Bounding Queries in the Analytic Polynomial-Time Hierarchy

Herbert Baier* Klaus W. Wagner
Lehrstuhl für Theoretische Informatik
Universität Würzburg
Am Exerzierplatz 3
D-97072 Würzburg, Germany
[baier,wagner]@informatik.uni-wuerzburg.de

August 13, 1997

Abstract

In a previous paper the present authors [4] investigated an $\exists$-$\forall$-hierarchy over *P* using word quantifiers as well as two types of set quantifiers, the so called analytic polynomial-time hierarchy. The fact that some constructions there result in a bounded number of oracle queries and the recent *PCP* results which can be expressed by set quantifiers with a bounded number of queries motivated us to examine a hierarchy which extends the analytic polynomial-time hierarchy by considering restrictions on the number of oracle queries. This hierarchy is called *bounded analytic polynomial-time hierarchy*. We show that every class from this hierarchy having a certain normal form coincides with one of the classes *NP*, co*NP*, *PSPACE*, Σ_k^{exp} or Π_k^{exp} ($k \geq 1$). All these characterizations remain valid if the queries are asked in a nonadaptive form, i.e. in "parallel".

1 Introduction

Quantifiers play an important role in the complexity theory. Interesting complexity classes can be defined (or characterized) by quantifiers on the base of some other complexity class. Take for example the classes of the (arithmetical) polynomial-time hierarchy [11, 10, 13], which can be characterized by polynomial length bounded existential and universal word quantifiers on the base of *P*.

Another way to consider quantifiers are varying over set of words (oracles if we think in a Turing machine model). In 1983, Orponen [8] investigated the existential and universal set quantifiers. He related a hierarchy defined by these quantifiers on the base of the class *PH* with the classes of the exponential-time alternation hierarchy. In 1988, Fortnow, Rompel, and Sipser [7] showed

*Supported by **CNPq** grant 290020/92.1 (Brazil) and in part by **DAAD** (Germany).

that the class of languages accepted by multi-prover interactive proof systems (*MIP*) coincides with the class defined by an existential set quantifier on the base of the polynomial-time bounded error probability class *BPP*. In 1990, Babai, Fortnow, and Lund [3] proved *MIP* = *NEXPTIME*. In 1992, Arora and Safra [2] introduced the notion of probabilistically checkable proofs (*PCP*) to "scale down" the previous result. The class *PCP* $(r(n), q(n))$ can be defined as an existential set quantifier applied to *BPP*, where an underlying machine is allowed to use $O(r(n))$ random bits for its computation and queries the oracle $O(q(n))$ times. Arora and Safra characterized the class *NP* by *PCP* $\left(\log n, (\log\log n)^{O(1)}\right)$ and few weeks later Arora, Lund, Motwani, Sudan, and Szegedy [1] improve this result showing *NP* = *PCP* $(\log n, O(1))$. A detailed discussion of scaling down results in this area can be found in [12]. In 1996, Book, Vollmer, and Wagner [5] investigated the power of probabilistic set quantifiers.

Recently, using a restricted oracle access mechanism Baier and Wagner [4] characterized the (one prover) interactive proof systems by an existential set quantifier applied to *BPP*. An $\exists$-$\forall$-hierarchy over *P* using word quantifiers as well as two types of set quantifiers was also considered, the so called analytic polynomial-time hierarchy. It was shown that every class of this hierarchy coincides with one of the classes Σ_k^p, Π_k^p $(k \geq 0)$, *PSPACE*, Σ_k^{exp} or Π_k^{exp} $(k \geq 1)$.

Some constructions in [4] result in a bounded number of oracle queries. This fact and the interesting results obtained in the study of the *PCP* classes [2, 1], which limit the number of oracle queries, motivate us to continue the study of the analytic polynomial-time hierarchy classes but now considering the number of oracle queries that an oracle machine can ask during its computation. We call this hierarchy the *bounded analytic polynomial-time hierarchy*. For every class of this hierarchy having a certain normal form we show that this class coincides with one of the classes *NP*, co*NP*, *PSPACE*, Σ_k^{exp} or Π_k^{exp} $(k \geq 1)$. In addition it is shown that all these characterizations remain valid if the oracle machines are allowed to make only parallel queries, i.e. they have to form a list of all queries before any of them is queried to the oracle.

2 Preliminaries

We will study a polynomial-time hierarchy built up by word and set operators (defined by word and set quantifiers, respectively) which extends the analytic polynomial-time hierarchy [4]. Thus, we have to start with a suitable class of polynomial time languages. Since the instances of these languages consist of words and sets of words, we will use polynomial-time oracle Turing machines to accept these languages. Every word input is given on a separate input tape and every set input is given as an oracle. The machines have a special query tape for every oracle. The oracles can be classified according to the type of queries that can be made by an oracle machine. An oracle is an *input of type 1* (*2*) if the query on the corresponding query tape is not erased (erased, respectively) after every query. Hence, the next query made to an oracle of type 1 is an extension of the previous query. Note that formally inputs of type 1 and 2 are the same objects, namely sets of words. Furthermore, if we limit the number of queries that a machine can ask to an oracle (input of type $\sigma \in \{1, 2\}$) during its computation by a function $r : \mathbb{N} \to \mathbb{N}$ depending on the length of the word inputs, i.e. if n is the length of the word inputs then at most $r(n)$ queries can be asked to the oracle, we say that this is an *input of type* $\sigma[r]$. We will call a word an *input of type 0*.

Let $\Phi =_{df} \{0, 1, 2\} \cup \{\sigma[r] : \sigma \in \{1, 2\}, r : \mathbb{N} \to \mathbb{N}^+\}$ be the set of types. Furthermore, for $Q \in \{\exists, \forall\}$ let $\Gamma_Q =_{df} \{Q^\tau : \tau \in \Phi\}$, and let $\Gamma =_{df} \Gamma_\exists \cup \Gamma_\forall$. For $k \geq 1$ and $\tau_1, \ldots, \tau_k \in \Phi$ we say that a Turing machine is of type $\tau_1 \dots \tau_k$ if it precesses instances of the form $(X_1, \dots, X_k)$, where X_i is an input of type τ_i $(i = 1, \dots, k)$. Such a machine has $\#\{i : \tau_i = 0\}$ ordinary input tapes and $k - \#\{i : \tau_i = 0\}$ query tapes. Thus, if the i-th input is of type $\tau_i = \sigma[r]$ with $\sigma \in \{1, 2\}$ then at most $r\left(\sum_{j<i,\tau_j=0} |X_j|\right)$ queries can be asked to the oracle X_i. For a Turing machine M of type $\tau_1 \dots \tau_k$ define

$$L(M) =_{\mathrm{df}} \{(X_1, \dots, X_k) : X_i \text{ is an input of type } \tau_i \text{ and } M \text{ halts on instance } (X_1, \dots, X_k) \text{ with an accepting state}\}$$

as the *language accepted by* M.

A Turing machine M of type $\tau_1 \dots \tau_k$ is called *polynomial-time* (*polynomial-space*, respectively) if there exists a polynomial p such that M halts on every instance $X = (X_1, \dots, X_k)$ in an accepting or rejecting state using no more than $p(|X|) =_{\mathrm{df}} p\left(\sum_{\tau_i=0} |X_i|\right)$ steps (tape cells, respectively). In what follows we assume that X contains at least one word input. Note that the space bound applies also to the length of oracle queries. Now, for $k \geq 1$ and $\tau_1, \dots, \tau_k \in \Phi$ define

$$\begin{aligned} P^{\tau_1 \dots \tau_k} &=_{\mathrm{df}} \{L(M) : M \text{ is a polynomial-time Turing machine of type } \tau_1 \dots \tau_k\} \\ PSPACE^{\tau_1 \dots \tau_k} &=_{\mathrm{df}} \{L(M) : M \text{ is a polynomial-space Turing machine of type } \tau_1 \dots \tau_k\} \end{aligned}$$

which are called *classes of type* $\tau_1 \dots \tau_k$. The classes $P^{\tau_1 \dots \tau_k}$ are the starting point for building up the polynomial-time hierarchies. Next, we define inductively new classes: If $\mathcal{K}$ is a class of type $\tau_1 \dots \tau_k \tau$ then $\exists^\tau \mathcal{K}$ and $\forall^\tau \mathcal{K}$ are classes of type $\tau_1 \dots \tau_k$ which are defined as follows.

