true if the advice consists only of the information of how many out of k words that are produced from the given input by an FP function are in SAT.

Cai et al. [15] give several characterizations of the boolean hierarchy, we take the following.

DEFINITION 5.1. *A set L is in the k-th level $\mathrm{NP}(k)$ of the boolean hierarchy, if there exist sets $L_1, \ldots, L_k \in \mathrm{NP}$ such that*

$$L = \begin{cases} (L_1 - L_2) \cup \ldots \cup (L_{k-2} - L_{k-1}) \cup L_k, & \text{if } k \text{ is odd} \\ (L_1 - L_2) \cup \ldots \cup (L_{k-1} - L_k), & \text{if } k \text{ is even} \end{cases}$$

The union $\bigcup_{k\geq 1} \mathrm{NP}(k)$ of all the levels of the boolean hierarchy is denoted by BH.

For a set A, χ^A denotes the characteristic function of A. χ_k^A is the k-ary characteristic function of A, $\#_k^A$ gives the number out of k words, that are in A, and $\oplus_k^A$ is the parity of this number, i.e.,

$$\begin{aligned} \chi_k^A(x_1, \ldots, x_k) &= \chi^A(x_1) \ldots \chi^A(x_k), \\ \#_k^A(x_1, \ldots, x_k) &= \sum_{i=1}^{k} \chi^A(x_i), \\ \oplus_k^A(x_1, \ldots, x_k) &= \#_k^A(x_1, \ldots, x_k) \bmod 2. \end{aligned}$$

The unbounded version of χ_k^A is $\chi_\omega^A = \bigcup\limits_{k\geq 1} \chi_k^A$.

Clearly, every set $L \in \mathrm{NP}(k)$ is k-truth-table reducible to SAT, i.e., $L \in \mathrm{P}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$ (here and in the following, the composition operator $\circ$ takes precedence over $//$). Every set that is k-truth-table reducible to SAT is in $\mathrm{NP}(k+1)$ ([28], see also [8]), thus

$$\mathrm{NP}(k) \subseteq \mathrm{P}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP} \subseteq \mathrm{NP}(k+1). \tag{5.1}$$

Since $\mathrm{P}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$ is closed under complementation, the classes in (5.1) are all different unless BH (and therefore PH [20]) collapses. It is interesting to note that a P machine needs only to know the parity of the number of k queries in SAT in order to decide a set in $\mathrm{P}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$ ([43], see also [8]),

$$\mathrm{P}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP} = \mathrm{P}//\#_k^{\mathrm{SAT}} \circ \mathrm{FP} = \mathrm{P}// \oplus_k^{\mathrm{SAT}} \circ \mathrm{FP}. \tag{5.2}$$

We show in the next theorem that the first equality in (5.2) also holds for the nondeterministic counterparts of these classes which furthermore coincide with the $(2k+1)$-th level of the boolean hierarchy. Since, as it is easily seen, $\mathrm{NP}// \oplus_k^{\mathrm{SAT}} \circ \mathrm{FP}$

is contained in $\mathrm{P}//\chi_{k+2}^{\mathrm{SAT}} \circ \mathrm{FP}$, we cannot replace P by NP, for $k \geq 2$, in the second equality of (5.2), unless BH, and thus PH, collapse. It is an open question whether also the classes $\mathrm{NP}// \oplus_k^{\mathrm{SAT}} \circ \mathrm{FP}$ characterize some levels of the boolean hierarchy.

We denote the bitwise ordering on strings of the same length by $\preceq$, i.e., $a_1 \dots a_k \preceq b_1 \dots b_k$, if $a_i \leq b_i$, for $i = 1, \dots, k$.

THEOREM 5.2. $\mathrm{NP}(2k+1) = \mathrm{NP}//\#_k^{\mathrm{SAT}} \circ \mathrm{FP} = \mathrm{NP}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$, *for all* $k \geq 0$.

