A Kernel for Function Definable Classes and Its Relation to Lowness

Diana Emme

Lehrstuhl für Theoretische Informatik
Institut für Informatik
Universität Würzburg
D-8700 Würzburg
Germany

Abstract

In the first part of this article we introduce function definability. Let $\mathcal{F}$ be a class of functions. We show that every $\mathcal{F}$-definable class of languages already contains $\mathrm{U}\mathcal{F} \cap \text{co-U}\mathcal{F}$, and if $\mathcal{F}$ is closed under subtraction from one then the kernel itself is $\mathcal{F}$-definable. In the next part we investigate lowness of function classes, where we specialize on classes basing on the class P with arbitrary oracles which are transformed into function classes by the operators #, Gap, span, and F. We prove for each of the four kinds of function classes that exactly the kernel of the classes of sets, which can be defined via functions from the observed class, is low. Moreover, the kernel is equal to the intersection of all these language classes. Next we restrict the class of allowed oracles to classes from the polynomial time counting hierarchy. We observe different ways to characterize classes in the counting hierarchy. Then we look at #-, Gap-, span-, and F-lowness. These cases fall into two groups: the low classes for function classes produced by #, Gap, and span can be expressed in a way that does not depend on P anymore, while the similar result for F-classes would imply the closure under complement for NP.

1 Introduction

Today in the field of complexity theory one can recognize quite a great interest in questions concerning the characterization of complexity classes via classes of counting functions, for example P and NP, the classes in the polynomial time counting hierarchy, or UP.

First steps in this direction were already made by Valiant who introduced the function class # P [Val79], which consists of functions that count the accepting paths of nondeterministic polynomial time bounded Turing machines. It is rather handy to characterize well known classes like NP or PP by the means of # P functions. For example the languages in NP are recognized exactly by nondeterministic polynomial time bounded Turing machines which accept at least on one computation path if and only if the input word belongs to the recognized language. Now one can characterize NP via # P in the following way: for every language L in NP there exists a function f in # P such that $x \in L$ if and only if

$f(x) > 0$. Between P and NP we find the class UP that can easily be defined by functions from $\#\mathrm{P}$ as well: a language L belongs to UP if and only if there exists a function f in $\#\mathrm{P}$ that $f(x) = 1$ if x in L, otherwise $f(x) = 0$.

If one counts the number of different outputs produced by a nondeterministic polynomial time bounded Turing machine instead of counting its accepting paths again one gets a function class called span-P. It was examined by Torán and Köbler [Tor88, Köb89].

Another interesting class of counting functions was defined by Fenner, Fortnow, and Kurtz [FFK91] who extended $\#\mathrm{P}$ by the closure under subtraction and called this Gap-P. Among a lot of other classes, which can be characterized by Gap-functions, there is a class XP[1] defined in analogy to UP. That means if an input is accepted by a Turing machine, the difference between accepting and rejecting computations is exactly one; if the input is rejected the difference is zero. XP has pleasant properties, e. g. XP is the smallest Gap-definable class, and XP is low[2] for Gap-P.

Gupta defined a class that consists of the ratios of Gap-P functions [Gup91] and has nearly the same properties as Gap-P itself. He was able to show the same results for his ratio class as Ogiwara and Hemachandra [OH91] did for $\#\mathrm{P}$ a bit earlier. They found out, e. g. that the polynomial time counting hierarchy collapses if it is possible to reduce the number of accepting computation paths of a Turing machine from a polynomial time computable number to one.

Eventually Vollmer and Wagner systematically studied classes of counting functions in [VW91a, VW91b] and generalized many of the known results about special function classes. They introduced an operator F, a maximum operator in a special sense, that transforms classes of sets into classes of functions with a similar complexity. We will later look at some of these function classes in connection with lowness.

Following the tradition of the cited work we limit our observations to nondeterministic polynomial time bounded computations. The main interest in this paper will be at first the generalization of the mentioned 'smallest class result' in [FFK91] for universal function classes $\mathcal{F}$. We prove the existence of a fixed kernel, the intersection of $\mathrm{U}\mathcal{F}$ and its complement, included in every language class that can be defined via functions from $\mathcal{F}$. Moreover, if we give a special requirement on the defining function class, the kernel itself is definable by that class: if $\mathcal{F}$ is closed under subtraction from one[3] then $\mathrm{U}\mathcal{F}$ is equal to its complement and therefore every $\mathcal{F}$-definable class of languages includes $\mathrm{U}\mathcal{F}$ as the smallest $\mathcal{F}$-definable subclass.

Still concerning the kernel of function definable classes the second part of the paper is dedicated to the question of lowness. We examine lowness for a number of function classes that are produced by applying the operators #, Gap, span, and F to the class P with an oracle from any arbitrary class. This is a rather general approach to look at lowness that was evoked by some results of Fenner, Fortnow, and Kurtz [FFK91], Torán [Tor88], and Köbler [Köb89] – and we are on the way to a rather general result which corresponds to the results of the just mentioned papers: in all of the special cases we examine, we find

[1] XP was introduced in [FFK91] and [OH91] at the same time.

[2] The notation of lowness was born out of a special interest in the weakness of oracles: an oracle is called low for a class if the class is as powerful with as without using the oracle. In structural complexity theory lowness was introduced by Schöning [Sch83].

[3] 'Subtraction from one' means the subtraction from the constant-one-function.

exactly the kernel to be low. But we can give good reasons that the result probably is not universally valid: otherwise $\Sigma_2^p \cap \Pi_2^p$ would be low for Σ_2^p. That would mean $\Sigma_2^p = \Pi_2^p$ and therefore the collapse of the polynomial time hierarchy on the second level.