For $\tau = 0$:

$L \in \exists^0 \mathcal{K} \Longleftrightarrow_{\mathrm{df}}$ there exist an $L' \in \mathcal{K}$ and a polynomial p, such that

$$(X_1, \dots, X_k) \in L \Longleftrightarrow \exists X \Big(|X| \leq p\big(\sum_{\substack{\tau_i=0 \\ i \leq k}} |X_i|\big) \wedge (X_1, \dots, X_k, X) \in L' \Big)$$

$L \in \forall^0 \mathcal{K} \Longleftrightarrow_{\mathrm{df}}$ there exist an $L' \in \mathcal{K}$ and a polynomial p, such that

$$(X_1, \dots, X_k) \in L \Longleftrightarrow \forall X \Big(|X| \leq p\big(\sum_{\substack{\tau_i=0 \\ i \leq k}} |X_i|\big) \rightarrow (X_1, \dots, X_k, X) \in L' \Big)$$

(Using simple encoding arguments it is easy to see that one can use equivalently "$=$" instead of "$\leq$" in these definitions.)

For $\tau \neq 0$:

$L \in \exists^\tau \mathcal{K} \Longleftrightarrow_{\mathrm{df}}$ there exists an $L' \in \mathcal{K}$, such that

$$(X_1, \dots, X_k) \in L \Longleftrightarrow \exists X \left((X_1, \dots, X_k, X) \in L'\right)$$

$L \in \forall^{\tau}\mathcal{K} \Longleftrightarrow_{\mathrm{df}}$ there exists an $L' \in \mathcal{K}$, such that

$$(X_1, \ldots, X_k) \in L \Longleftrightarrow \forall X\,((X_1, \ldots, X_k, X) \in L')$$

In what follows we also write $\exists^{\sigma}[r]$ instead of $\exists^{\sigma[r]}$, and $\forall^{\sigma}[r]$ instead of $\forall^{\sigma[r]}$.

Obviously, a class of type $\tau_1 \ldots \tau_k$ consists only of languages $L \subseteq \Sigma_1^{(\tau_1)} \times \cdots \times \Sigma_k^{(\tau_k)}$ for suitable finite alphabets $\Sigma_1, \ldots, \Sigma_k$, where we define $\Sigma^{(0)} =_{\mathrm{df}} \Sigma^*$ and $\Sigma^{(\tau)} =_{\mathrm{df}} \mathcal{P}(\Sigma^*)$ in all other cases (without loss of generality we restrict ourselves to the alphabet $\Sigma = \{0, 1\}$). We are particularly interested in the classes of type 0, i.e. in "ordinary" classes of languages. In this case, the superscripts to P are omitted, i.e. for $Q_1, \ldots, Q_k \in \{\exists, \forall\}$ we write $Q_1^{\tau_1} \ldots Q_k^{\tau_k} P$ instead of $Q_1^{\tau_1} \ldots Q_k^{\tau_k} P^{0\tau_1 \ldots \tau_k}$.

Define the classes of the *(arithmetic) polynomial-time hierarchy* as $\Sigma_0^p =_{\mathrm{df}} \Pi_0^p =_{\mathrm{df}} P$ and, for $k \geq 0$, $\Sigma_{k+1}^p =_{\mathrm{df}} \exists^0 \Pi_k^p$ and $\Pi_{k+1}^p =_{\mathrm{df}} \forall^0 \Sigma_k^p$. Furthermore define *PH* as the union of all classes of the polynomial-time hierarchy. It is well-known that each of the classes $Q_1^0 Q_2^0 \ldots Q_k^0 P$ with $Q_i \in \{\exists, \forall\}$ coincides with a class of the polynomial-time hierarchy. For $k \geq 0$, $Q_1, \ldots, Q_k \in \{\exists, \forall\}$, and $\tau_1, \ldots, \tau_k \in \{0, 1, 2\}$ the classes $Q_1^{\tau_1} \ldots Q_k^{\tau_k} P$ are called the classes of the *analytic polynomial-time hierarchy*. The class *APH* is defined as the union of the classes of the analytic polynomial-time hierarchy.

For $k \geq 0$, $Q_1, \ldots, Q_k \in \{\exists, \forall\}$, and $\tau_1, \ldots, \tau_k \in \Phi$ the classes $Q_1^{\tau_1} \ldots Q_k^{\tau_k} P$ are called the classes of the *bounded analytic polynomial-time hierarchy*. Because of

$$Q_1^{\tau_1} \ldots Q_{i-1}^{\tau_{i-1}} Q_i^{\sigma[r]} Q_{i+1}^{\tau_{i+1}} \ldots Q_k^{\tau_k} P \subseteq Q_1^{\tau_1} \ldots Q_{i-1}^{\tau_{i-1}} Q_i^{\sigma} Q_{i+1}^{\tau_{i+1}} \ldots Q_k^{\tau_k} P$$

the union of all classes of the bounded analytic polynomial-time hierarchy is also *APH*.

For $l \geq 1$, let $Q_l =_{\mathrm{df}} \exists$ if l is odd and $Q_l =_{\mathrm{df}} \forall$ otherwise. For $k, m \geq 0$ and $\tau_1, \ldots, \tau_k \in \Phi \setminus \{0\}$ we define $\mathcal{K}(\tau_1\tau_2\tau_3 \ldots \tau_k, m) =_{\mathrm{df}} \exists^{\tau_1}\forall^{\tau_2}\exists^{\tau_3} \ldots Q_k^{\tau_k} Q_{k+1}^0 \ldots Q_{k+m}^0 P$. Furthermore, for $\tau \in \Phi$ we write τ^k instead of $\underbrace{\tau \ldots \tau}_{k \text{ times}}$.

For $k \geq 0$, $Q_1, \ldots, Q_k \in \{\exists, \forall\}$, $\tau_1, \ldots, \tau_k \in \Phi$, let $\tau(Q_1^{\tau_1} \ldots Q_k^{\tau_k}) =_{\mathrm{df}} \tau_1 \ldots \tau_k$ be the type of the operator (or quantifier) string $Q_1^{\tau_1} \ldots Q_k^{\tau_k}$. Furthermore, for $Q = Q_1^{\tau_1} \ldots Q_k^{\tau_k}$ and $X = (X_1, \ldots, X_k)$ we write QX instead of $Q_1^{\tau_1} X_1 \ldots Q_k^{\tau_k} X_k$. We define $\overline{\exists} =_{\mathrm{df}} \forall$ and $\overline{\forall} =_{\mathrm{df}} \exists$, and for $Q = Q_1^{\tau_1} \ldots Q_k^{\tau_k}$ we define $\overline{Q} =_{\mathrm{df}} \overline{Q}_1^{\tau_1} \ldots \overline{Q}_k^{\tau_k}$. For a class $\mathcal{K}$ of languages, define $\mathrm{co}\mathcal{K} =_{\mathrm{df}} \left\{\overline{L} : L \in \mathcal{K}\right\}$. Obviously, we have $\mathrm{co}QP^{\mu\tau(Q)} = \overline{Q}P^{\mu\tau(Q)}$ for all $\mu \in \Phi^*$.

Finally, some more notations are given to help us in the proofs of the results: For $a \in \{0, 1\}$ and $u \in \{0, 1\}^*$, define the encoding $\widehat{ua} =_{\mathrm{df}} \widehat{u}aa$ ($\widehat{\epsilon} =_{\mathrm{df}} \epsilon$). For a set $U \subseteq \{0, 1\}^*$, let c_U be the characteristic function of U. With every set $U \subseteq \{0, 1\}^*$ and every $m \in \mathbb{N}$ we associate the word $\langle U, m\rangle =_{\mathrm{df}} c_U(1)c_U(1^2)c_U(1^3) \ldots c_U(1^m)$. For $u \in \{0, 1\}^*$ we denote the i-th bit of u by $u(i)$.

