On Relation Classes and Solution Relations

Dissertation
zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der
Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Diplom-Mathematiker André Große
geboren am 16. Februar 1975 in Jena

Gutachter:

1. Prof. Dr. Gerd Wechsung
2. Prof. Dr. Ingo Althöfer

Tag der letzten Prüfung des Rigorosums:	4. Juni 2004
Tag der öffentlichen Verteidigung:	1. Juli 2004

To my parents

Acknowledgements

First of all I would like to thank my advisor Professor Gerd Wechsung for his inspiring lecture on "Complexity Theory" during my time as an undergraduate student and for his excellent supervision over the past years. With his inimitable verve and enthusiasm he introduced me into the field of research and without his support neither my "Diplom"-thesis nor the current thesis would have been possible.

Special thanks to all my former and current colleagues from Jena, especially, Harald Hempel, Jörg Rothe, Stefan Schwarz, and Thomas Schneider. Furthermore, I am grateful to all my friends of the $\sqrt{\text{Wurzel}}$-Association for making the last years as interesting as they have been.

For their encouragement and support I am much obliged to my parents, to my family, to all my friends, and, chiefly to my loving girlfriend Sabine. If it had not been for the assistance I have been receiving, creating this thesis would have been impossible.

I further thank Harald Hempel for his generous permission to present the results of our joint paper [GH03] in Chapter 3.

Contents

List of Figures

Chapter 1

Introduction

Computational complexity is concerned with the classification of problems with respect to their complexity. To give a sense to this statement we have to explain some parts of it. If we refer to problems we usually mean decision problems. Does a given object belongs to a certain set or not?

To solve such problems means that we have an algorithm deciding whether a given input x belongs to the set or not.

This gives us a possibility to measure how difficult such a decision problem is, but only with respect to the algorithm used. We measure how many resources the algorithm needs for its decision, depending on the length of the input x.

To achieve this in a reasonable way, a computational model is necessary. In 1936 Turing [Tur36] developed a universal computational model, the so-called Turing machine. We distinguish two versions, a deterministic and a nondeterministic one.

Which resources are usually considered? One possibility is to measure how much time an algorithm needs. For this purpose we count the number of steps carried out by the appropriate Turing machine, from the input up to the final configuration. This allows us to classify problems with regard to the running time of a solving algorithm. In this manner, we only get upper bounds for the complexity of a problem. Optimal lower bounds are harder to determine and sometimes this is impossible.

As an example, we consider the class P, firstly defined by Edmonds [Edm65]. This is the class of all sets that can be decided by a deterministic polynomial-time-bounded Turing machine. That means, for every set in P there exists an appropriate Turing machine that for every input x carries out at most $p(|x|)$ many steps, for some polynomial p. The sets in P are considered as feasible problems. Many natural and nontrivial problems are contained in P, including finding a maximum matching in a general graph [Edm65], linear programming [Kha79] and the problem of testing whether an integer is prime [AKS02].

Furthermore, there exists the class NP – the class of all sets that can be accepted by a nondeterministic polynomial-time-bounded Turing machine. All problems in

P are in NP, too, of course. However, the \$1,000,000 question[1] is: are there sets in NP that are not in P. Nearly all complexity theoreticians would guess that NP contains more sets than P. There are a lot of candidates for such problems. Many of them have the property that they are the "hardest" problems of NP, in the following sense: If only one of these hard problems is in P, then it follows that P = NP. One such problem is the Traveling Sales Person Problem: A sales person wants to visit a number of cities. Is there a route shorter than a given length?

The question whether P = NP has been the starting point of long and intensive research. This research gave rise to a lot of new questions. Many other classes of problems than P and NP were observed allowing a deep understanding of this area. There were many attempts to answer the P = NP question, but this problem has been unsolved by today.

Beside decision problems, relations play an important role in computational complexity. Not only do they appear as tools but also as objects of research themselves. The complexity of relations has received much attention in the last decade. This development was essentially influenced by Selman in the early nineties (see [Sel94, Sel96]).

Many different classes of relations and – as a special case – classes of functions were studied. To mention two classes of relations: FP – the class of all functions computable by a deterministic polynomial-time-bounded Turing machine and NPMV – the class of relations computable by a nondeterministic polynomial-time-bounded Turing machine [BLS84, BLS85]. Exact definitions will be given in Chapter 3.

We follow [Wec00] and [HW00] to define classes of relations. The crucial point of this systematic approach is to base the definition of relation classes on well-studied complexity classes instead of the computation of Turing machines. This approach to classes of relations does not only lead to natural and intuitive notations. It also allows us to prove very general theorems, special cases of which are widely spread over the literature.