As a next step of examination we restrict the oracles that P is allowed to use to classes from the polynomial time counting hierarchy. We can show that in the #, span, and Gap cases the low class can be expressed in another way, independent from P – but if this was possible in the F case, the polynomial time hierarchy would collapse on the first level.

Those results are meant to be a first attempt to understand lowness for general complexity classes.

2 Preliminaries

Most of the notation used in this article is standard according to [BDG88, BDG90].

All languages are over some finite alphabet Σ, Σ^* is the set of all words over Σ, and for $x \in \Sigma^*$, $|x|$ denotes the length of x. $\mathbb{N}$ means the set of all natural numbers, $\mathbb{Z}$ the set of integers. The numbers shall be binary encoded. The set of polynomial time computable functions is denoted by FP. As our model of computation we take Turing machines; we abbreviate a nondeterministic Turing machine by NTM and if it additionally uses oracles by OTM.

Further on we give some basic definitions that are essential for the following contents. Valiant introduced the class $\#\mathrm{P}$ in [Val79], and Wagner defined the polynomial time counting hierarchy in [Wag86]. Everything concerning Gap-P is adopted from [FFK91]. The operator F was defined in [VW91b].

- For a given Turing machine M
  $acc_M(x)$ is the number of accepting paths of M on input x,
  $rej_M(x)$ is the number of rejecting paths of M on input x.
- When we talk of function classes, we mean classes $\mathcal{F}$ of the following type:
  Let $\mathbb{D} \in \{\mathbb{N}, \mathbb{Z}\}$. $\mathcal{F}$ consists of functions $f : \Sigma^* \longrightarrow \mathbb{D}$, such that there exists a function $g \in \mathrm{FP}$ and $\forall x \in \Sigma^*,\ f(x) \leq g(x)$.
  The function g acts as a range function for f. We assume at least the constant-zero-function *zero* and the constant-one-function *one* to be in $\mathcal{F}$.
- $\#\mathrm{P}$ was originally defined as the class of all functions acc_M such that M is a polynomial time NTM. We define the # operator for arbitrary classes of languages: let $f : \Sigma^* \longrightarrow \mathbb{N}$, then
  $\#\mathcal{C} = \{\, f \mid (\exists A \in \mathcal{C},\, g \in \mathrm{FP})\ (\forall x)\ [f(x) = \#\{\, y \mid y \leq g(x) \text{ and } (x,y) \in A \,\}] \,\}$.
  A special case is span-$\mathcal{C}$ defined as $\#\exists\mathcal{C}$.[4]
- Let $\mathbb{D} = \mathbb{Z}$.
  Gap-P $= \{\, f \mid (\exists \text{ NTM } M)\ (\forall x)\ [f(x) = acc_M(x) - rej_M(x)] \,\}$. We can relativize the definition of Gap-P by taking an OTM M.

[4]Notice that in contrast to Valiants convention [Val79] we divide between the notations $\#\mathcal{C}$ and $\#\mathrm{P}^{\mathcal{C}}$. For example span-P = NP [VW91a] and span-P = $\#\mathrm{P}^{\mathrm{NP}} \Longrightarrow$ NP = co-NP [Köb89].

- $\mathrm{U}\mathcal{F} = \left\{ L \mid (\exists f \in \mathcal{F})\ (\forall x \in \Sigma^*) \left[\begin{array}{lcl} x \in L & \Leftrightarrow & f(x) = 1 \\ x \notin L & \Leftrightarrow & f(x) = 0 \end{array} \right] \right\}$.

- $\mathrm{X}\mathcal{F} = \left\{ L \mid (\exists f \in \mathcal{F}\,,\, g \in \mathrm{FP})\ (\forall x \in \Sigma^*) \left[\begin{array}{lcl} x \in L & \Leftrightarrow & f(x) = g(x) + 1 \\ x \notin L & \Leftrightarrow & f(x) = g(x) \end{array} \right] \right\}$.

- $\exists\mathcal{F} = \{\, L \mid (\exists f \in \mathcal{F})\ (\forall x \in \Sigma^*)\ [x \in L \Leftrightarrow f(x) > 0] \,\}$.

- $\mathrm{C}\mathcal{F} = \{\, L \mid (\exists f \in \mathcal{F}, g \in \mathrm{FP})\ (\forall x \in \Sigma^*)\ [x \in L \Leftrightarrow f(x) > g(x)] \,\}$.

- For the operators above we introduce the following convention.
  Let $\mathcal{K}$ be a class of languages, $\mathcal{O} \in \{\mathrm{U}, \mathrm{X}, \exists, \mathrm{C}\}$. Then we write $\mathcal{O}\mathcal{K}$ instead of $\mathcal{O}\,\#\,\mathcal{K}$. We also use $\forall\mathcal{K}$ to stand for co-$\exists$co-$\mathcal{K}$.

- The polynomial time counting hierarchy CH is defined inductively as follows:

  1. $\mathrm{P} \in \mathrm{CH}$.
  2. For all classes $\mathcal{K} \in \mathrm{CH}:\ \exists\mathcal{K}, \forall\mathcal{K}, \mathrm{C}\mathcal{K} \in \mathrm{CH}$.
  3. No other class is in CH.

- A function f belongs to the class $\mathrm{F}\mathcal{K}$ if and only if there exists a language L in $\mathcal{K}$ and a function g in FP such that for all x,

  1. $(x, y+1) \in L \Rightarrow (x, y) \in L$ for all y, and
  2. $f(x) = \max\{\, y \mid y \leq g(x) \text{ and } (x, y) \in L \,\}$.