3 Results

The reader is assumed to be familiar with the most popular complexity classes like *P*, *NP*, Σ_k^p, Π_k^p ($k \geq 0$), and *PSPACE*. The classes $\Sigma_k^{\exp}$ and $\Pi_k^{\exp}$ ($k \geq 1$) of the exponential-time hierarchy are defined by exponential time alternating machines which start with an existential, or universal respectively, state and having at most $k - 1$ alternations.

In [4] a complete characterization of the classes of the analytic polynomial-time hierarchy is given by showing that every such class coincides with one of the classes Σ_k^p, Π_k^p ($k \geq 0$), *PSPACE*, Σ_k^{exp} or Π_k^{exp} ($k \geq 1$) and vice versa. This is shown in [4] in the following two steps:

1. Every class of the analytic polynomial-time hierarchy coincides with one of the classes *P*, $\mathcal{K}(\sigma, m)$ or co$\mathcal{K}(\sigma, m)$, where $\sigma \in \{1, 2\}^*$ and $m \geq 1$.

2. Every class $\mathcal{K}(\sigma_1\sigma_2 \ldots \sigma_k, m)$ such that $k \geq 0$, $m \geq 1$ and $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2\}$ can be characterized as follows:

   - For $k = 0$ and $m \geq 1$ we have $\mathcal{K}(\cdot, m) = \Sigma_m^p$.
   - For $k > 0$ we have

$m = 1$	$k = 1$	$\mathcal{K}(1, 1) =$ *PSPACE*
		$\mathcal{K}(2, 1) =$ *NEXPTIME*
	$k \geq 2$	$\mathcal{K}(\sigma_1 \ldots \sigma_{k-1}1, 1) = \Sigma_{k-1}^{exp}$
		$\mathcal{K}(\sigma_1 \ldots \sigma_{k-1}2, 1) = \Sigma_k^{exp}$
$m \geq 2$	$k \geq 1$	$\mathcal{K}(\sigma_1 \ldots \sigma_{k-1}\sigma_k, m) = \Sigma_k^{exp}$

Note that here and in what follows we will state and prove only the results for $\mathcal{K}(\cdot)$. The results for co$\mathcal{K}(\cdot)$ are immediate consequences.

In this paper we concentrate on the second step for the classes of the bounded analytic polynomial-time hierarchy, i.e. we will characterize all the classes of the bounded analytic polynomial-time hierarchy which have the form $\mathcal{K}(\tau_1\tau_2 \ldots \tau_k, m)$, where $k, m \geq 1$ and $\tau_1, \tau_2, \ldots, \tau_k \in \{1, 2\} \cup \{\sigma[r] : \sigma \in \{1, 2\}, r : \mathbb{N} \to \mathbb{N}^+\}$. It will turn out that each of these classes coincides with one of the classes *NP*, co*NP*, *PSPACE*, Σ_k^{exp} or Π_k^{exp} ($k \geq 1$) and vice versa.

The following table includes the corresponding results for the classes $\mathcal{K}(\tau_1\tau_2 \ldots \tau_k, m)$, where $k, m \geq 1$ and $\tau_1, \tau_2, \ldots, \tau_k \in \{\sigma[r] : \sigma \in \{1, 2\}, r : \mathbb{N} \to \mathbb{N}^+\}$, i.e. for the cases where there is a restriction to the number of queries for every set quantifier. The last column refers to the Theorem(s) which states the corresponding result.

$m = 1$	$r_k = 1$	$k = 1$	$\mathcal{K}(\sigma_1[1], 1) =$ co*NP*	9, 10
		$k = 2$	$\mathcal{K}(\sigma_1[r_1]\,\sigma_2[1], 1) =$ *NP*	12
		$k = 3$	$\mathcal{K}(\sigma_1[1]\,\sigma_2[r_2]\,\sigma_3[1], 1) =$ *PSPACE*	12
			$\mathcal{K}(\sigma_1[r_1]\,\sigma_2[r_2]\,\sigma_3[1], 1) =$ *NEXPTIME* $(r_1 \geq 2)$	12
		$k \geq 4$	$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,\sigma_k[1], 1) = \Sigma_{k-3}^{exp}$ $(r_{k-2} = 1)$	12
			$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,\sigma_k[1], 1) = \Sigma_{k-2}^{exp}$ $(r_{k-2} \geq 2)$	12
	$r_k = 2$	$k = 1$	$\mathcal{K}(\sigma_1[2], 1) =$ *PSPACE*	9, 10
		$k \geq 2$	$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,\sigma_k[2], 1) = \Sigma_{k-1}^{exp}$	11
	$r_k \geq 3$	$k = 1$	$\mathcal{K}(1[r_1], 1) =$ *PSPACE*	9
			$\mathcal{K}(2[r_1], 1) =$ *NEXPTIME*	10
		$k \geq 2$	$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,1[r_k], 1) = \Sigma_{k-1}^{exp}$	11
			$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,2[r_k], 1) = \Sigma_k^{exp}$	11
$m \geq 2$	$r_k \geq 1$	$k \geq 1$	$\mathcal{K}(\sigma_1[r_1] \ldots \sigma_{k-1}[r_{k-1}]\,\sigma_k[r_k], m) = \Sigma_k^{exp}$	8

Obviously we have in all cases

$$\mathcal{K}(\tau_1\tau_2 \ldots \tau_{i-1}\sigma_i[r_i]\,\tau_{i+1} \ldots \tau_k, m) = \mathcal{K}(\tau_1\tau_2 \ldots \tau_{i-1}\sigma_i[3]\,\tau_{i+1} \ldots \tau_k, m)$$

for $\sigma_i = 1,2$ and $r_i \geq 3$. We can chose r_i so large that it is not a real restriction. Hence we have also

$$\mathcal{K}(\tau_1\tau_2\ldots\tau_{i-1}\sigma_i\tau_{i+1}\ldots\tau_k, m) = \mathcal{K}(\tau_1\tau_2\ldots\tau_{i-1}\sigma_i[3]\,\tau_{i+1}\ldots\tau_k, m)$$

for $\sigma_i = 1,2$ and $\tau_1,\tau_2,\ldots,\tau_{i-1},\tau_{i+1}\ldots,\tau_k \in \{\sigma[r] : \sigma \in \{1,2\}, r : \mathbb{N} \to \mathbb{N}^+\}$. Iterating this step we obtain results for all $\mathcal{K}(\tau_1\tau_2\ldots\tau_k, m)$, where $k, m \geq 1$ and $\tau_1,\tau_2,\ldots,\tau_k \in \{1,2\} \cup \{\sigma[r] : \sigma \in \{1,2\}, r : \mathbb{N} \to \mathbb{N}^+\}$. In particular, we obtain the results of [4] (see the first table) as special cases.

Finally, let us point out how heavily (and nicely) the results can depend on the number of queries allowed. A good example is the following:

$$\begin{aligned}
\mathcal{K}(2[1], 1) &= \mathrm{co}\mathit{NP} \\
\mathcal{K}(2[2], 1) &= \mathit{PSPACE} \\
\mathcal{K}(2[3], 1) &= \mathit{NEXPTIME}
\end{aligned}$$

4 Inclusion Rules

To establish the above mentioned results *inclusion rules* are very helpful. For $R, S \in \Gamma^*$, the inclusion rule $R \to S$ is valid if the replacement of the operator string R by the string S in any context does not diminish the class in question, i.e. $RQP^{\mu\tau(R)\tau(Q)} \subseteq SQP^{\mu\tau(S)\tau(Q)}$ for all $Q \in \Gamma^*$ and $\mu \in \Phi^*$. We say that the *equivalence rule* $R \leftrightarrow S$ is valid if the replacement of the operator string R by the string S in any context does not change the class in question, i.e. $RQP^{\mu\tau(R)\tau(Q)} = SQP^{\mu\tau(S)\tau(Q)}$ for all $Q \in \Gamma^*$ and $\mu \in \Phi^*$. Obviously, we have $R \leftrightarrow S$ if and only if $R \to S$ and $S \to R$.

Throughout this paper, a rule $R \to S$ and "its complement" $\overline{R} \to \overline{S}$ have to be proved. Obviously we have to prove only one of them.

Proposition 1 (Complementation). [4] *Let* $R, S \in \Gamma^*$. *If* $R \to S$ *then* $\overline{R} \to \overline{S}$.