Proof. Let L be in $\mathrm{NP}(2k+1)$. Then there exist sets $L_1, \dots, L_{2k+1} \in \mathrm{NP}$ such that $L = (L_1 - L_2) \cup \dots \cup (L_{2k-1} - L_{2k}) \cup L_{2k+1}$. Define the function

$$f(x) = \sum_{i=1}^{k} \chi^{L_{2i}}(x)$$

and let A be the set defined as

$$\begin{array}{lcl}\langle x, m\rangle \in A & \Leftrightarrow & x \in L_{2k+1} \text{ or there exist } I \subseteq \{2i \mid x \in L_{2i}\} \text{ and } j \leq k \\ & & \text{such that } |I| = m,\ x \in L_{2j-1}, \text{ and } 2j \notin I.\end{array}$$

Clearly, $f \in \#_k^{\mathrm{SAT}} \circ \mathrm{FP}$ and $A \in \mathrm{NP}$, and it holds that $x \in L$ if and only if $\langle x, f(x)\rangle \in A$. To see this, observe that there is exactly one set $I \subseteq \{2i \mid x \in L_{2i}\}$ of cardinality $f(x)$, namely $I = \{2i \mid x \in L_{2i}\}$. Therefore, $L \in \mathrm{NP}//\#_k^{\mathrm{SAT}} \circ \mathrm{FP}$.

It is clear that $\mathrm{NP}//\#_k^{\mathrm{SAT}} \circ \mathrm{FP} \subseteq \mathrm{NP}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$. It remains to show that $\mathrm{NP}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP} \subseteq \mathrm{NP}(2k+1)$. For this we adapt a proof technique used by Buss and Hay [14]. Let L be in $\mathrm{NP}//\chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$, i.e., there exist a set $A \in \mathrm{NP}$ and a function $f \in \mathrm{FP}$ such that $x \in L$ if and only if $\langle x, \chi_k^{\mathrm{SAT}}(f(x))\rangle \in A$. For $m \geq 0$, consider the NP sets

$$\begin{array}{lcl}B_m & = & \{x \mid \#_k^{\mathrm{SAT}}(f(x)) \geq m\}, \\ A_m & = & \{x \mid \exists\, a = a_1 \dots a_k \in \Sigma^k : \displaystyle\sum_{i=1}^{k} a_i = m,\ \ a \preceq \chi_k^{\mathrm{SAT}}(f(x)), \text{ and } \langle x, a\rangle \in A\}.\end{array}$$

It is easy to see that $A_m \subseteq B_m$ and $B_{m+1} \subseteq B_m$. Furthermore, $B_m - B_{m+1} = \{x \mid \#_k^{\mathrm{SAT}}(f(x)) = m\}$ and $A_m - B_{m+1} = \{x \in B_m - B_{m+1} \mid \langle x, \chi_k^{\mathrm{SAT}}(f(x))\rangle \in A\}$. The latter equality follows from the fact that for any $x \in B_m - B_{m+1}$, there is only one string $a \in \Sigma^k$ containing m 1's and fulfilling $a \preceq \chi_k^{\mathrm{SAT}}(f(x))$, namely $a = \chi_k^{\mathrm{SAT}}(f(x))$. Therefore, $x \in L$ if and only if $x \in A_m - B_{m+1}$, for some $m \leq k$. Since $B_{k+1} = \emptyset$, it follows that $L = (A_0 - B_1) \cup \dots \cup (A_{k-1} - B_k) \cup A_k$. □

Hemachandra [18] (see also Buss and Hay [14]) has shown that the classes $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ and $\mathrm{P}//\chi_\omega^{\mathrm{SAT}} \circ \mathrm{FP}$ coincide. By a slight modification in the above proof we get the following corollary yielding a further characterization of $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$.

COROLLARY 5.3. $\mathrm{P}//\chi_\omega^{\mathrm{SAT}} \circ \mathrm{FP} = \mathrm{NP}//\chi_\omega^{\mathrm{SAT}} \circ \mathrm{FP}$.