Following [Wec00], we define the operators rel and fun which transform a complexity class to a class of relation or a class of functions:

- $r \in \mathrm{rel} \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$ $[r(x) = \{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}]$,
- $f \in \mathrm{fun} \cdot \mathcal{C} \iff f \in \mathrm{rel} \cdot \mathcal{C} \wedge (\forall x \in \Sigma^*)[||f(x)|| \leq 1]$.

First we prove some general results. To give an example: The well-known projection theorem carries over to classes of relations. We will show that even

[1] See http://www.claymath.org/Millennium_Prize_Problems to find out how to earn this money.

though $\mathrm{fun} \cdot \mathrm{NP}$ and $\mathrm{fun} \cdot \mathrm{coNP}$ are incomparable with respect to set inclusion unless $\mathrm{NP} = \mathrm{coNP}$, their counterparts containing only total functions, $\mathrm{fun}_t \cdot \mathrm{NP}$ and $\mathrm{fun}_t \cdot \mathrm{coNP}$, satisfy $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP}$.

We point out a possibility to use relations as oracles. To ask an relation r as an oracle for a word x means to obtain one element of the set $r(x)$. Note that for the same question different answers are possible.

We use the so-called operator method to carry over certain properties from the underlying complexity class to the classes of relations. The operator method was already successfully applied to other scenarios [VW93, HW00] to argue that the inclusions not proven here are unlikely to hold.

Two more examples:

$$\mathrm{rel} \cdot \mathrm{P} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \mathrm{P} \implies \mathrm{NP} = \mathrm{UP}.$$
$$\mathrm{rel} \cdot \mathrm{P}^{\mathrm{NP}} \subseteq_{\mathrm{c}} \mathrm{FP}^{\mathrm{NP}} \implies \mathrm{P}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}}.$$

One type of inclusion for which the operator method fails, will be treated using nonuniform complexity classes. This allows for the following result.

$$\mathrm{rel} \cdot \Pi_k^{\mathrm{p}} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \Pi_k^{\mathrm{p}} \implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma_{k+1}^{\mathrm{p}}}.$$
$$\mathrm{rel} \cdot \Sigma_k^{\mathrm{p}} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \Sigma_k^{\mathrm{p}} \implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma_k^{\mathrm{p}}}.$$

In the second part of this thesis, we study so-called *easy*-languages. These are languages having easily computable solution relations. That means, it is easy to compute on which path a corresponding nondeterministic Turing machine accepts.

This research starts from a result of Borodin and Demers [BD76]. They showed that under a hypothesis most complexity theoreticians would suppose to be true, it follows that there exist easily decidable sets, yet it is hard to compute why, i. e. it is hard to compute the corresponding solution relation.

Following [HRW97], we define two complexity classes, $\mathrm{Easy}_\forall$ and $\mathrm{Easy}_\exists$. The class $\mathrm{Easy}_\forall$ contains all languages for which *every* accepting nondeterministic Turing machine possesses a solution function from FP_t. For $\mathrm{Easy}_\exists$ only *one* Turing machine is required to have an easy solution function.

At first we are interested in what happens if we do not demand for a solution function but a function computing only one bit of an accepting path. Furthermore, we study whether it makes a difference which bit is concerned. It will turn out that it makes no difference.

Further, we ask which languages we obtain if we modify the definition of $\mathrm{Easy}_\exists$ and allow other solution relations instead of the functions from FP_t. We define the operators wsol and ssol mapping from classes of functions to complexity classes.

The classes $\mathrm{wsol} \cdot \mathcal{R}$ and $\mathrm{ssol} \cdot \mathcal{R}$ contain all languages that can be accepted by nondeterministic Turing machines having a weak or a strong solution relation, from $\mathcal{R}$, respectively. The difference between wsol and ssol lies in the treatment of words not belonging to the language in question. For languages in $\mathrm{wsol} \cdot \mathcal{R}$, the solution relations are not defined and for languages in $\mathrm{ssol} \cdot \mathcal{R}$, the solution relations are required to indicate whether a given word does not belong to the language.

We prove the following results among others.

$$
\begin{array}{lcllcl}
\mathrm{wsol} \cdot \mathrm{FP} & = & \mathrm{P} & \mathrm{ssol} \cdot \mathrm{FP} & = & \mathrm{P} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} & = & \mathrm{UP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{P} & = & \mathrm{UP} \cap \mathrm{coUP} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP} & = & \mathrm{UP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{UP} & = & \mathrm{UP} \cap \mathrm{coUP} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} & = & \mathrm{NP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{NP} & = & \mathrm{NP} \cap \mathrm{coNP}
\end{array}
$$

Chapter 2

Preliminaries

In this chapter we define basic concepts of computational complexity that are used in this thesis. Almost everything can be found in a standard book on computational complexity theory, for instance [WW86, BDG88, BDG90, Pap94]. We assume that the reader is familiar with the meaning and notation of the basic set theoretic and logical concepts and introduce only the most important things.