In [4] some rules were proved. In our new context we add as necessary the terms $[r]$. Some of the rules show relations between the existential (universal, respectively) quantifiers of different types. They can be restated as follows.

Lemma 2. [4] *Let* $\sigma \in \{1,2\}$ *and* $r : \mathbb{N} \to \mathbb{N}$. *Then*

(1) $\epsilon \to \exists^0$ *and* $\epsilon \to \forall^0$

(2) $\epsilon \leftrightarrow \exists^\sigma[0]$ *and* $\epsilon \leftrightarrow \forall^\sigma[0]$

(3) $\exists^1[r] \to \exists^2[r]$ *and* $\forall^1[r] \to \forall^2[r]$

Next, we restate some rules showing how to melt neighboured existential (universal, respectively) quantifiers.

Lemma 3. [4] *Let* $\sigma \in \{1,2\}$ *and* $r, r' : \mathbb{N} \to \mathbb{N}$. *Then*

(1) $\exists^0\exists^0 \leftrightarrow \exists^0$ *and* $\forall^0\forall^0 \leftrightarrow \forall^0$

(2) $\exists^\sigma\,[r]\,\exists^0 \rightarrow \exists^\sigma$ *and* $\forall^\sigma\,[r]\,\forall^0 \rightarrow \forall^\sigma$

If we follow the proof of the rule $\exists^2 \rightarrow \exists^1\exists^1\forall^0$ in [4] we obtain $\exists^2\,[r] \rightarrow \exists^1\,[1]\,\exists^1\,[r+1]\,\forall^0$. However, in this proof we can make some refinements and obtain a similar rule requiring less queries in the second set operator. This is shown in the following lemma.

Lemma 4. *Let* $\sigma \in \{1,2\}$ *and* $r : \mathbb{N} \rightarrow \mathbb{N}^+$. *Then*

$$\exists^2\,[r] \rightarrow \exists^1\,[1]\,\exists^1\,[r-1]\,\forall^0 \quad \textit{and} \quad \forall^2\,[r] \rightarrow \forall^1\,[1]\,\forall^1\,[r-1]\,\exists^0$$

Proof. We prove the first statement, since the second follows by complementation. For $r = 1$ is evident. For $r \geq 2$ let $Q \in \Gamma^*$ and $\mu \in \Phi^*$, and let $L \in \exists^2\,[r]\,QP^{\mu 2[r]\tau(Q)}$. There exists an $L_1 \in P^{\mu 2[r]\tau(Q)}$ such that for a suitable polynomial p

$$\begin{aligned}
X \in L \Longleftrightarrow\; & \exists^2 U\, QY\,((X,U,Y) \in L_1) \\
\Longleftrightarrow\; & \exists^2 U\, \exists^1 W \Big(\forall^0 u\, \forall^0 v \Big(|u|\,,|v| \leq p\,(|X|) \rightarrow (u \in U \leftrightarrow v01\widehat{u} \in W)\Big) \\
& \qquad\qquad \wedge\, QY\Big((X,U,Y) \in L\,(M)\Big)\Big)
\end{aligned}$$

$\big($take $W = \{v01\widehat{u} : u, v \in \{0,1\}^* \wedge u \in U\}$ for example, and let M be a polynomial-time machine of type $\mu 2\,[r]\,\tau\,(Q)$ accepting $L_1\big)$

$$\begin{aligned}
\Longleftrightarrow\; & \exists^1 U\, \exists^1 W\, \forall^0 a\, \forall^0 u\, \forall^0 v\, QY \Big(|u|\,,|v| \leq p\,(|X|) \rightarrow \\
& \qquad \Big((a = 0 \rightarrow (u \in U \leftrightarrow v01\widehat{u} \in W)) \wedge \\
& \qquad (a = 1 \rightarrow (X,U,W,Y) \in L\,(M'))\Big)\Big)
\end{aligned}$$

$\big(M'$ works on input (X,U,W,Y) as M on input (X,U,Y) but it only asks the first query to U like M does. Instead of asking u to U after queries $u_1,\ldots,u_m$ ($m \geq 1$) it asks $01\widehat{u}_1 01\widehat{u}_2 01\ldots01\widehat{u}_m 01\widehat{u}$ to W. Note that for every (X,a,u,v,Y), oracle U is asked exactly once and oracle W is asked at most $r-1$ times in a type 1 manner.)

This shows $L \in \exists^1\,[1]\,\exists^1\,[r-1]\,\forall^0\forall^0\forall^0 QP^{\mu 1[1]1[r-1]000\tau(Q)}$ and using the rules of Lemma 3, $L \in \exists^1\,[1]\,\exists^1\,[r-1]\,\forall^0 QP^{\mu 1[1]1[r-1]0\tau(Q)}$. ❑

Finally we restate from [4] an "equivalence rule" which is valid only in a special context. It says that a set quantifier without query restriction is exactly as powerful as the corresponding word quantifier when applied to *P*.

Lemma 5. [4] *Let* $\mu \in \Phi^*$. *Then*

$$\exists^0 P^{\mu 0} = \exists^1 P^{\mu 1} = \exists^2 P^{\mu 2} \quad \textit{and} \quad \forall^0 P^{\mu 0} = \forall^1 P^{\mu 1} = \forall^2 P^{\mu 2}$$

5 Characterizing the classes $\mathcal{K}(\cdot)$ and co$\mathcal{K}(\cdot)$

In this section, we characterize the classes having the form $\mathcal{K}(\sigma_1[r_1]\,\sigma_2[r_2]\ldots\sigma_k[r_k], m)$ or co$\mathcal{K}(\sigma_1[r_1]\,\sigma_2[r_2]\ldots\sigma_k[r_k], m)$ by well-known complexity classes for $k, m \geq 1$, $\sigma_1, \ldots, \sigma_k \in \{1, 2\}$ and $r_1, \ldots, r_k : \mathbb{N} \to \mathbb{N}^+$. We start with translating results from [4] in the language of the bounded analytic polynomial-time hierarchy. A result there states which classes of the analytic polynomial-time hierarchy contain a level of the exponential-time hierarchy. In the proof of this result some remarks on the number of queries to the oracles were made. This can be restated as follows:

Theorem 6. [4] *For* $k \geq 1$ *we have*

$$\Sigma_k^{\exp} \subseteq \mathcal{K}\left((1\,[1])^k, 2\right) \cap \mathcal{K}\left((1\,[1])^{k-1}2\,[3], 1\right)$$

Another theorem in [4] shows which classes of the form $\mathcal{K}(\cdot)$ are included in a level of the exponential-time hierarchy.

Theorem 7. [4] *For* $k, m \geq 1$ *we have*

$$\mathcal{K}\left(2^k, m\right) \cup \mathcal{K}\left(2^k 1, 1\right) \subseteq \Sigma_k^{\exp}$$

As a direct consequence of these theorems we obtain that each class containing at least one set operator and at least two word operators coincides with one class of the exponential-time hierarchy.

Theorem 8. *Let* $k \geq 1$, $m \geq 2$, $\sigma_1, \ldots, \sigma_k \in \{1, 2\}$ *and* $r_1, \ldots, r_k : \mathbb{N} \to \mathbb{N}^+$. *Then*

$$\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k], m) = \Sigma_k^{\exp}$$

Thus, remains to consider the classes $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k], 1)$ and co$\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k], 1)$ with $k \geq 1$. The simplest classes of these types are those containing only one set operator of type 1 which turn out to coincide with *NP*, co*NP* or *PSPACE*.