Beigel [8] shows that $\mathrm{P}//\mathrm{OptP}[k] = \mathrm{P}//\chi_{2^k-1}^{\mathrm{SAT}} \circ \mathrm{FP}$. From Theorem 5.2 and the following Theorem 5.4, it follows that this equation remains valid when P is replaced by NP. Theorem 5.4 restates an observation in [26] that $\#^{\mathrm{SAT}}_{2^k-1}$ is complete for OptP[k].

THEOREM 5.4. [26] $\mathrm{OptP}[k] = \#^{\mathrm{SAT}}_{2^k-1} \circ \mathrm{FP} \cup \#^{\overline{\mathrm{SAT}}}_{2^k-1} \circ \mathrm{FP}$, *for all* $k \geq 0$.

COROLLARY 5.5. $\mathrm{NP}(2^{k+1}-1) = \mathrm{NP}//\mathrm{OptP}[k]$, *for all* $k \geq 0$.

Ladner, Lynch, and Selman [30] transformed the recursion theoretic truth-table reducibility into complexity theory. They also give a definition of a nondeterministic truth-table reduction in the following way: A is nondeterministically truth-table reducible to B, if there exists an NP transducer G (the *generator*) and an NP machine E (the *evaluator*) such that for every x,

$$x \in A \quad \Leftrightarrow \quad \begin{array}{l}\text{there exists a branch of } G(x) \text{ yielding an output}\\ y = \langle y_1, \ldots, y_k\rangle \text{ such that } E(x, \chi^B_\omega(y_1, \ldots, y_k)) \text{ accepts.}\end{array}$$

It is known that this definition is equivalent with the nondeterministic Turing reducibility [30] and therefore does not lead to a new reducibility notion. We modify the above definition by restricting the generator G to be a *single-valued* NP transducer, i.e., the output must be the same on every accepting branch. Let NPSV be the set of functions computed by single-valued NP transducers [11].

This reducibility first appeared in [11] (there denoted by NP.UNIF.ALL), and was explicitely called *nondeterministic truth-table reducibility* by Book and Ko [10]. Subsequently, Book and Tang [12] and Rich [35] introduced the following terminology.

DEFINITION 5.6. *A set A is* nondeterministically truth-table reducible *to B ($A \leq^{\mathrm{NP}}_{tt} B$), if $A \in \mathrm{NP}//\chi^B_\omega \circ \mathrm{NPSV}$. A is* nondeterministically k-truth-table reducible *to B ($A \leq^{\mathrm{NP}}_{k\text{-}tt} B$), if $A \in \mathrm{NP}//\chi^B_k \circ \mathrm{NPSV}$. For a class $\mathcal{C}$ of sets let $\mathrm{NP}^{\mathcal{C}}_{tt}$ be the class $\{A \mid \exists B \in \mathcal{C} : A \leq^{\mathrm{NP}}_{tt} B\}$ of all sets $\leq^{\mathrm{NP}}_{tt}$-reducible to some set in $\mathcal{C}$, and let $\mathrm{NP}^{\mathcal{C}}_{k\text{-}tt} = \{A \mid \exists B \in \mathcal{C} : A \leq^{\mathrm{NP}}_{k\text{-}tt} B\}$.*

In [11], it is shown that there exist recursive sets A and B such that $A \leq^{\mathrm{NP}}_{T} B$ and $A \not\leq^{\mathrm{NP}}_{tt} B$. This means that $\leq^{\mathrm{NP}}_{tt}$ is properly stronger than $\leq^{\mathrm{NP}}_{T}$. The question whether $\leq^{\mathrm{P}}_{tt}$ is properly stronger than $\leq^{\mathrm{NP}}_{tt}$ is equivalent to the P =?NP problem [11, 35]. However, as we will see in Corollary 5.8, every set A that is nondeterministically truth-table reducible to some NP-complete set B is also deterministically truth-table reducible to B, i.e.,

$$A \leq^{\mathrm{NP}}_{tt} B \Rightarrow A \leq^{\mathrm{P}}_{tt} B.$$

Thus, we have the surprising result that while the definition in [30] of a nondeterministic truth-table reduction was too weak, the definition of Rich seems to be too strong to yield a new reduction class between $\{L \mid L \leq^{\mathrm{P}}_{tt} \mathrm{SAT}\}$ and $\{L \mid L \leq^{\mathrm{NP}}_{T} \mathrm{SAT}\}$. As a further consequence of Theorem 5.7, we get a characterization of the odd levels of the boolean hierarchy in terms of the nondeterministic k-truth-table reducibility notion.