  Note that F applied to P is exactly the well known class of polynomial time computable functions FP, and Vollmer and Wagner showed FNP to be the same as Krentel's class Opt P [Kre88].

3 A kernel of function definable classes

In [FFK91] Fenner, Fortnow, and Kurtz introduced Gap-definability, a handy way to describe classes of languages by functions of the class Gap-P. Very close to their idea we will now give a definition that declares how languages can be characterized by functions of more general classes.

Definition Let $\mathcal{F}$ be a class of functions. Let $A, R \subseteq \Sigma^* \times \mathbb{D}$ be disjoint sets, such that there exist functions f_1, f_2 in $\mathcal{F}$ such that for all $x \in \Sigma^*$, $(x, f_1(x)) \in A$ and $(x, f_2(x)) \in R$.

A language L is said to be in $\mathcal{F}(A, R)$ if and only if there is a function $f \in \mathcal{F}$ that for every $x \in \Sigma^*$,

$$\begin{array}{lcl} x \in L & \Leftrightarrow & (x, f(x)) \in A, \\ x \notin L & \Leftrightarrow & (x, f(x)) \in R. \end{array}$$

A class $\mathcal{C}$ is called $\mathcal{F}$*-definable* if and only if there exist two disjoint sets $A, R \subseteq \Sigma^* \times \mathbb{D}$ such that $\mathcal{C} = \mathcal{F}(A, R)$.

Assume $\mathcal{C}$ to be an $\mathcal{F}$-definable class, and let us examine the question, whether $\mathcal{C}$ contains any non trivial language. The following theorem shows if $\mathcal{F}$ is closed under addition and multiplication then $\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$ is always a subset of $\mathcal{C}$. We call it the *kernel* of each $\mathcal{F}$-definable class $\mathcal{C}$.

But first we note that $\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$ cannot be the empty class.

Proposition 1 *Σ^* and $\emptyset$ are languages in $\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$.*

PROOF *zero* and *one* are functions in $\mathcal{F}$ and they are the characteristic functions for $\emptyset$ and Σ^*. □

Theorem 2 *Let $\mathcal{C}$ be an $\mathcal{F}$-definable class and let $\mathcal{F}$ be closed under addition and multiplication. Then*

$$\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F} \subseteq \mathcal{C}.$$

PROOF If $\mathcal{C} = \mathcal{F}(A, R)$, then there exist $f_1, f_2 \in \mathcal{F}$ such that for all $x \in \Sigma^*$, $(x, f_1(x)) \in A$ and $(x, f_2(x)) \in R$.

Let $L \in \mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$, then there exist f and $\bar{f}$ in $\mathcal{F}$ such that for all x
$x \in L \Leftrightarrow f(x) = 1$ and $x \notin L \Leftrightarrow f(x) = 0$, respectively
$x \in L \Leftrightarrow \bar{f}(x) = 0$ and $x \notin L \Leftrightarrow \bar{f}(x) = 1$.

Now define a function $h = f \cdot f_1 + \bar{f} \cdot f_2$.
Since $\mathcal{F}$ is closed under addition and multiplication, clearly $h \in \mathcal{F}$.
For all $x \in \Sigma^*$:

$$\begin{array}{lllllll} x \in L & \Longrightarrow & f(x) = 1 \wedge \bar{f}(x) = 0 & \Longrightarrow & h(x) = f_1(x) & \Longrightarrow & (x, h(x)) \in A, \\ x \notin L & \Longrightarrow & f(x) = 0 \wedge \bar{f}(x) = 1 & \Longrightarrow & h(x) = f_2(x) & \Longrightarrow & (x, h(x)) \in R. \end{array}$$

Thus L is a language in $\mathcal{C}$. □

Proposition 3 *For any function class $\mathcal{F}$, $\mathrm{U}\mathcal{F}$ and $\text{co-}\mathrm{U}\mathcal{F}$ are $\mathcal{F}$-definable.*

PROOF $\mathrm{U}\mathcal{F} = \mathcal{F}(\Sigma^* \times \{1\}, \Sigma^* \times \{0\})$ and $\text{co-}\mathrm{U}\mathcal{F} = \mathcal{F}(\Sigma^* \times \{0\}, \Sigma^* \times \{1\})$.

Corollary 4 *Let $\mathcal{F}$ be closed under addition and multiplication. Then*

$$\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F} = \bigcap_{\mathcal{C} \text{ is } \mathcal{F}\text{-definable}} \mathcal{C}.$$

Remark $\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$ itself is only known to be $\mathcal{F}$-definable if $\mathrm{U}\mathcal{F}$ is closed under complement. Moreover, if $\{one\} - \mathcal{F} \subseteq \mathcal{F}$ holds (especially if $\mathcal{F}$ is closed under subtraction), it follows easily that $\mathrm{U}\mathcal{F} = \text{co-}\mathrm{U}\mathcal{F}$: for $L \in \mathrm{U}\mathcal{F}$ there exists an $f \in \mathcal{F}$ such that $x \in L \Leftrightarrow f(x) = 1$ and $x \notin L \Leftrightarrow f(x) = 0$. Then $\bar{f} = one - f$ is in $\mathcal{F}$ as well and witnesses that $L \in \text{co-}\mathrm{U}\mathcal{F}$.