Theorem 9. *Let* $r : \mathbb{N} \to \mathbb{N}$. *Then*

(1) $\mathcal{K}(1\,[1], 1) =$ co*NP*

(2) $\mathcal{K}(1\,[r], 1) =$ *PSPACE for* $r \geq 2$

Proof. Statement (1): The proof of co$NP = \forall^0 P \subseteq \exists^1[1]\,\forall^0 P = \mathcal{K}(1\,[1], 1)$ is evident. Next we prove $\exists^1[1]\,\forall^0 P \subseteq \forall^0 P$. Let $L \in \exists^1[1]\,\forall^0 P$. By definition there exist an $L_1 \in P^{01[1]0}$ and a polynomial p such that

$$x \in L \iff \exists^1 U\, \forall^0 v\, (|v| = p(|x|) \to (x, U, v) \in L_1)$$

where M is a polynomial-time machine of type $01\,[1]\,0$ accepting L_1 is such a way that the oracle U is queried only once. Let f be a function computable in polynomial-time such that the machine

M on input (x, U, v) queries $f(x, v)$ to oracle U. Thus, there exist an $L_2 \in P^{000}$ and a polynomial q such that

$$\begin{aligned} x \in L &\iff \exists^1 U\, \forall^0 v\, (|v| = p\,(|x|) \rightarrow (x, v, c_U(f(x, v))) \in L_2) \\ &\iff \forall^0 u\, \forall^0 v\, \forall^0 w \Big(|u| \le q\,(|x|) \wedge |v| = |w| = p\,(|x|) \rightarrow \Big((f(x, v) = u \rightarrow (x, v, 1) \in L_2) \\ &\qquad\qquad \vee (f(x, w) = u \rightarrow (x, w, 0) \in L_2) \Big)\Big) \end{aligned}$$

Therefore, $L \in \forall^0\forall^0\forall^0 P$. Now, using Lemma 3 we get the desired result.
Statement (2): The inclusion $\exists^1\, [2]\, \forall^0 P \subseteq \exists^1\forall^0 P$ is evident and $\exists^1\forall^0 P = PSPACE$ follows by [4]. Thus, only $PSPACE \subseteq \exists^1\, [2]\, \forall^0 P$ has to be proved. We use Cai and Furst's characterization of *PSPACE* [6]. Let A_5 be the group of even permutations on $[5] =_{\mathrm{df}} \{0, \ldots, 4\}$, and let $\circ$ be the multiplication of this group. For $u \in \{0, 1\}^n$, let $\overleftarrow{u} \in \{0, 1\}^n$ denote the predecessor of u in $\{0, 1\}^n$ in lexicographical order. Cai and Furst proved that for every language $L \in PSPACE$ there exist a function $f : \{0, 1\}^* \times \{0, 1\}^* \rightarrow A_5$ computable in polynomial time and a polynomial p such that

$$x \in L \iff f(x, 1^{p(|x|)}) \circ f(x, 1^{p(|x|)-1}0) \circ \cdots \circ f(x, 0^{p(|x|)})(1) = 1$$

Consequently,

$$\begin{aligned} x \in L \iff &\exists g \Big((g : \{0, 1\}^* \rightarrow [5]) \wedge \forall u \Big(|u| = p\,(|x|) \rightarrow \\ &\qquad \Big(\Big(u = 0^{p(|x|)} \rightarrow f\,(x, u)\,(1) = g\,(u) \Big) \\ &\qquad \wedge \Big(u \notin \big\{0^{p(|x|)}, 1^{p(|x|)}\big\} \rightarrow f\,(x, u)\,(g(\overleftarrow{u})) = g\,(u) \Big) \\ &\qquad \wedge \Big(u = 1^{p(|x|)} \rightarrow f\,(x, u)\,(g(\overleftarrow{u})) = 1 \Big) \Big)\Big)\Big) \\ \iff &\exists^2 U\, \forall^0 u\, \forall^0 i\, \forall^0 j \Big(|u| = p\,(|x|) \wedge 0 \le i, j \le 4 \rightarrow \\ &\qquad \Big(\Big(u = 0^{p(|x|)} \rightarrow (f\,(x, u)\,(1) = i \leftrightarrow u0^i \in U) \Big) \\ &\qquad \wedge \Big(u \notin \big\{0^{p(|x|)}, 1^{p(|x|)}\big\} \wedge \overleftarrow{u}0^i \in U \rightarrow (f\,(x, u)\,(i) = j \leftrightarrow u0^j \in U) \Big) \\ &\qquad \wedge \Big(u = 1^{p(|x|)} \wedge \overleftarrow{u}0^i \in U \rightarrow f\,(x, u)\,(i) = 1 \Big) \Big)\Big) \end{aligned}$$

However, for each (x, U, u, i, j) such that $u \notin \{0^{p(|x|)}, 1^{p(|x|)}\}$ the two queries $\overleftarrow{u}0^i$ and $u0^j$ are asked to U which is not a type 1 querying. To overcome this difficulty we encode the words from $(\{0, 1\}^{p(|x|)} \setminus \{1^{p(|x|)}\}) \times \{0\}^{\le 4}$ by an injective function α which has the property that, for every $u \notin \{0^{p(|x|)}, 1^{p(|x|)}\}$ and $0 \le i, j \le 4$, either $\alpha(\overleftarrow{u}, 0^i)$ is an initial word of $\alpha(u, 0^j)$ or vice versa. (For example take $\alpha(a_1a_2 \ldots a_m01^{p(|x|)-m-1}, 0^i) =_{\mathrm{df}} 0^4a_10^4a_2 \ldots 0^4a_m0^i$ where $m \ge 0$ and $a_1, a_2, \ldots, a_m \in \{0, 1\}$.) Now instead of querying $\overleftarrow{u}0^i$ and $u0^j$ to U the queries $\alpha(\overleftarrow{u}, 0^i)$ and $\alpha(u, 0^j)$ are made to the oracle $V =_{\mathrm{df}} \{\alpha(u, 0^i) : 0 \le i \le 4 \wedge u0^i \in U\}$ in a type 1 manner.

Therefore,

$$x \in L \Longleftrightarrow \exists^1 V \forall^0 u \forall^0 i \forall^0 j \Big((|u| = p(|x|) \wedge 0 \le i, j \le 4) \to (x, V, u, i, j) \in L_1 \Big)$$

where $(x, V, u, i, j) \in L_1 \Longleftrightarrow_{\mathrm{df}}$

$$\begin{aligned}
&\Big(u = 0^{p(|x|)} \to (f(x,u)(1) = i \leftrightarrow \alpha(u, 0^i) \in V) \Big) \\
\wedge &\Big(u \notin \{0^{p(|x|)}, 1^{p(|x|)}\} \wedge \alpha(\overleftrightarrow{u}, 0^i) \in V \to (f(x,u)(i) = j \leftrightarrow \alpha(u, 0^j) \in V) \Big) \\
\wedge &\Big(u = 1^{p(|x|)} \wedge \alpha(\overleftrightarrow{u}, 0^i) \in V \to f(x,u)(i) = 1 \Big)
\end{aligned}$$

Obviously, L_1 can be accepted by a deterministic polynomial-time oracle Turing machine which, on input (x, V, u, i, j), queries the oracle V in type 1 manner and at most two times. Hence, $L_1 \in P^{01[2]000}$ and $L \in \exists^1 [2] \forall^0 \forall^0 \forall^0 P$. Now, using Lemma 3 we conclude $L \in \exists^1 [2] \forall^0 P$. ❑

The next result shows that for $\mathcal{K}(\sigma[r], 1)$ a single type 2 operator is probably more powerful than a type 1 operator when more than two queries are allowed. For $a \in \{0, 1\}$ let $a^1 =_{\mathrm{df}} a$ and $a^0 =_{\mathrm{df}} 1 - a$, and for a set U let $U^1 =_{\mathrm{df}} U$ and $U^0 =_{\mathrm{df}} \overline{U}$.