2.1 Words and Languages

Let $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ denote the set of natural numbers and $\mathbb{N}^+ = \mathbb{N} - \{0\}$ the set of all positive natural numbers. Let Pol denote the set of all polynomials in one variable over $\mathbb{N}$.

In complexity theory we study the complexity of sets of words over a finite alphabet. Without loss of generality we use $\Sigma = \{0, 1\}$ as our alphabet. For two words u and v we define the concatenation of u and v as the word uv. For a word w and a language A we define the concatenation as well, $wA = \{wu : u \in A\}$. For letters $a \in \Sigma$ let $a^0 = \varepsilon$ and $a^{n+1} = aa^n$ for all $n \in \mathbb{N}$, where ε denotes the empty word. We define $\Sigma^0 = \{\varepsilon\}$ and $\Sigma^{i+1} = \{uv : u \in \Sigma \wedge v \in \Sigma^i\}$. The set $\Sigma^* = \bigcup_{i \in \mathbb{N}} \Sigma^i$ is the set of all finite words over Σ. The length $|u|$ of a word u is the unique $i \in \mathbb{N}$ such that $u \in \Sigma^i$. For an element $w \in \Sigma^*$, $w = a_1 a_2 a_3 \ldots a_n$, $a_i \in \Sigma$, we define $\mathrm{bit}_i(w) = a_i$ and $\mathrm{lsb}(w) = a_n$.[1]

We define some special subsets of Σ^*. The set $\Sigma^{\leq n} = \bigcup_{i \leq n} \Sigma^i$ of all words of length at most n and the set $\Sigma^{<n} = \bigcup_{i<n} \Sigma^i$ of all words shorter than n.

Let $\leq_{\mathrm{lex}}$ denote the standard quasi lexicographical ordering on Σ^* defined as follows. For two words u and v it holds that $u \leq_{\mathrm{lex}} v$ if and only if $|u| < |v|$, or

[1] The abbreviation lsb stands for least significant bit.

$|u| = |v|$ and there exist three words $w, u', v' \in \Sigma^*$ such that $u = w0u'$ and $v = w1v'$.

A language A over Σ is a subset of Σ^*. For a language A we define $A^{\leq n} = A \cap \Sigma^{\leq n}$, $A^{<n} = A \cap \Sigma^{<n}$ and $A^{=n} = A \cap \Sigma^n$. The cardinality of a set A is denoted by $||A||$. The set FINITE is the set of all finite languages

$$\mathrm{FINITE} = \{L \subseteq \Sigma^* : ||L|| < \infty\}.$$

The characteristic function c_A of a language A is defined as

$$c_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

The complement $\overline{A}$ of a language A in Σ^*, is the set of words not being in A, $\overline{A} = \Sigma^* - A$.

We often need to map pairs of words to words. Let $\langle .,. \rangle$ be a pairing function having the standard properties such as being polynomial-time computable and polynomial-time invertible. We overload the notation $\langle \ldots \rangle$ to also denote pairing functions mapping from $\underbrace{\Sigma^* \times \cdots \times \Sigma^*}_{k}$ to Σ^* for $k \geq 2$, $\mathbb{N} \times \mathbb{N}$ to Σ^* and $\Sigma^* \times \mathbb{N}$ to Σ^* that are also computable and invertible in polynomial time.

Additional to the standard quantifiers $\exists$ and $\forall$, we use the symbol $\exists!!$ to express that there exists something exactly once.

In structural complexity theory sets of languages – so-called complexity classes – are studied. There is a large number of quite useful operators that map complexity classes to complexity classes. Those of them that are used in this thesis will be defined below.

For a complexity class $\mathcal{C}$ the class of all complements of languages in $\mathcal{C}$ is denoted by co$\mathcal{C}$, co$\mathcal{C} = \{\overline{A} : A \in \mathcal{C}\}$. For a complexity class $\mathcal{C}$, $\exists \cdot \mathcal{C}$ is the set of all languages L such that there exists a language $C \in \mathcal{C}$ and a polynomial $p \in \mathrm{Pol}$ such that for all $x \in \Sigma^*$,

$$x \in L \iff (\exists y \in \Sigma^* : |y| \leq p(|x|))[\langle x, y \rangle \in C].$$