So it is clear that especially for function classes $\mathcal{F} = \mathcal{F}' - \mathcal{F}'$, where $\mathcal{F}'$ is closed under addition, we get $\mathrm{U}\mathcal{F} = \text{co-}\mathrm{U}\mathcal{F} = \mathrm{X}\mathcal{F}$ is $\mathcal{F}$-definable and therefore $\mathrm{X}\mathcal{F}$ is the smallest $\mathcal{F}$-definable class. The best known example is $\mathrm{XP}(= \mathrm{UGap}\text{-}\mathrm{P})$.

4 Lowness for some function classes

In connection with oracles there are two main points of view. When adding an oracle to a class of languages one can ask on the one hand how powerful is the new class if one takes a special oracle, or on the other hand what kind of oracles do not help to make the basic class more powerful. The second problem is known as the problem of *lowness*. A number of papers concerning lowness have been published; e. g. about the low hierarchy in NP by Schöning [Sch83].

Our task is to investigate low classes for some function classes that we obtain by applying the operators #, Gap, span, and F to the class P together with an additional oracle from any arbitrary class. It will be interesting to note the connection between these low classes and the kernel. Later we restrict the additional oracles to classes from the polynomial time counting hierarchy where we find new lowness results for #, span, and Gap; on the other side $\mathrm{FP}^{\mathcal{K}}$, $\mathcal{K} \in \mathrm{CH}$ is an example of a function class that behaves a little unlike #, Gap, and span function classes.

Now we give a general definition of lowness for classes of functions or of sets, basing on P:

Definition Let $\mathcal{K}$ be an arbitrary class of sets or functions and $\mathcal{O}$ an operator, transforming classes of sets into either classes of functions or again classes of sets. Then a set L is called $(\mathcal{O}, \mathcal{K})$*-low* if and only if $\mathcal{O}\mathrm{P}^{\mathcal{K},L} = \mathcal{O}\mathrm{P}^{\mathcal{K}}$.
We define $\text{Low-}(\mathcal{O}, \mathcal{K}) = \{ L \mid L \text{ is } (\mathcal{O}, \mathcal{K})\text{-low} \}$.

The notation "$\mathrm{P}^{\mathcal{K},L}$" means that a set in $\mathcal{K}$ and the set L are oracles to the P-computation and none of the both oracles can use the other one. The commitment to the independence of the oracles is necessary because if we allowed the class $\mathcal{K}$ to use L as an oracle we would have to look at low classes for $\mathcal{K}$, but what we really want to do is to add an oracle to P.

The idea to examine lowness for function classes was evoked by some facts from the literature. If you look at the following items 1. and 2. you will find the kernel introduced in Section 3. Let $k \geq 0$. Then for any language L:

1. [Tor88] $\#\,\mathrm{P}^{\Sigma_k^p, L} = \#\,\mathrm{P}^{\Sigma_k^p} \;\Leftrightarrow\; L \in \mathrm{UP}^{\Sigma_k^p} \cap \text{co-}\mathrm{UP}^{\Sigma_k^p}$,
2. [FFK91] $\text{Gap-P}^{L} = \text{Gap-P} \;\Leftrightarrow\; L \in \mathrm{XP}$,
3. [Tor88] $\text{span-P}^{\Sigma_k^p, L} = \text{span-P}^{\Sigma_k^p} \;\Leftrightarrow\; L \in \Sigma_{k+1}^p \cap \Pi_{k+1}^p$.

Our results are a generalization of the just mentioned lowness results.

In the following we direct our attention to some examples of lowness that we introduce in detail now:

1. L is $(\#, \mathcal{K})$-low if and only if $\#\,\mathrm{P}^{\mathcal{K},L} = \#\,\mathrm{P}^{\mathcal{K}}$,
2. L is $(\mathrm{Gap}, \mathcal{K})$-low if and only if $\text{Gap-P}^{\mathcal{K},L} = \text{Gap-P}^{\mathcal{K}}$,

3. L is $(\mathrm{span}, \mathcal{K})$-low if and only if $\#\,\mathrm{NP}^{\mathcal{K},L} = \#\,\mathrm{NP}^{\mathcal{K}}$,

4. L is $(\mathrm{F}, \mathcal{K})$-low if and only if $\mathrm{FP}^{\mathcal{K},L} = \mathrm{FP}^{\mathcal{K}}$.

Note that the items 1. and 3. accord to Torán's definition of # and span-lowness; therefore our definition seems to be natural.

4.1 Lowness for #, span, Gap, and F function classes with arbitrary oracles

In the following subsection we observe four special function classes; they all base on the class P with an oracle from a class $\mathcal{K}$, $\mathcal{K}$ shall not be restricted. The operators #, span, Gap, and F transform this base into function classes. Remember that these function classes define a number of language classes in the sense of Section 3, and one can fix the kernel of all those classes of sets. We can show that it is exactly this kernel which is low for the given function class and can offer a handier form of the low class in each special case.

Theorem 5 *Let $\mathcal{K}$ be an arbitrary class.*

1. $$\begin{aligned}\text{Low-}(\#, \mathcal{K}) &= \mathrm{UP}^{\mathcal{K}} \cap \text{co-UP}^{\mathcal{K}} \\ &= \mathrm{U}\,\#\,\mathrm{P}^{\mathcal{K}} \cap \text{co-U}\,\#\,\mathrm{P}^{\mathcal{K}}.\end{aligned}$$

2. $$\begin{aligned}\text{Low-}(\mathrm{Gap}, \mathcal{K}) &= \mathrm{XP}^{\mathcal{K}} \\ &= \text{UGap-P}^{\mathcal{K}} \cap \text{co-UGap-P}^{\mathcal{K}}.\end{aligned}$$