Theorem 10. *Let* $r : \mathbb{N} \to \mathbb{N}$. *Then*

(1) $\mathcal{K}(2[1], 1) = \mathrm{co}NP$

(2) $\mathcal{K}(2[2], 1) = PSPACE$

(3) $\mathcal{K}(2[r], 1) = NEXPTIME$ *for* $r \ge 3$

Proof. The first statement follows by Theorem 9 and the last statement follows by Theorems 6 and 7. Next we prove the Statement (2). By Theorem 9 and Lemma 2 follows $PSPACE \subseteq \mathcal{K}(1[2], 1) \subseteq \mathcal{K}(2[2], 1)$. Next we prove $\mathcal{K}(2[2], 1) \subseteq PSPACE$. For $L \in \exists^2 [2] \forall^0 P$ there exist polynomial-time computable functions f, g_1, g_2 and polynomial p such that f is a 0-1-function ($g_i \approx$ the i-th query to oracle) and

$$\begin{aligned}
x \in L &\Leftrightarrow \exists U \forall u \Big(|u| = p(|x|) \to \bigvee_{\substack{a,b \in \{0,1\} \\ f(x,u,a,b)=1}} (g_1(x,u) \in U^a \wedge g_2(x,u,a) \in U^b) \Big) \\
&\Leftrightarrow \text{the boolean formula } \bigwedge_{|u|=p(|x|)} \bigvee_{\substack{a,b \in \{0,1\} \\ f(x,u,a,b)=1}} (z^a_{g_1(x,u)} \wedge z^b_{g_2(x,u,a)}) \text{ with the variables} \\
&\quad\ z_0, \ldots, z_{2^{q(|x|)}} \text{ is satisfiable } (q \text{ suitable polynomial})
\end{aligned}$$

A simple computation shows that the formula $\bigvee_{\substack{a,b \in \{0,1\} \\ f(x,u,a,b)=1}} (z^a_{g_1(x,u)} \wedge z^b_{g_2(x,u,a)})$ is equivalent to a conjunction of clauses with at most two literals each, though the formula has three variables. (An easier way to see that is to use Lemma 13 which shows that we can without loss of generality assume that the queries are made in parallel. In this case the formula has two variables at all and hence its conjunctive normal form has only clauses with at most two literals.)

Hence, L can be m-reduced to the 2-SAT problem using polynomial space. Because 2-SAT $\in NL$ (see [9]) we obtain $L \in PSPACE$. ❑

Thus, we have characterized the classes of the bounded analytic polynomial-time hierarchy having the form $\mathcal{K}(\sigma[r],1)$ and $\mathrm{co}\mathcal{K}(\sigma[r],1)$. To reach our goal in this section it remains to consider the classes containing at least two set operators followed by one word operator. This will be done in two steps. First we consider the classes where the last set operator is not restricted to one query.

Theorem 11. *Let* $k \geq 1$, $\sigma_1, \ldots, \sigma_k, \sigma \in \{1,2\}$ *and* $r_1, \ldots, r_k, r : \mathbb{N} \to \mathbb{N}^+$. *Then*

(1) $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,\sigma[2],1) = \Sigma_k^{\exp}$

(2) $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,1[r],1) = \Sigma_k^{\exp}$ *for* $r \geq 3$

(3) $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,2[r],1) = \Sigma_{k+1}^{\exp}$ *for* $r \geq 3$

Proof. The last statement follows by Theorems 6 and 7. For the second statement, we conclude

$$\begin{aligned}
\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,1[r],1) &\subseteq \mathcal{K}\left(2^k 1,1\right) \subseteq \Sigma_k^{\exp} && \text{by Theorem 7}\\
&\subseteq \mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,1[2],1) && \text{by Statement (1)}\\
&\subseteq \mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,1[r],1)
\end{aligned}$$

Now, the first statement is proved. It is sufficient to show $\Sigma_k^{\exp} \subseteq \mathcal{K}\left((1[1])^k 1[2],1\right)$ and $\mathcal{K}\left(2^k 2[2],1\right) \subseteq \Sigma_k^{\exp}$.
"$\Sigma_k^{\exp} \subseteq \mathcal{K}\left((1[1])^k 1[2],1\right)$": By Theorem 6 we obtain $\Sigma_k^{\exp} \subseteq \mathcal{K}\left((1[1])^k,2\right)$. Hence, it is easy to see that we have to prove only $\exists^0\forall^0 \mathbf{P}^{0(1[1])^k 00} \subseteq \exists^1[2]\forall^0\mathbf{P}^{0(1[1])^k 1[2]0}$. For a language $L \in \exists^0\forall^0\mathbf{P}^{0(1[1])^k 00}$ there exist an $L_1 \in \mathbf{P}^{0(1[1])^k 00}$ and polynomials p and q such that

$$X \in L \iff \exists^0 u\,\forall^0 w\,(|u| = p(|X|) \wedge (|w| = q(|X|) \to (X,u,w) \in L_1))$$

For $a \in \{0,1,2\}$, $u \in \{0,1\}^{p(|X|)}$ and $w \in \{0,1\}^{q(|X|)}$ define the function f as follows

$$f(X,uw)(a) =_{\mathrm{df}} \begin{cases} 1, & \text{if } a = 1\\ 2, & \text{if } a = 0 \text{ and } w = 0^{q(|X|)} \text{ and } (X,u,w) \in L_1\\ 2, & \text{if } a = 2 \text{ and } w \neq 1^{q(|X|)} \text{ and } (X,u,w) \in L_1\\ 1, & \text{if } a = 2 \text{ and } w = 1^{q(|X|)} \text{ and } (X,u,w) \in L_1\\ 0, & \text{otherwise}\end{cases}$$

Obviously, f is computable in polynomial time and

$$X \in L \iff f(X,1^{p(|X|)+q(|X|)}) \circ f(X,1^{p(|X|)+q(|X|)-1}0) \circ \cdots \circ f(X,0^{p(|X|)+q(|X|)})(0) = 1$$

where $\circ$ is the traditional composition of functions. Now, in the same way we prove $\mathit{PSPACE} \subseteq \exists^1[2]\forall^0\mathbf{P}$ in Theorem 9 follows the inclusion $\exists^0\forall^0\mathbf{P}^{0(1[1])^k 00} \subseteq \exists^1[2]\forall^0\mathbf{P}^{0(1[1])^k 1[2]0}$.
"$\mathcal{K}\left(2^k 2[2],1\right) \subseteq \Sigma_k^{\exp}$": Let k be even. Note that the proof of $\mathcal{K}(2[2],1) \subseteq \mathit{PSPACE}$ (Theorem 10) remains valid if the machines have additionally k oracles of type 2, i.e. $\exists^2\forall^2\ldots\forall^2\exists^2[2]\forall^0\mathbf{P} \subseteq \exists^2\forall^2\ldots\forall^2\mathit{PSPACE}^{02^k}$. However, $\exists^2\forall^2\ldots\forall^2\mathit{PSPACE}^{02^k} \subseteq \Sigma_k^{\exp}$ since an alternating exponential-time machine with $k-1$ alternations which stars with an existential state can accept a language from $\exists^2\forall^2\ldots\forall^2\mathit{PSPACE}^{02^k}$.

The case of odd k is treated analogously on the base of $\mathrm{co}\mathcal{K}(2[2],1) \subseteq \mathit{PSPACE}$. ❑

Now we consider the case at least two set operators where the last one is restricted to one query.

Theorem 12. *Let* $k \geq 1$, $\sigma_1, \ldots, \sigma_k, \sigma, \tau \in \{1, 2\}$ *and* $r_1, \ldots, r_k, r : \mathbb{N} \to \mathbb{N}^+$. *Then*

(1) $\mathcal{K}(\sigma[r]\,\tau[1], 1) = NP$

(2) $\mathcal{K}(\sigma_1[1]\,\sigma[r]\,\tau[1], 1) = PSPACE$

(3) $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_{k-1}[r_{k-1}]\,\sigma_k[1]\,\sigma[r]\,\tau[1], 1) = \Sigma^{\exp}_{k-1}$ *for* $k \geq 2$

(4) $\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,\sigma[r]\,\tau[1], 1) = \Sigma^{\exp}_{k}$ *for* $r_k \geq 2$

Proof. For $l \geq 1$, let $Q_l =_{\mathrm{df}} \exists$ if l is odd and $Q_l =_{\mathrm{df}} \forall$ otherwise. In the proof of $\exists^\tau[1]\,\forall^0 P \subseteq \forall^0 P$ (Theorems 9 and 10) we need to simulate only two times the original machine for each word uvw. Therefore, the number of queries to every oracle is doubled and the type 1 property cannot be guaranteed. Thus (if k is even, take the base $\forall^\tau[1]\,\exists^0 P \subseteq \exists^0 P$)

$$\begin{aligned}\exists^{\sigma_1}[r_1]\ldots Q_k^{\sigma_k}[r_k]\,Q_{k+1}^{\sigma}[r]\,Q_{k+2}^{\tau}[1]\,Q_{k+3}^{0}P &\subseteq \exists^2[2r_1]\ldots Q_k^2[2r_k]\,Q_{k+1}^2[2r]\,Q_{k+3}^0 P \\ &\subseteq \exists^2[2r_1]\ldots Q_k^2[2r_k]\,Q_{k+1}^0 P\end{aligned} \tag{1}$$

since $Q_{k+1} = Q_{k+3}$ and by Lemmas 3 and 5 we can melt these last 2 operators to one operator of type 0. Hence we conclude
Statement (1): by Equation (1)

$$\mathcal{K}(\sigma[r]\,\tau[1], 1) \subseteq \mathcal{K}(\epsilon, 1) \subseteq NP \subseteq \mathcal{K}(\sigma[r]\,\tau[1], 1)$$