THEOREM 5.7.

i) $\chi_k^{\mathrm{SAT}} \circ \mathrm{NPSV} = \chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$ *for all* $k \geq 1$,

ii) $\chi_\omega^{\mathrm{SAT}} \circ \mathrm{NPSV} \subseteq \mathrm{FP}_{tt}^{\mathrm{SAT}}$.

Proof. To see *i)* let f be in NPSV and define the NP set

$$A = \{\langle x, m\rangle \mid \exists\, z_1, \ldots, z_k : f(x) = (z_1, \ldots, z_k) \text{ and } z_m \in \mathrm{SAT}\}.$$

Then $\chi_k^{\mathrm{SAT}}(f(x)) = \chi_k^A(\langle x, 1\rangle, \ldots, \langle x, k\rangle)$ for all x, and thus, $\chi_k^{\mathrm{SAT}} \circ f \in \chi_k^A \circ \mathrm{FP} \subseteq \chi_k^{\mathrm{SAT}} \circ \mathrm{FP}$.

For the proof of *ii)* define the NP set

$$B = \{\langle x, k, m, b\rangle \mid \exists\, z_1, \ldots, z_k : f(x) = (z_1, \ldots, z_k) \text{ and } b \leq \chi^{\mathrm{SAT}}(z_m)\},$$

and observe that $\chi_\omega^{\mathrm{SAT}}(f(x))$ can be read off $B's$ answers to the parallel queries $\langle x, k, m, b\rangle$, for $k = 1, \ldots, p(|x|)$, $m = 1, \ldots, k$, and $b = 0, 1$, where p is a polynomial bounding the running time of the NP transducer that computes f. □

Corollary 5.8.

i) $\mathrm{NP}(2k+1) = \mathrm{NP}_{k\text{-}tt}^{\mathrm{NP}}$, *for all* $k \geq 1$,

ii) $\mathrm{P}_{tt}^{\mathrm{NP}} = \mathrm{NP}_{tt}^{\mathrm{NP}}$.

Remark 5.9. *Book and Tang [12] especially consider the* $O(\log n)$ *bounded version* $\leq_{\log n\text{-}tt}^{\mathrm{NP}}$ *of the nondeterministic truth-table reduction obtained by logarithmically bounding the number of queries produced by the* NPSV *generator. It follows from (appropriately modified versions of) Theorem 5.7, Corollary 5.3, and Lemma 3.6 that*

$$\mathrm{NP}_{\log n\text{-}tt}^{\mathrm{NP}} = \mathrm{P}_{\log n\text{-}tt}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}[O(1)+\log\log n]}.$$

This class is also considered by Wagner [42] (there denoted by $\mathrm{P}_{||}^{\mathrm{NP}}[O(\log n)]$*), who shows that it coincides with the class of languages that are full-truth-table reducible*[2] *to* SAT. *As a consequence, it follows that* $A \leq_{\log n\text{-}tt}^{\mathrm{NP}} \mathrm{SAT}$ *if and only if* A *is full-truth-table reducible to* SAT.