3. $$\begin{aligned}\text{Low-}(\mathrm{span}, \mathcal{K}) &= \mathrm{NP}^{\mathcal{K}} \cap \text{co-NP}^{\mathcal{K}} \\ &= \text{Uspan-P}^{\mathcal{K}} \cap \text{co-Uspan-P}^{\mathcal{K}}.\end{aligned}$$

4. $$\begin{aligned}\text{Low-}(\mathrm{F}, \mathcal{K}) &= \mathrm{P}^{\mathcal{K}} \\ &= \mathrm{UFP}^{\mathcal{K}} \cap \text{co-UFP}^{\mathcal{K}}.\end{aligned}$$

PROOF 1. We have mentioned a result of Torán where he shows that $\#\,\mathrm{P}^{\Sigma_k^p, L} = \#\,\mathrm{P}^{\Sigma_k^p} \Leftrightarrow L \in \mathrm{UP}^{\Sigma_k^p} \cap \text{co-UP}^{\Sigma_k^p}$. His proof works as well if one does not restrict the oracles to classes from the polynomial time hierarchy.

2. The proof is the relativized version of a proof in [FFK91], where the authors show that $\text{Gap-P}^L = \text{Gap-P} \Leftrightarrow L \in \mathrm{XP}$. In Section 3 we mentioned that $\mathrm{XP} = \text{UGap-P} \cap \text{co-UGap-P}$. Moreover, XP is closed under complement.

3. We find a hint in the literature: $\mathrm{NP}^L = \mathrm{NP} \Leftrightarrow L \in \mathrm{NP} \cap \text{co-NP}$ [Sch83], and the proof turns out to be relativizable. It follows easily that the equivalence still holds after the application of the operator # to NP.
Finally a small sequence makes clear why $\mathrm{NP}^{\mathcal{K}} = \text{Uspan-P}^{\mathcal{K}}$: $\mathrm{NP} = \exists\mathrm{P}$, moreover, Vollmer and Wagner showed that $\text{span-P} = \#\,\exists\mathrm{P}$. It is easy to see that $\mathrm{U}\exists\mathcal{K} \subseteq \exists\exists\mathcal{K} \subseteq \exists\mathcal{K} \subseteq \mathrm{U}\exists\mathcal{K}$.

4. At first we want to prove the equality of $\mathrm{P}^{\mathcal{K}}$ and $\mathrm{UFP}^{\mathcal{K}} \cap \text{co-UFP}^{\mathcal{K}}$. We can show the following relation:

$$\text{For any class } \mathcal{K},\ \mathrm{P}^{\mathcal{K}} = \mathrm{UFP}^{\mathcal{K}}.$$

[$\subseteq$]: This direction is easy to see.
[$\supseteq$]: $L \in \mathrm{UFP}^{\mathcal{K}}$ means that there exists a polynomial time bounded transducer with an oracle from $\mathcal{K}$ which computes the value 1, if the input is in L, and 0, if it is not in L. Clearly this computation can be simulated by a polynomial time bounded Turing machine using an oracle from $\mathcal{K}$. Just interpret acceptance as output of value 1 and rejection as output of value 0.
As $\mathrm{P}^{\mathcal{K}}$ is closed under complement, $\mathrm{UFP}^{\mathcal{K}} = \text{co-UFP}^{\mathcal{K}}$ and so $\mathrm{P}^{\mathcal{K}} = \mathrm{UFP}^{\mathcal{K}} \cap \text{co-UFP}^{\mathcal{K}}$.

Now we are ready to prove the lowness result:
[$\subseteq$]: We can assume that $\mathrm{FP}^{\mathcal{K},L} = \mathrm{FP}^{\mathcal{K}}$; then obviously the characteristic function of L, c_L, is in $\mathrm{FP}^{\mathcal{K}}$. It suffices to show that $L \in \mathrm{UFP}^{\mathcal{K}}$: this requires the existence of a function $f \in \mathrm{FP}^{\mathcal{K}}$ such that

$$\begin{aligned} x \in L &\Leftrightarrow f(x) = 1 \\ x \notin L &\Leftrightarrow f(x) = 0. \end{aligned}$$

Choose $f = c_L$.
[$\supseteq$]: A transducer that recognizes $f \in \mathrm{FP}^{\mathcal{K},L}$ with $L \in \mathrm{P}^{\mathcal{K}}$ can be simulated by another one which replaces the oracle queries to L with a $\mathrm{P}^{\mathcal{K}}$-computation. That means $f \in \mathrm{FP}^{\mathcal{K}}$. □

Let $\mathcal{O} \in \{\#, \text{span-}, \text{Gap-}, \mathrm{F}\}$. Note that the low class for $\mathcal{O}\mathrm{P}^{\mathcal{K}}$, $\mathrm{U}\mathcal{O}\mathrm{P}^{\mathcal{K}} \cap \text{co-U}\mathcal{O}\mathrm{P}^{\mathcal{K}}$, is exactly the intersection of all $\mathcal{O}\mathrm{P}^{\mathcal{K}}$-definable classes.

4.2 Lowness for #, span, Gap, and F function classes with oracles from the polynomial time counting hierarchy

In this subsection we are concerned with similar function classes as before, only the oracle class $\mathcal{K}$ is restricted to classes from the polynomial time counting hierarchy. We bring the low classes for #, span, and Gap to a new form, and on the other hand we show that the same result for F would have serious consequences: the collapse of the counting hierarchy on the first level.