Statement (2):

$$\begin{aligned}\mathcal{K}(\sigma_1[1]\,\sigma[r]\,\tau[1], 1) &\subseteq \mathcal{K}(2[2], 1) && \text{by Equation (1)} \\ &\subseteq PSPACE \subseteq \mathcal{K}(1[2], 1) && \text{by Theorems 10 and 9} \\ &\subseteq \mathcal{K}\big((1[1])^3, 1\big) && \text{by Lemmas 4 and 2} \\ &\subseteq \mathcal{K}(\sigma_1[1]\,\sigma[r]\,\tau[1], 1)\end{aligned}$$

Statement (3):

$$\begin{aligned}&\mathcal{K}(\sigma_1[r_1]\ldots\sigma_{k-1}[r_{k-1}]\,\sigma_k[1]\,\sigma[r]\,\tau[1], 1) \\ &\quad\subseteq \mathcal{K}(2[2r_1]\ldots 2[2r_{k-1}]\,2[2], 1) && \text{by Equation (1)} \\ &\quad\subseteq \Sigma^{\exp}_{k-1} \subseteq \mathcal{K}\big((1[1])^{k-1}1[2], 1\big) && \text{by Theorem 11} \\ &\quad\subseteq \mathcal{K}\big((1[1])^{k+2}, 1\big) && \text{by Lemmas 4 and 2} \\ &\quad\subseteq \mathcal{K}(\sigma_1[r_1]\ldots\sigma_{k-1}[r_{k-1}]\,\sigma_k[1]\,\sigma[r]\,\tau[1], 1)\end{aligned}$$

Statement (4):

$$\begin{aligned}\mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,\sigma[r]\,\tau[1], 1) &\subseteq \mathcal{K}(2[2r_1]\ldots 2[2r_k], 1) && \text{by Equation (1)} \\ &\subseteq \Sigma^{\exp}_{k} \subseteq \mathcal{K}\big((1[1])^{k-1}2[3], 1\big) && \text{by Theorems 7 and 6} \\ &\subseteq \mathcal{K}\big((1[1])^{k-1}1[2]\,1[1]\,1[1], 1\big) && \text{by Lemmas 4 and 2} \\ &\subseteq \mathcal{K}(\sigma_1[r_1]\ldots\sigma_k[r_k]\,\sigma[r]\,\tau[1], 1)\end{aligned}$$

❑

6 Parallel Queries

In this section we consider oracle machines that during their computations ask the queries in a parallel manner, i.e. they compute a list of all queries before asking one of them. We will represent the parallel restriction over an oracle bounded by set operator $Q^{\sigma}[r]$ by $Q^{\sigma}[\| r]$, where $Q \in \{\exists, \forall\}$, $\sigma \in \{1, 2\}$ and $r : \mathbb{N} \to \mathbb{N}$. Next, we show equivalences between classes involving and not involving parallel queries.

Lemma 13. *Let* $\mu \in \Phi^*$. *Then, for* $k \in \mathbb{N}$ *and* $\sigma \in \{1, 2\}$

$$\exists^{\sigma}[\| k]\, \forall^0 \boldsymbol{P}^{\mu\sigma[\| k]0} = \exists^{\sigma}[k]\, \forall^0 \boldsymbol{P}^{\mu\sigma[k]0}$$

Proof. The direction "$\subseteq$" is evident. For the other direction let $k \geq 2$ (for $k \in \{0, 1\}$ is evident) and $L \in \exists^{\sigma}[k]\, \forall^0 \boldsymbol{P}^{\mu\sigma[k]0}$. There exist an $L' \in \boldsymbol{P}^{\mu\sigma[k]0}$ and a polynomial p such that

$$X \in L \Longleftrightarrow \exists^{\sigma} U\, \forall^0 v\, (|v| = p(|X|) \to (X, U, v) \in L')$$

Let M be a polynomial-time machine of type $\mu\sigma[k]\,0$ accepting L' and, let $f, g_1, \ldots, g_k$ be functions computable in polynomial-time such that for M on input (X, U, v) and for $i = 1, \ldots, k$, $g_i(X, v, u_1, \ldots, u_{i-1})$ is the i-th query of M to U assumed the answers to the $i-1$ first queries were $u_1, u_2, \ldots, u_{i-1}$, and $f(X, v, u_1, \ldots, u_k)$ is the result of M on input (X, U, v) if the answers of U to the k queries are $u_1, \ldots, u_k$. Hence,

$$\begin{aligned}
X \in L \Leftrightarrow \exists U\, \forall v \Big(|v| = p(|X|) \to \bigvee_{|u|=k} \Big(& (g_1(X, v) \in U \leftrightarrow u(1) = 1) \wedge \cdots \wedge \\
& \wedge (g_k(X, v, u(1), \ldots, u(k-1)) \in U \leftrightarrow u(k) = 1) \\
& \wedge f(X, v, u(1), \ldots, u(k)) = 1 \Big) \Big) \\
\Leftrightarrow \exists U\, \forall v \Big(|v| = p(|X|) \to \bigwedge_{|u|=k} \Big(& (g_1(X, v) \in U \leftrightarrow u(1) = 0) \vee \cdots \vee \\
& \vee (g_k(X, v, u(1), \ldots, u(k-1)) \in U \leftrightarrow u(k) = 0) \\
& \vee f(X, v, u(1), \ldots, u(k)) = 1 \Big) \Big) \\
\Leftrightarrow \exists U\, \forall v\, \forall u \Big(& (|v| = p(|X|) \wedge |u| = k) \to (X, U, v, u) \in L_1 \Big)
\end{aligned}$$

where $(X, U, v, u) \in L_1 \Longleftrightarrow_{\mathrm{df}} |u| = k \wedge$

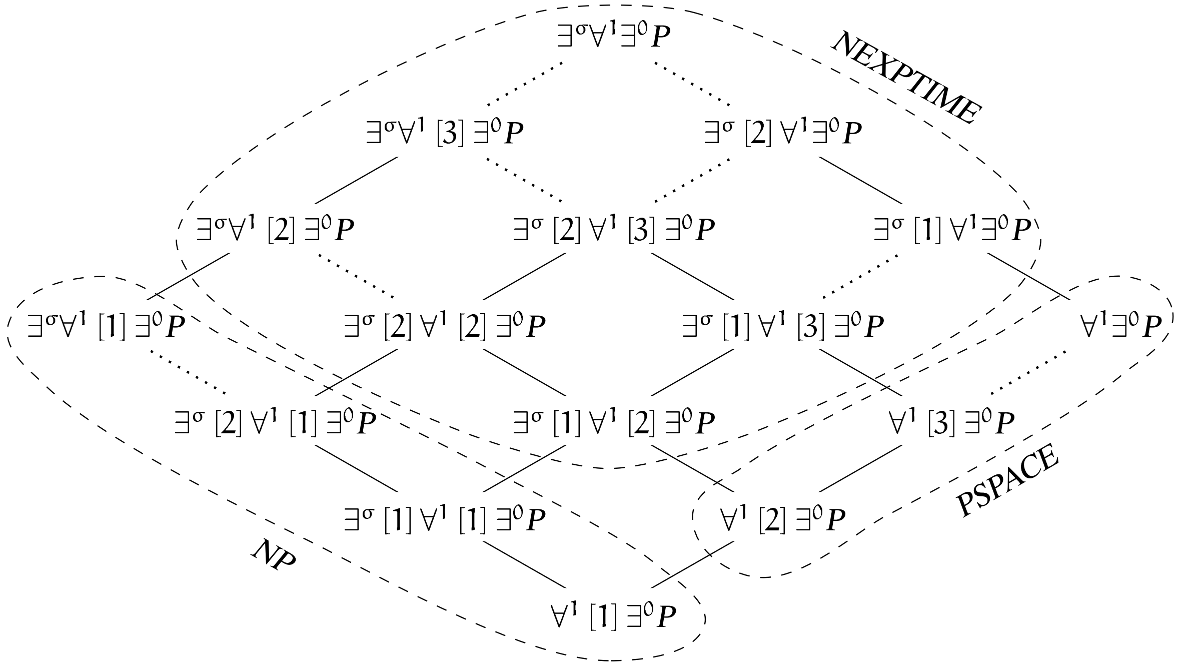
$$\Big((g_1(X, v) \in U \leftrightarrow u(1) = 0) \vee \cdots \vee (g_k(X, v, u(1), \ldots, u(k-1)) \in U \leftrightarrow u(k) = 0)$$
$$\vee f(X, v, u(1), \ldots, u(k)) = 1 \Big)$$