Remark 5.10. *Book and Tang [12] generalized the nondeterministic truth-table reducibility to a* Σ_k *truth-table reducibility by giving the generator and the evaluator access to a* Σ_{k-1} *oracle:* A *is* Σ_k truth-table reducible *to* B $(A \leq_{tt}^{\Sigma_k} B)$, *if* $A \in \Sigma_k//\chi_\omega^B \circ \mathrm{NPSV}^{\Sigma_{k-1}}$. *For a class* $\mathcal{C}$ *of sets let* $\Sigma_{k,tt}^{\mathcal{C}}$ *be the class* $\{A \mid \exists\, B \in \mathcal{C} : A \leq_{tt}^{\Sigma_k} B\}$. *Then Corollary 5.8 ii) generalizes to*

$$\Sigma_{k,tt}^{\Sigma_k} = \mathrm{P}_{tt}^{\Sigma_k} = \mathrm{P}^{\Sigma_k[O(\log n)]},$$

i.e., every set that is Σ_k *truth-table reducible to a set in* Σ_k *is already (deterministically) truth-table reducible to a set in* Σ_k.

[2] A set A is full-truth-table reducible [28, 14] to a set B, if there is a function $g \in \mathrm{FP}$ such that for all x, $g(x)$ is of the form $\langle a_0 \ldots a_{2^m-1}, y_1, \ldots, y_m\rangle$, where $a_i \in \{0, 1\}$ $(0 \leq i \leq 2^m - 1)$, and $y_i \in \Sigma^*$ $(1 \leq i \leq m)$, and it holds that $x \in A \Leftrightarrow a_j = 1$, where j is the number whose binary representation is given by $\chi_\omega^B(y_1, \ldots, y_m)$.

Thierauf [38] showed that allowing the generator in the nondeterministic truth-table reduction to produce polynomially many different outputs (i.e., to compute an NPPV function [11]) does not increase the class of sets reducible to SAT.

THEOREM 5.11. [38] *Let L is a set, G an* NPPV *transducer, and E an* NP *set such that*

$$x \in L \;\Leftrightarrow\; \exists \langle y_1, \ldots, y_k\rangle \in out_G(x): \;\; \langle x, \chi_\omega^{\mathrm{SAT}}(y_1, \ldots, y_k)\rangle \in E,$$

then L is in $\mathrm{P}_{tt}^{\mathrm{NP}}$.

We end this section by proving that also the strong nondeterministic truth-table reducibility, introduced by Long [31], when applied to SAT, is only as powerful as $\leq_{tt}^{\mathrm{P}}$. Like in the definition of Ladner, Lynch, and Selman [30], the generator in a strong nondeterministic truth-table reduction can produce exponentially many different outputs, but the evaluator either has to accept all the outputs or it has to reject all of them.

DEFINITION 5.12. [31] *A is* strong nondeterministic truth-table reducible *to B ($A \leq_{tt}^{\mathrm{SN}} B$), if there is an* NP *transducer G and a* P *machine E such that for all x the set $out_G(x)$ is nonempty, and for all $\langle y_1, \ldots, y_k\rangle$ in $out_G(x)$, $E(x, \chi_\omega^B(y_1, \ldots, y_k)) = \chi^A(x)$. For a class of sets $\mathcal{C}$ we denote by $\mathrm{SN}_{tt}^{\mathcal{C}}$ the class $\{A \mid \exists\, B \in \mathcal{C} : A \leq_{tt}^{\mathrm{SN}} B\}$.*

Clearly, $\leq_{tt}^{\mathrm{SN}}$ lies in strength between $\leq_{tt}^{\mathrm{P}}$ and $\leq_{T}^{\mathrm{NP}}$. Long [31] showed that $\leq_{tt}^{\mathrm{SN}}$ is properly stronger than $\leq_{T}^{\mathrm{NP}}$ by constructing two sets A and B such that $A \not\leq_{tt}^{\mathrm{SN}} B$ and $A \leq_{T}^{\mathrm{NP}} B$. The question whether $\leq_{tt}^{\mathrm{P}}$ is properly stronger than $\leq_{tt}^{\mathrm{SN}}$ is closely related to two major open questions in complexity theory [31]:

$$\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP} \;\;\Rightarrow\;\; \leq_{tt}^{\mathrm{P}} \neq \leq_{tt}^{\mathrm{SN}} \;\;\Rightarrow\;\; \mathrm{P} \neq \mathrm{NP}.$$

THEOREM 5.13. $\mathrm{SN}_{tt}^{\mathrm{NP}} = \mathrm{P}_{tt}^{\mathrm{NP}}$.