Classes $\mathcal{K}$ in the polynomial time counting hierarchy are usually characterized by quantifiers in the following way: $\mathcal{K} = Q_1 \ldots Q_n \mathrm{P}$ with $n \in \mathbb{N}$ and $Q_i \in \{\exists, \forall, \mathrm{C}\}$. For the following studies it is helpful that there exists an oracle characterization for the classes in CH, which is equivalent to the quantifier characterization. Therefore we must distinguish between the three types of classes in CH, starting with $\exists$, $\forall$ or C and we get three different "bases" for these types, NP, co-NP, and PP.

Theorem 6 [Tor88] *Let $\mathcal{K}$ be in* CH. *Then the following facts hold.*

1. $\exists \mathrm{Q}\mathcal{K} = \mathrm{NP}^{\mathrm{Q}\mathcal{K}} = \mathrm{NP}^{\text{co-}\mathrm{Q}\mathcal{K}}$ *for* $\mathrm{Q} \in \{\forall, \mathrm{C}\}$.[5]
2. $\forall \mathrm{Q}\mathcal{K} = \text{co-NP}^{\mathrm{Q}\mathcal{K}} = \text{co-NP}^{\text{co-}\mathrm{Q}\mathcal{K}}$ *for* $\mathrm{Q} \in \{\exists, \mathrm{C}\}$.

[5]Note that for any class $\mathcal{K} \in$ CH

(a) $\exists \ldots \exists \mathcal{K} = \exists \mathcal{K}$,

(b) $\forall \ldots \forall \mathcal{K} = \forall \mathcal{K}$.

So in the following, without loss of generality, we will sometimes exclude critical quantifiers to avoid the appearance of two $\exists$- or two $\forall$-quantifiers in the beginning of the characterization of classes in CH.

3. $\mathrm{C}\mathcal{K} = \mathrm{PP}^{\mathcal{K}} = \mathrm{PP}^{\text{co-}\mathcal{K}}$.

Note that e. g. $\text{co-NP}^{\exists\mathcal{K}} = \text{co-NP}^{\forall\text{co-}\mathcal{K}}$ because it does not matter if you take a set or its complement as an oracle, but $\forall\exists\mathcal{K}$ is not known to be equal to $\forall\forall\text{co-}\mathcal{K} = \text{co-}\exists\mathcal{K}$.

Now we apply the operator # to the bases NP, co-NP, and PP and discover that if we look at lowness for these classes we have to consider # and span-lowness. The next theorem shows an interesting correlation where the bases # co-NP and # PP are transformed into an expression with the base # P.

Theorem 7

1. [Köb89] $\#\,\text{co-NP} = \#\,\mathrm{P}^{\mathrm{NP}} = \#\,\mathrm{P}^{\text{co-NP}}$.

2. $\#\,\mathrm{PP} = \#\,\mathrm{P}^{\mathrm{PP}}$.

Proof We prove the second equation:

[$\subseteq$]: This direction is obvious.

[$\supseteq$]: Now we show the inclusion for $\#\,\mathrm{P}^{\mathrm{PP}} \subseteq \#\,\mathrm{PP}$:

If $f \in \#\,\mathrm{P}^{\mathrm{PP}}$, then there is a language $A \in \mathrm{P}^{\mathrm{PP}}$ and a function $g \in \mathrm{FP}$ such that for all x

$$f(x) = \#\{\, y \mid y \le g(x) \text{ and } (x,y) \in A \,\}.$$

Let A be recognized by a polynomial time bounded OTM M with oracle $L \in \mathrm{PP}$, and let N be an NTM that accepts L. We may assume without loss of generality that OTM M makes on input x exactly $k(|x|)$ queries to L on every computation path for suitable $k \in \mathrm{FP}$. Now define

$$A' = \left\{ (x, y\$a_1\$\ldots\$a_{k(|x|)}) \;\middle|\; \begin{array}{l} a_1,\ldots,a_{k(|x|)} \text{ are correct answers} \\ \text{on queries of } M \text{ on input } (x,y) \end{array} \right\}.$$

A' is in PP, for there exists an NTM M', which, on path $u = y\$a_1\$\ldots\$a_{k(|x|)}$, simulates M, answers the oracle queries q_i made on input (x,y) by a_i and in the end of the computation it checks, whether the answers were correct. (It works because PP is closed under truth table reduction [FR91].) I. e. there exists a function $g' \in \mathrm{FP}$ such that

$$f(x) = \#\{\, u \mid u \le g'(x) \text{ and } (x,u) \in A' \,\}.$$

Hence $f \in \#\,\mathrm{PP}$. □

Remark Note that # co-NP is equal to $\#\,\mathrm{P}^{\mathrm{NP}}$ and to $\#\,\mathrm{P}^{\text{co-NP}}$. It is not known whether # NP is expressible as # P with any oracle.

We can extend the result of Theorem 7 to all classes in CH, whose quantifier characterizations start with $\forall$ or C, by relativizing the proof. With the help of Theorem 6 we formulate the following corollary:

Corollary 8 *For $\mathcal{K}$ in* CH*,*

$$\begin{aligned} &(a)\ \#\,\forall\mathrm{Q}\mathcal{K} &&= \#\,\mathrm{P}^{\forall\mathrm{Q}\mathcal{K}} \\ & &&= \#\,\mathrm{P}^{\exists\text{co-}\mathrm{Q}\mathcal{K}} \quad \textit{for } \mathrm{Q} \in \{\exists, \mathrm{C}\}, \\ &(b)\ \#\,\mathrm{C}\mathcal{K} &&= \#\,\mathrm{P}^{\mathrm{C}\mathcal{K}}. \end{aligned}$$

We are able to carry over all the $\#\,\mathrm{P}$ results, as Fenner, Fortnow, and Kurtz showed a very pleasant property of Gap-P: Gap-P $= \#\,\mathrm{P} - \#\,\mathrm{P}$ [FFK91].