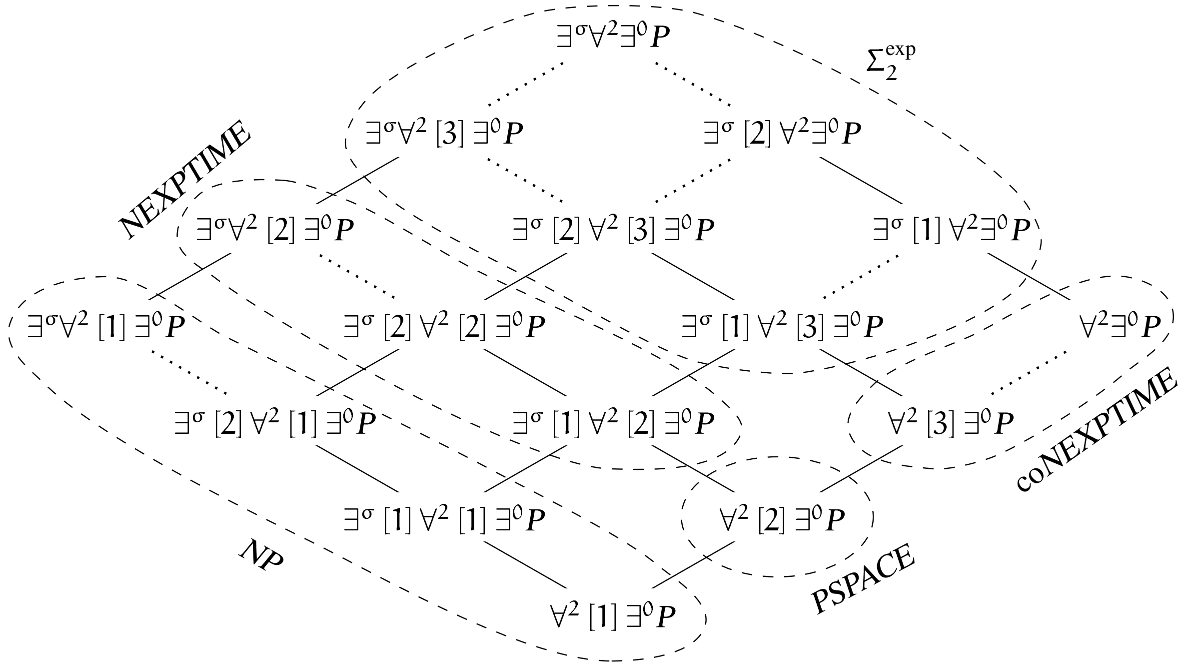
Obviously, a deterministic polynomial-time machine of type $\mu\sigma[k]\,00$ can accept L_1 asking the k queries to the oracle bounded by the last set operator in parallel. Hence, $L_1 \in \boldsymbol{P}^{\mu\sigma[\| k]00}$ and $L \in \exists^{\sigma}[\| k]\, \forall^0 \forall^0 \boldsymbol{P}^{\mu\sigma[\| k]00}$. Then $L \in \exists^{\sigma}[\| k]\, \forall^0 \boldsymbol{P}^{\mu\sigma[\| k]0}$ follows by Lemma 3. ❑

All the characterizations obtained in the previous section remain valid for the counterpart classes $\mathcal{K}(\cdot)$ and co$\mathcal{K}(\cdot)$ having the parallel queries restriction. This can be seen as follows: For the Theorem 8 is evident, since only one query for each oracle is enough. For the other results with exception Theorem 12, only the number of queries to the oracle bounded by the last set operator is relevant (for each one of the remainder oracles only one query is enough). Hence and by Lemma 13 we have the desired characterization. For Theorem 12, only in the proof of Statement (4) can arise problems. However, observe that the proof in [4] can yields $\Sigma_k^{\exp} \subseteq \mathcal{K}\left((1\,[1])^{k-1} 2\,[\|3]\,,1\right)$ and that the rules of Lemma 4 remain valid if we consider the parallel restriction, i.e. $\exists^2\,[\|r] \rightarrow \exists^1\,[1]\,\exists^1\,[\|(r-1)]\,\forall^0$ and $\forall^2\,[\|r] \rightarrow \forall^1\,[1]\,\forall^1\,[\|(r-1)]\,\exists^0$.

7 Conclusions

For $k, m \geq 1$, $\sigma_1, \ldots, \sigma_k \in \{1,2\}$ and functions $r_1, \ldots, r_k : \mathbb{N} \rightarrow \mathbb{N}^+$, in the Section 5 we gave a complete characterization of the classes of the bounded analytic polynomial-time hierarchy having the form $\mathcal{K}(\sigma_1\,[r_1] \ldots \sigma_k\,[r_k]\,, m)$ and co$\mathcal{K}(\sigma_1\,[r_1] \ldots \sigma_k\,[r_k]\,, m)$, and in the Section 6 we show that these characterizations remain valid if the parallel queries restriction is required. Some these results show the power of the set quantifiers with respect to their types and the number of queries. The following figures give an overview on the results on classes $\exists^\sigma\,[r]\,\forall^\tau\,[s]\,\exists^0 \boldsymbol{P}$ where $\sigma, \tau \in \{0,1\}$ and $r, s : \mathbb{N} \rightarrow \mathbb{N}$ with $r \geq 0$ and $s \geq 1$. In the left direction we increase r and in the right direction we increase s (in the figure we write $\forall^\tau\,[s]\,\exists^0 \boldsymbol{P}$ instead of $\exists^\sigma\,[0]\,\forall^\tau\,[s]\,\exists^0 \boldsymbol{P}$).





References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *33rd Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[2] S. Arora and S. Safra. Probabilistic checking of proofs; a new characterization of NP. In *33rd Symposium on Foundations of Computer Science*, pages 2–13, 1992.

[3] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. In *31st Symposium on Foundations of Computer Science*, pages 16–25, 1990.

[4] H. Baier and K. W. Wagner. The analytic polynomial-time hierarchy. Technical Report 148, Institut für Informatik, Universität Würzburg, Germany, September 1996.

[5] R. V. Book, H. Vollmer, and K. W. Wagner. On type-2 probabilistic quantifiers. In *23rd International Colloqium on Automata, Languages and Programming*, volume 1099 of *Lecture Notes in Computer Science*, pages 369–380. Springer Verlag, 1996.

[6] J. Cai and M. Furst. PSPACE survives constant-width bottlenecks. *International Journal of Foundations of Computer Sciences*, 2(1):67–76, 1991.

[7] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *3rd Structure in Complexity Theory Conference*, pages 156–161, 1988.

[8] P. Orponen. Complexity classes of alternating machines with oracles. In *10th International Colloqium on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 573–584. Springer Verlag, 1983.

[9] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company Inc, 1994.

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

[11] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *5th ACM Symposium on the Theory of Computing*, pages 1–9, 1973.

[12] H. Vollmer and K. W. Wagner. On operators of higher types. In *12th Annual IEEE Conference on Computational Complexity*, pages 174–184, 1997.

[13] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.