Proof. Let L be in $\mathrm{SN}_{tt}^{\mathrm{NP}}$ via a generator G, an evaluator E, and a set $A \in \mathrm{NP}$. In order to decide membership of a given input x, it suffices to find out whether there is some output $\langle y_1, \ldots, y_k\rangle$ of $G(x)$ such that E accepts $\langle x, \chi_\omega^A(y_1, \ldots, y_k)\rangle$. But this becomes an NP problem, provided that the maximum number $\#_\omega^A(y_1, \ldots, y_k)$ of yes-answers from A over all outputs $\langle y_1, \ldots, y_k\rangle$ of $G(x)$ is given along with the input x.

More precisely, define the function

$$h(x) = \max\{\#_\omega^A(y_1, \ldots, y_k) \mid \langle y_1, \ldots, y_k\rangle \in out_G(x)\}$$

and let B be the set defined as

$$\begin{aligned}\langle x, m\rangle \in B \quad \Leftrightarrow \quad & \exists\, a = a_1 \ldots a_k \in \Sigma^k \;\exists \langle y_1, \ldots, y_k\rangle \in out_G(x): \\ & \sum_{i=1}^{k} a_i = m, \;\; a \preceq \chi_\omega^A(y_1, \ldots, y_k) \text{ and } E(x, a) = 1.\end{aligned}$$

Then $h \in \mathrm{OptP}[O(\log n)]$ and $B \in \mathrm{NP}$, and it holds for all x that $x \in L$ if and only if $\langle x, h(x)\rangle \in B$, i.e., L is in $\mathrm{NP}//\mathrm{OptP}[O(\log n)] = \mathrm{P}//\mathrm{OptP}[O(\log n)]$. $\square$

Note that by the above proof, Theorem 5.13 remains true if the evaluator E is allowed to be an NP machine.

Acknowledgments. For helpful conversations and suggestions on this work we are grateful to V. Arvind, L. Hemachandra, and U. Schöning. We also thank the two referees for their valuable suggestions.

REFERENCES

[1] A. Amir, W. Gasarch. Polynomial terse sets. In *Proceedings 2nd Structure in Complexity Theory Conference,* p. 259-269. IEEE Computer Society, 1987.

[2] T. Baker, J. Gill, R. Solovay. Relativizations of the P =?NP question. In *SIAM Journal on Computing*, 4(4):431-442, 1975.

[3] J. Balcázar. Nondeterministic witnesses and nonuniform advice. In *Proceedings 4th Structure in Complexity Theory Conference,* p. 259-269. IEEE Computer Society, 1989.

[4] J. Balcázar, J. Díaz, J. Gabarró. *Structural Complexity I.* Springer, 1987.

[5] J. Balcázar, U. Schöning. Logarithmic advice classes. In *Theoretical Computer Science*, 99:279-290, 1992.

[6] R. Beigel. Query-limited reducibilities. Dissertation. Stanford University, 1987.

[7] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. In *Proceedings 7th Structure in Complexity Theory Conference,* p. 14-19. IEEE Computer Society, 1989.

[8] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. In *Theoretical Computer Science*, 84(2):199-223, 1991.

[9] R. Beigel, L. Hemachandra, G. Wechsung. Probabilistic polynomial time is closed under parity reductions. In *Information Processing Letters*, 37(2):91-94, 1991.

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

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

[12] R. Book, S. Tang. Characterizing Polynomial Complexity Classes by Reducibilities. In *Mathematical System Theory*, 23(3):165-174, 1990.

[13] R. Book, S. Tang. A note on sparse sets and the polynomial-time hierarchy. In *Information Processing Letters*, 33:141-143, 1989.

[14] S. Buss, L. Hay. On Truth-Table Reducibility to SAT. In *Information and Computation*, 91(1):86-102, 1991.