Extending this property of the operator Gap we define for any class of sets $\mathcal{C}$: Gap-$\mathcal{C} = \#\,\mathcal{C} - \#\,\mathcal{C}$. So we get an analogue to Corollary 8:

Corollary 9 *For $\mathcal{K}$ in* CH*,*

$$\begin{aligned} &(a)\ \text{Gap-}\forall\mathrm{Q}\mathcal{K} &&= \text{Gap-}\mathrm{P}^{\forall\mathrm{Q}\mathcal{K}} \\ & &&= \text{Gap-}\mathrm{P}^{\exists\text{co-}\mathrm{Q}\mathcal{K}} \quad \textit{for } \mathrm{Q} \in \{\exists, \mathrm{C}\}, \\ &(b)\ \text{Gap-}\mathrm{C}\mathcal{K} &&= \text{Gap-}\mathrm{P}^{\mathrm{C}\mathcal{K}}. \end{aligned}$$

Now we turn to the question of lowness again.

Theorem 10 *Let $\mathcal{K}$ be a class in* CH*.*

$$\begin{array}{llrcll} \textit{1.} & (a) & \text{Low-}(\#, \forall\mathrm{Q}\mathcal{K}) & = & \mathrm{U}\forall\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\forall\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\exists, \mathrm{C}\}, \\ & (b) & \text{Low-}(\#, \exists\mathrm{Q}\mathcal{K}) & = & \mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\forall, \mathrm{C}\}, \\ & (c) & \text{Low-}(\#, \mathrm{C}\mathcal{K}) & = & \mathrm{U}\mathrm{C}\mathcal{K} \cap \text{co-}\mathrm{U}\mathrm{C}\mathcal{K}. & \end{array}$$

$$\begin{array}{llrcll} \textit{2.} & (a) & \text{Low-}(\text{Gap}, \forall\mathrm{Q}\mathcal{K}) & = & \mathrm{X}\forall\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\exists, \mathrm{C}\}, \\ & (b) & \text{Low-}(\text{Gap}, \exists\mathrm{Q}\mathcal{K}) & = & \mathrm{X}\forall\text{co-}\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\forall, \mathrm{C}\}, \\ & (c) & \text{Low-}(\text{Gap}, \mathrm{C}\mathcal{K}) & = & \mathrm{X}\mathrm{C}\mathcal{K}. & \end{array}$$

$$\begin{array}{llrcll} \textit{3.} & (a) & \text{Low-}(\text{span}, \forall\mathrm{Q}\mathcal{K}) & = & \exists\forall\mathrm{Q}\mathcal{K} \cap \text{co-}\exists\forall\mathrm{Q}\mathcal{K} & \\ & & & = & \mathrm{U}\exists\forall\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\exists\forall\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\exists, \mathrm{C}\}, \\ & (b) & \text{Low-}(\text{span}, \exists\mathrm{Q}\mathcal{K}) & = & \exists\forall\text{co-}\mathrm{Q}\mathcal{K} \cap \text{co-}\exists\forall\text{co-}\mathrm{Q}\mathcal{K} & \\ & & & = & \mathrm{U}\exists\forall\text{co-}\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\exists\forall\text{co-}\mathrm{Q}\mathcal{K} & \textit{for } \mathrm{Q} \in \{\forall, \mathrm{C}\}, \\ & (c) & \text{Low-}(\text{span}, \mathrm{C}\mathcal{K}) & = & \exists\mathrm{C}\mathcal{K} \cap \text{co-}\exists\mathrm{C}\mathcal{K} & \\ & & & = & \mathrm{U}\exists\mathrm{C}\mathcal{K} \cap \text{co-}\mathrm{U}\exists\mathrm{C}\mathcal{K}. & \end{array}$$

Proof Theorem 10 follows directly from the lowness results in Theorem 5, and the Corollaries 8 and 9.

Exemplary we prove 1.(b): From Theorem 5 we know that for $\mathrm{Q} \in \{\forall, \mathrm{C}\}$, Low-$(\#, \exists\mathrm{Q}\mathcal{K}) = \mathrm{UP}^{\exists\mathrm{Q}\mathcal{K}} \cap \text{co-}\mathrm{UP}^{\exists\mathrm{Q}\mathcal{K}}$, and from Corollary 8 that $\#\,\mathrm{P}^{\exists\mathrm{Q}\mathcal{K}} = \#\,\mathrm{P}^{\forall\text{co-}\mathrm{Q}\mathcal{K}} = \#\,\forall\text{co-}\mathrm{Q}\mathcal{K}$. So we can conclude that $\mathrm{UP}^{\exists\mathrm{Q}\mathcal{K}} \cap \text{co-}\mathrm{UP}^{\exists\mathrm{Q}\mathcal{K}} = \mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K}$, and hence Low-$(\#, \exists\mathrm{Q}\mathcal{K}) = \mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K} \cap \text{co-}\mathrm{U}\forall\text{co-}\mathrm{Q}\mathcal{K}$. □