Analogously we define $\forall \cdot \mathcal{C}$ to be the set of all languages L such that there exist a language $C \in \mathcal{C}$ and a polynomial $p \in \mathrm{Pol}$ such that for all $x \in \Sigma^*$,

$$x \in L \iff (\forall y \in \Sigma^* : |y| \leq p(|x|))[\langle x, y \rangle \in C].$$

For classes of sets $\mathcal{C}_1$ and $\mathcal{C}_2$,

$$\mathcal{C}_1 \wedge \mathcal{C}_2 = \{A \cap B : A \in \mathcal{C}_1 \wedge B \in \mathcal{C}_2\}$$

and

$$\mathcal{C}_1 \vee \mathcal{C}_2 = \{A \cup B : A \in \mathcal{C}_1 \wedge B \in \mathcal{C}_2\}$$

and

$$\mathcal{C}_1 - \mathcal{C}_2 = \{A - B : A \in \mathcal{C}_1 \wedge B \in \mathcal{C}_2\}.$$

For a set A we define $\mathrm{proj}_1^2(A) = \{x \in \Sigma^* : (\exists y \in \Sigma^*)[\langle x, y\rangle \in A]\}$ and for a class $\mathcal{C}$ we define $A \in \pi_1^2 \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})[A = \mathrm{proj}_1^2(B)]$.

We will need some more operators that will be defined later.

In this thesis we will provide figures that illustrate the inclusion structure of the studied complexity classes. Since $\subseteq$ is a partial order on the power set of Σ^* we will use Hasse diagrams.

2.2 Turing Machines and Reductions

The underlying computational model is the multi-tape Turing machine. A more formal definition can be found in [WW86]. Due to the generally accepted thesis of Church that the intuitively computable functions are the same as the Turing computable ones, we can describe algorithms sometimes in an intuitive way. Polynomial-time Turing machines are Turing machines that on every input x carry out at most polynomially many steps before they reach a final state. We consider deterministic and nondeterministic polynomial-time Turing machines, DPTMs and NPTMs, respectively.

A DPTM M accepts a language L if and only if on every input $x \in \Sigma^*$, M halts on input x in an accepting configuration if and only if $x \in L$.

Without loss of generality, every configuration of a nondeterministic Turing machine that is not final has exactly two succeeding configurations. Let M be a nondeterministic Turing machine and x an input. The tree of all configurations on this computation is denoted by $M(x)$. The root of this tree is the start configuration and every inner node has its two succeeding configurations as children.

A computation path is a path in the computation tree from the root to any leaf. Such a path is represented by a 0-1-word. For this purpose the succeeding configurations of any configuration are identified by 0 and 1, respectively. The set of all accepting paths of a computation $M(x)$ is denoted by $\mathrm{acc}_M(x)$. A NPTM M accepts a language L if and only if on every input $x \in \Sigma^*$, $x \in L$ if and only if there exists an accepting path of $M(x)$. The language $L(M)$ is the set of all inputs accepted by some DPTM or NPTM M.

A normalized computation is a nondeterministic computation if all paths of the computation tree have the same length. If for every input the corresponding computation of a nondeterministic Turing machine is normalized than we call this machine normalized. Without loss of generality all Turing machines are assumed to be normalized in this thesis.

We can provide a Turing machine M with an oracle A as an additional resource. Such an oracle Turing machine M^A has a special query tape in order to test membership of words to a set A, called the oracle. Whenever the machine reaches a special query state it receives the answer "Yes" if the word on the query tape belongs to A and receives "No" otherwise. This answer requires only one step in the computation.

So we can interpret an oracle Turing machine as a Turing machine with a subroutine testing membership for A. The resources needed by this subroutine are irrelevant.

Reductions are the standard method to compare languages with regard to complexity. We will need many-one reductions [Kar72] (also known as Karp reductions) and Turing-reductions [Coo71] (also known as Cook reductions).

Definition 2.2.1 *Let A and B be two languages.*

(1) A language A is said to be many-one reducible to a language B ($A \leq^{\mathrm{p}}_{\mathrm{m}} B$) if and only if there exists a polynomial-time computable total function f such that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \in B.$$

(2) A language A is said to be Turing-reducible to a language B ($A \leq^{\mathrm{p}}_{\mathrm{T}} B$) if and only if there exists an oracle-DPTM M such that

$$A = L(M^B).$$

We define the completeness of a language with respect to a reduction $\leq^{\mathrm{p}}_{\omega}$ as above and a complexity class $\mathcal{C}$. A set A is called $\leq^{\mathrm{p}}_{\omega}$-complete for $\mathcal{C}$ if and only if

(1) $A \in \mathcal{C}$, and

(2) $(\forall X \in \mathcal{C})[X \leq^{\mathrm{p}}_{\omega} A]$.