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

[16] A.Chandra, D. Kozen, L. Stockmeyer. Alternation. In *Journal of the Association for Computing Machinery*, 28(1):114-133, 1981.

[17] J. Hartmanis, Y. Yesha. Computation times of NP sets of different densities. In *Automata, Languages and Programming,* p. 319-330, Springer-Verlag *Lecture Notes in Computer Science*, 154, 1983.

[18] L. Hemachandra. The strong exponential hierarchy collapses. In *Proc. of the 19th ACM Symposium on Theory of Computing,* p. 110-122, 1987.

[19] L. Hemachandra. Personal communication. Barcelona 1990.

[20] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. In *SIAM Journal on Computing*, 17(6), p. 1263-1282, December 1988.

[21] J. Kadin. P^{NP} and sparse Turing-complete Sets for NP. In *Journal of Computer and System Sciences*, 39:282-298, 1989.

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

[23] R. Karp, R. Lipton. Turing machines that take advice. In *L'Enseignement Mathématique*, 28:191-209, 1982.

[24] K. Ko. Relativized polynomial time hierarchies having exactly k levels. In *SIAM Journal on Computing*, 18(2):392-408, 1989.

[25] J. Köbler. *Strukturelle Komplexität von Anzahlproblemen.* Ph.D. Thesis. University of Stuttgart, 1989.

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

[27] J. Köbler, U. Schöning, S. Toda, J. Torán. Turing machines with few accepting computations and low sets for PP. In *Journal of Computer and System Sciences,* 44(2):272-286, 1992.

[28] J. Köbler, U. Schöning, K. Wagner. The difference and truth-table hierarchies of NP. In *Theoretical Informatics and Applications*, 21(4):419-435, 1987.

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

[30] R. Ladner, N. Lynch, A. Selman. A comparison of polynomial-time reducibilities. In *Theoretical Computer Science* 1, p. 103-123, 1975.

[31] T. Long. Strong nondeterministic polynomial-time reducibilities. In *Theoretical Computer Science* 21, p. 1-25, 1982.

[32] T. Long. On restricting the size of oracles compared with restricting access to the oracle. In *SIAM Journal on Computing*, 14(3):585-597, 1985.

[33] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. In *Journal of Computer and System Sciences* 25:130-143, 1982.

[34] N. Pippenger. On simultaneous resource bounds. In *Proceedings 20th Sym-*

posium on Foundations of Computer Science, p. 307-311, IEEE Computer Society 1979.

[35] C. RICH. Positive relatvizations of the P =?NP problem. In *Journal of Computer and System Sciences* 38, p. 511-523, 1989.

[36] U. SCHÖNING. *Complexity and Structure.* Springer-Verlag *Lecture Notes in Computer Science* 211, 1986.

[37] L. STOCKMEYER. The polynomial time hierarchy. In *Theoretical Computer Science*, 3:1-22, 1977.

[38] T. THIERAUF. NP-*uniforme Komplexitätsklassen.* Ph.D. Thesis. University of Ulm, 1990.

[39] S. TODA. PP is as hard as the polynomial-time hierarchy. In *SIAM Journal on Computing* 20(5):865–877, 1991.

[40] S. TODA. Restricted relativizations of probabilistic polynomial time. In *Theoretical Computer Science*, 93(2):265-277, 1992.

[41] K. WAGNER. On restricting the access to an NP-oracle. In *Automata, Languages and Programming,* p. 682-692, Springer-Verlag *Lecture Notes in Computer Science*, 317, *ICALP* 1988.

[42] K. WAGNER. Bounded query computations. In *SIAM Journal on Computing,* 19(5):833-846, 1990.

[43] G. WECHSUNG. On the boolean closure of NP. In *Proceedings of the 5th Conference on Fundamentals of Computation Theory*, pages 485–493. Springer-Verlag *Lecture Notes in Computer Science*, 199, 1985. (An unpublished precursor of this paper was coauthored by K. Wagner).