Let us now look at the lowness results concerning F. We might expect the class $\mathrm{UF}\mathcal{K} \cap \text{co-}\mathrm{UF}\mathcal{K}$ to be equal to Low-$(\mathrm{F}, \mathcal{K})$, but we find out that probably $\mathrm{UF}\mathcal{K} \cap \text{co-}\mathrm{UF}\mathcal{K}$ is a real subset of $\mathrm{P}^{\mathcal{K}}$. It is not difficult to see that $\mathrm{UF}\mathcal{K} \subseteq \mathrm{P}^{\mathcal{K}} (= \mathrm{UFP}^{\mathcal{K}})$ as clearly $\mathrm{F}\mathcal{K} \subseteq \mathrm{FP}^{\mathcal{K}}$; so $\mathrm{UF}\mathcal{K} \cap \text{co-}\mathrm{UF}\mathcal{K}$ is included in $\mathrm{P}^{\mathcal{K}}$. On the other hand it is unlikely that $\mathrm{P}^{\mathcal{K}} \subseteq \mathrm{UF}\mathcal{K} \cap \text{co-}\mathrm{UF}\mathcal{K}$ because of its strong consequences. For example if you take $\mathcal{K} = \mathrm{NP}$ that would mean the collapse of the polynomial time hierarchy on the first level:

Theorem 11 $\mathrm{P}^{\mathrm{NP}} = \mathrm{U\,Opt\,P} \cap \text{co-}\mathrm{U\,Opt\,P} \implies \mathrm{NP} = \text{co-NP}$.

Proof First of all note that $\mathrm{NP} = \mathrm{U\,Opt\,P}$: $\mathrm{NP} \subseteq \mathrm{U\,Opt\,P}$ is trivial; remember that $\mathrm{Opt\,P} = \mathrm{FNP}$, so if $L \in \mathrm{UFNP}$, then there exists a function f in FNP and a set A in NP such that $f(x) = \max\{\, y \mid y \leq 1 \wedge (x,y) \in A \,\}$. This means that $L \in \mathrm{NP}$.

Now we can transform the assumption: $\mathrm{P}^{\mathrm{NP}} = \mathrm{NP} \cap \text{co-NP}$; but any language from NP or from co-NP can be recognized by a P-computation with an oracle from NP. Hence $\mathrm{P}^{\mathrm{NP}} = \mathrm{NP} = \text{co-NP}$. □

4.3 Conclusion

In the last section we have looked at the kernel $\mathrm{U}\mathcal{F} \cap \text{co-}\mathrm{U}\mathcal{F}$ for some function classes $\mathcal{F}$ and its connection to lowness for $\mathcal{F}$. We can as well express our results as examples for a rather generally formulated theorem:

Theorem 12 *Let $\mathcal{K}$ be an arbitrary class, $\mathcal{O}$ an operator in* $\{\#, \exists, \forall, \#\,\exists (= \mathrm{span}), \mathrm{Gap}\}$. *Then*

$$\text{Low-}(\mathcal{O}, \mathcal{K}) = \mathrm{U}\mathcal{O}\mathrm{P}^{\mathcal{K}} \cap \text{co-}\mathrm{U}\mathcal{O}\mathrm{P}^{\mathcal{K}}.$$

There remains some more work to do as two rather natural questions arise:

- What properties must operators $\mathcal{O}$ have in order for Theorem 12 to hold?
- When does Theorem 12 hold for the concatenation of operators (under the assumption that it holds for all of them)?
  Note that the theorem holds for Low-$(\#, -)$, Low-$(\exists, -)$, and Low-$(\forall, -)$. We could prove that it holds for Low-$(\#\,\exists, -)$, too, but if it holds for Low-$(\exists\forall, -)$, i.e. $\exists\forall\mathrm{P}^{\mathrm{U}\exists\forall\mathrm{P} \cap \text{co-}\mathrm{U}\exists\forall\mathrm{P}} = \exists\forall\mathrm{P}$, that would mean $\Sigma_2^p \cap \Pi_2^p$ is low for Σ_2^p and this is probably not true.

Acknowledgements Special thanks are due to Klaus Wagner and Heribert Vollmer for their engaged care, moreover to Gerhard Buntrock, Ulrich Hertrampf, Krzysztof Loryś, Martin Mundhenk and Jacobo Torán for fruitful discussions and many precious hints.

References

[BDG88] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity I.* Springer-Verlag, 1988.

[BDG90] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity II.* Springer-Verlag, 1990.

[FFK91] Steve Fenner, Lance Fortnow, and Stuart Kurtz. Gap-definable counting classes. In *Proc. of 6th Conference on Structure in Complexity Theory*, pages 30–42, 1991.

[FR91] Lance Fortnow and Nick Reingold. PP is closed under truth-table reductions. In *Proc. of 6th Conference on Structure in Complexity Theory*, pages 13–15, 1991.

[Gup91] Sanjay Gupta. The power of witness reduction. In *Proc. of 6th Conference on Structure in Complexity Theory*, pages 43–59, 1991.

[Köb89] Johannes Köbler. *Strukturelle Komplexität von Anzahlproblemen.* PhD thesis, Universität Stuttgart, 1989.

[Kre88] Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and Systems Sciences*, 36:490–509, 1988.

[OH91] Mitsunori Ogiwara and Lane A. Hemachandra. A complexity theory for feasible closure properties. In *Proc. of 6th Conference on Structure in Complexity Theory*, pages 16–29, 1991.

[Sch83] Uwe Schöning. A low and a high hierarchy within NP. *Journal of Computer and Systems Sciences*, 27:14–28, 1983.

[Tor88] Jacobo Torán. *Structural Properties of the Counting Hierarchies.* PhD thesis, Universidad de Barcelona, 1988.

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[VW91a] Heribert Vollmer and Klaus W. Wagner. Classes of counting functions and complexity theoretic operators. Technical Report 31, Universität Würzburg, 1991.

[VW91b] Heribert Vollmer and Klaus W. Wagner. Complexity of functions versus complexity of sets. Technical Report 35, Universität Würzburg, 1991.

[Wag86] Klaus W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta informatica*, 23:325–356, 1986.