A class $\mathcal{C}$ is closed under $\leq^{\mathrm{p}}_{\omega}$ reductions, if for all sets A and B,

$$(A \leq^{\mathrm{p}}_{\omega} B \wedge B \in \mathcal{C}) \implies A \in \mathcal{C}.$$

We say that a set is trivial if it is the empty set $\varnothing$ or Σ^*, and otherwise we call it nontrivial. We often need a complexity class $\mathcal{C}$ to be closed under intersection and union, respectively, with P sets. Note that this property is ensured by $\mathcal{C}$ being closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions and containing nontrivial sets. From now on, let a complexity class be a class of sets containing nontrivial sets.

2.3 Important Complexity Classes

The complexity of computations of sets can be compared on the basis of resources which the corresponding Turing machine needs. The main resources we consider are space and time.

The complexity class P is the set of all languages that can be decided by a deterministic polynomial-time Turing machine. Analogously, the complexity class NP is the set of all languages that can be accepted by a nondeterministic polynomial-time Turing machine. The class NP contains $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets. The standard example is the set of all satisfiable boolean formulas SAT.

For a complexity class $\mathcal{C}$, the classes $\mathrm{P}^{\mathcal{C}}$ and $\mathrm{NP}^{\mathcal{C}}$ are the sets of languages that can be decided by a deterministic polynomial-time oracle Turing machine (DPOM) or accepted by a nondeterministic polynomial-time oracle Turing machine (NPOM), respectively, with some oracle from $\mathcal{C}$.

2.3.1 The Polynomial Hierarchy

To provide a generalization of the classes P and NP, the polynomial hierarchy was defined by Meyer and Stockmeyer [MS73, Sto77]. In addition to Meyer and Stockmeyer, Wrathall proved several important properties [Wra77].

Definition 2.3.1 *[MS73, Sto77]*

(1) $\Delta_0^{\mathrm{p}} = \Sigma_0^{\mathrm{p}} = \Pi_0^{\mathrm{p}} = \mathrm{P}$

(2) For $k \geq 1$, $\Delta_k^{\mathrm{p}} = \mathrm{P}^{\Sigma_{k-1}^{\mathrm{p}}}$, $\Sigma_k^{\mathrm{p}} = \mathrm{NP}^{\Sigma_{k-1}^{\mathrm{p}}}$, *and* $\Pi_k^{\mathrm{p}} = \mathrm{co}\Sigma_k^{\mathrm{p}}$.

(3) The polynomial hierarchy is defined as

$$\mathrm{PH} = \bigcup_{k \in \mathbb{N}} \Sigma_k^{\mathrm{p}}.$$

So for instance $\Delta_1^{\mathrm{p}} = \mathrm{P}$, $\Sigma_1^{\mathrm{p}} = \mathrm{NP}$, and $\Delta_2^{\mathrm{p}} = \mathrm{P}^{\mathrm{NP}}$. The concept polynomial hierarchy will be used simultaneously for the complexity class PH and the hierarchy formed by the classes Σ_k^{p}, Π_k^{p}, and Δ_k^{p}, $k \geq 1$.

The inclusion structure of the polynomial hierarchy is shown in Figure 2.1.

The operators $\exists$ and $\forall$ can be used to characterize the Σ_k^{p} and Π_k^{p} levels of the polynomial hierarchy. It is known that $\exists \cdot \Sigma_k^{\mathrm{p}} = \Sigma_k^{\mathrm{p}}$, $\exists \cdot \Pi_k^{\mathrm{p}} = \Sigma_{k+1}^{\mathrm{p}}$, $\forall \cdot \Sigma_k^{\mathrm{p}} = \Pi_{k+1}^{\mathrm{p}}$, and $\forall \cdot \Pi_k^{\mathrm{p}} = \Pi_k^{\mathrm{p}}$ for all $k \geq 1$.

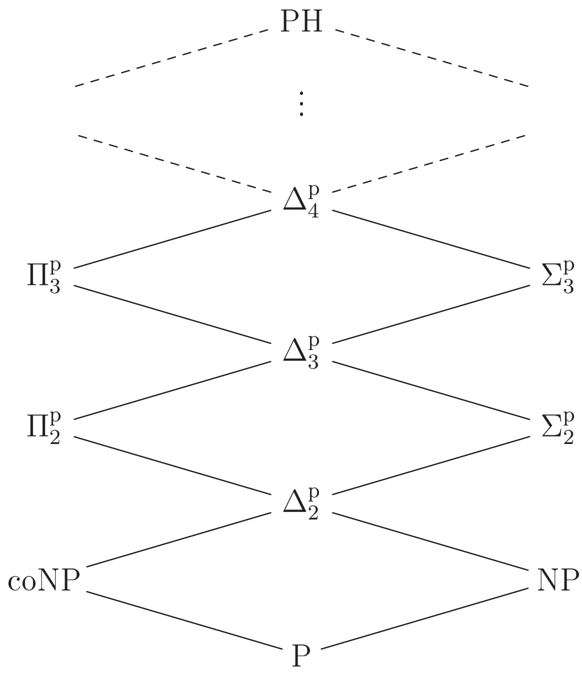


Figure 2.1: The Polynomial Hierarchy

With $\exists \cdot \mathrm{P} = \mathrm{NP}$ and $\forall \cdot \mathrm{P} = \mathrm{coNP}$ we get:

$$\Sigma_k^p = \underbrace{\exists \cdot \forall \cdot \exists \cdot \ \cdots \ \mathcal{Q} \cdot}_{k \text{ alternating operators}} \mathrm{P},$$

where $\mathcal{Q} = \exists$ if k is odd and $\mathcal{Q} = \forall$ if k is even. Similarly,

$$\Pi_k^p = \underbrace{\forall \cdot \exists \cdot \forall \cdot \ \cdots \ \mathcal{Q} \cdot}_{k \text{ alternating operators}} \mathrm{P},$$

where $\mathcal{Q} = \forall$ if k is odd and $\mathcal{Q} = \exists$ if k is even.

All classes of the polynomial hierarchy are closed under many-one reductions and contain many-one complete sets.

It is not known whether the polynomial hierarchy is finite. But there are many conditions under which the polynomial hierarchy collapses. In particular, the polynomial hierarchy satisfies the upward collapse property [Sto77].

For every $k \geq 1$,

(1) $\Sigma_k^{\mathrm{p}} = \Pi_k^{\mathrm{p}} \implies \mathrm{PH} = \Sigma_k^{\mathrm{p}}$.

(2) $\Sigma_k^{\mathrm{p}} = \Sigma_{k+1}^{\mathrm{p}} \implies \mathrm{PH} = \Sigma_k^{\mathrm{p}}$.

(3) $\Delta_k^{\mathrm{p}} = \Sigma_k^{\mathrm{p}} \implies \mathrm{PH} = \Delta_k^{\mathrm{p}}$.

Much more can be said about the polynomial hierarchy. We refer the interested reader to any textbook on complexity theory, for instance [WW86, BDG88, BDG90, Pap94].

2.3.2 The Boolean Hierarchy

The structure of the complexity classes below Δ_2^{p} has been receiving much attention. One hierarchy, the boolean (or hausdorff) hierarchy, is of interest for our work. It has been introduced by a number of authors using a variety of definitions [Wec85, CH86, KSW87, CGH+88, CGH+89].

Hausdorff proved [Hau14] that for a set-ring $\mathcal{S}$ the boolean closure $\mathrm{BC}(\mathcal{S})$ consists of all differences of nested sets from $\mathcal{S}$.[2]

Lemma 2.3.2 *[Hau14] Let $\mathcal{S}$ be a set-ring.*

$$\mathrm{BC}(\mathcal{S}) = \{A_1 \setminus A_2 \setminus \ldots \setminus A_{k-1} \setminus A_k : A_k \subseteq A_{k-1} \subseteq \ldots \subseteq A_1 \wedge A_1, \ldots, A_k \in \mathcal{S} \ \wedge \ k \in \mathbb{N}^+\}.$$

The concept of differences of nested sets can be used to define the hausdorff or boolean hierarchy.

Definition 2.3.3

(1) For all $k \geq 1$,

$$\mathrm{BH}_k(\mathrm{NP}) = \{A_1 \setminus A_2 \setminus \ldots \setminus A_{k-1} \setminus A_k : A_k \subseteq A_{k-1} \subseteq \ldots \subseteq A_1 \wedge A_1, \ldots, A_k \in \mathrm{NP}\}.$$

(2) The hausdorff or boolean hierarchy over NP *is defined as*

$$\mathrm{BH}(\mathrm{NP}) = \bigcup_{k \geq 1} \mathrm{BH}_k(\mathrm{NP}).$$

The classes $\mathrm{BH}_k(\mathrm{NP})$ *and* $\mathrm{coBH}_k(\mathrm{NP})$ *form its k-th level.*

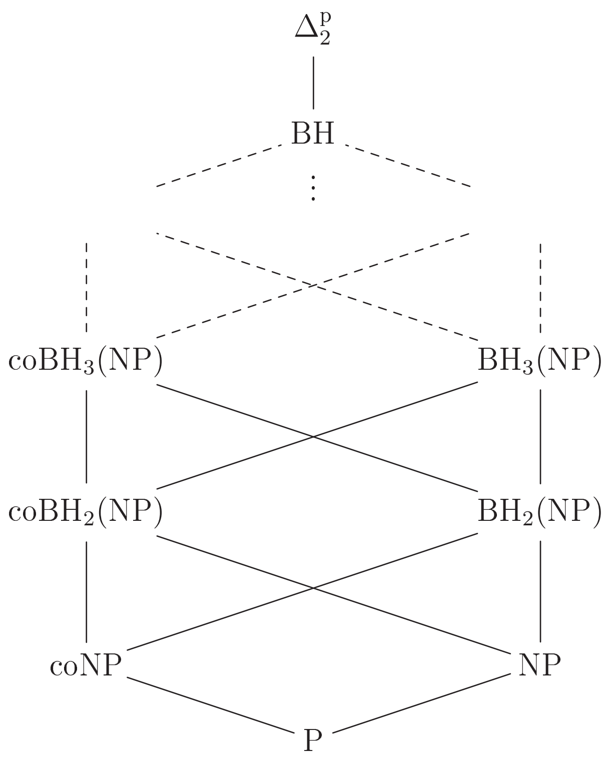


Figure 2.2: The Boolean Hierarchy

We will refer to the boolean hierarchy over NP as the boolean hierarchy and use the classical notation for it. That means $\text{BC}(\text{NP}) = \text{BH}(\text{NP}) = \text{BH}$ and $\text{BH}_k(\text{NP}) = \text{BH}_k$. So for instance BH_2 is exactly the class DP [PY84].

The inclusion structure of the boolean hierarchy is shown in Figure 2.2.

The boolean hierarchy is a well-studied object, a few papers shall be mentioned, [Wec85, CH86, KSW87, CGH+88, CGH+89].

As in the case of the polynomial hierarchy, we do not know whether the boolean hierarchy is finite. But the boolean hierarchy possesses the upward collapse property. In particular, for all $k \geq 1$,

(1) $\text{BH}_k = \text{coBH}_k \implies \text{BH} = \text{BH}_k$.

(2) $\text{BH}_k = \text{BH}_{k+1} \implies \text{BH} = \text{BH}_k$.

[2] For the sake of simplicity, we write $A_1 \setminus A_2 \setminus \ldots \setminus A_{k-1} \setminus A_k$ instead of $A_1 \setminus (A_2 \setminus (\ldots \setminus (A_{k-1} \setminus A_k) \ldots))$.

2.4 Miscellaneous

The concept of Turing machines is a uniform model of computation. Following Schnorr [Sch76] we can use Turing machines in a nonuniform way. We use the definition of Karp and Lipton [KL80]:

Definition 2.4.1 *Let $\mathcal{F}$ bet a set of functions mapping from $\mathbb{N}$ to Σ^* and let $\mathcal{C}$ be a complexity class. The nonuniform complexity class $\mathcal{C}/\mathcal{F}$ is the set of all languages A for which there exist a set $C \in \mathcal{C}$ and a function $f \in \mathcal{F}$ such that for all $x \in \Sigma^*$,*

$$x \in A \iff \langle x, f(|x|)\rangle \in C.$$

We denote the set of all polynomial-length bounded functions by poly,

$$\text{poly} = \{f \in \mathbb{F} : (\exists p \in \text{Pol})(\forall n \in \mathbb{N})[|f(n)| \leq p(n)]\}.$$

The set $\mathbb{F}$ is the set of all functions and particularly contains noncomputable functions.

Furthermore we need the complexity classes UP defined in [Val76], RP and ZPP defined in [Gil77].

A language L belongs to UP if there exists an NPTM M having no accepting path for each $x \notin L$ and accepting on exactly one path for each $x \in L$. For a language L from RP there exists an NPTM M not accepting for all $x \notin L$ and accepting on at least 50% of the paths for each $x \in L$.

It seems that the class RP is not closed under complement. We denote RP∩coRP by ZPP. Note that UP, RP and ZPP are promise classes.

Chapter 3

Function and Relation Classes

In this chapter we present a uniform definition for classes of functions and relations. We completely analyze the inclusion structure of such classes. In order to compare classes of relations and functions with respect to the existence of refinements, we extend the so-called operator method [VW93, HW00] to make it applicable to such cases. Our approach sheds new light on well-studied classes like NPSV and NPMV, allows to give simpler proofs for known results, and shows that the spectrum of function and relation classes closely resembles the spectrum of well-known complexity classes.

3.1 Introduction

In his influential papers "A Taxonomy of Complexity Classes of Functions" [Sel94] and "Much Ado about Functions" [Sel96], Selman started a line of research that studies the structural complexity of classes of relations and functions. In this paper an important role is played by the function class NPSV and the relation class NPMV (see [BLS84, BLS85]). A function f is in NPSV if and only if there exists a nondeterministic polynomial-time Turing machine (NPTM) M such that for all $x \in \Sigma^*$, $f(x)$ is the only output made on any path of $M(x)$ if $f(x)$ is defined, and $M(x)$ outputs no value if $f(x)$ is undefined. NPSV stands for nondeterministically polynomial-time computable "single-valued functions". Since NPTMs have the ability to compute different values on different computation paths, it is natural to define a class that takes advantage of this. A relation r is in NPMV if and only if there exists an NPTM M such that for all $x \in \Sigma^*$, $\langle x, y\rangle \in r$, if and only if y is output on some computation path of $M(x)$.[1] NPMV stands for nondeterministically polynomial-time computable "multi-valued functions". The classes NPMV and NPSV have played an important role in studying the possibility of computing unique solutions [HNOS96]. Other pa-

[1] The literature uses the notation $r(x) \mapsto y$.

pers have studied the power of NPMV and NPSV when used as oracles [FHOS97] and complements of NPMV relations [FGH+96].

Even though NPMV and the notion of relations are well-established in theoretical computer science, we will take a mathematical point of view and call the objects in NPMV and similarly in any class $\mathrm{rel} \cdot \mathcal{C}$ relations.

In this chapter we take a systematic approach to classes like $\mathrm{fun} \cdot \mathrm{NP}$ and $\mathrm{rel} \cdot \mathrm{NP}$. Our approach to classes of functions and relations does not only lead to natural and intuitive notations. It also allows to prove very general theorems, special cases of which are widely spread over the literature. We mention that a systematic approach to function and relation classes yields obvious notational benefits (see in [HV95]) and has been successfully taken for classes of median functions in [VW93] and for classes of optimization functions in [HW00].

The crucial point of this systematic approach is to base the definition of relation classes on well-studied complexity classes instead of the computation of Turing machines. We will focus on function and relation classes being defined over the polynomial hierarchy, though our results apply to a wide variety of complexity classes. Following Wechsung [Wec00] we define general operators fun and rel. For a complexity class $\mathcal{C}$ let

(1) $r \in \mathrm{rel} \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$[r(x) = \{y : |y| \leq p(|x|) \wedge \langle x, y \rangle \in B\}]$.

(2) $f \in \mathrm{fun} \cdot \mathcal{C} \iff f \in \mathrm{rel} \cdot \mathcal{C} \wedge (\forall x \in \Sigma^*)[||f(x)|| \leq 1]$.

One can easily see that $\mathrm{rel} \cdot \mathrm{NP} = \mathrm{NPMV}$ and $\mathrm{fun} \cdot \mathrm{NP} = \mathrm{NPSV}$. Interestingly enough, also $\mathrm{rel} \cdot \mathrm{P}$ and $\mathrm{fun} \cdot \mathrm{P}$ have appeared in the literature before, denoted by NPMV_g and NPSV_g [Sel96], respectively. The class $\mathrm{rel} \cdot \mathrm{coNP}$ has been studied in detail in [FGH+96], dubbed as complements of NPMV relations.

Our approach sheds new light on a wide variety of seemingly isolated results involving the mentioned function and relation classes. For instance, the difference hierarchy based on NPMV as considered in [FHOS97] is the "rel-version" of the boolean hierarchy (over NP), i. e., for all k, $\mathrm{NPMV}(k) = \mathrm{rel} \cdot \mathrm{BH}_k$. After proving a number of inclusion relations we use the so-called operator method that has already been successfully applied to other scenarios [VW93, HW00] to argue that the inclusions we did not prove are unlikely to hold. We extend the operator method to make it applicable to the case of comparing classes of functions and relations.

The chapter is organized as follows. After giving the most relevant definitions in Section 3.2 we prove general results regarding the inclusion relations of classes of functions and classes of relations in Section 3.4. The interaction of operators as $\exists, \forall$, and others with our operators fun and rel is studied in section 3.5. This enables us to use the operator method for our purposes in Section 3.6 and we completely analyze the inclusion structure of classes of functions and classes of relations that are based on P, NP and coNP. In particular, not only do we give the positive inclusion results all of which follow from the theorems of Section 3.4, but we also show that the positive results given are the best to be expected, under reasonable complexity theoretic assumptions. The latter is achieved by exploiting the modified operator method and the results from section 3.5. As an example, it turns out that even though $\mathrm{fun} \cdot \mathrm{NP}$ and $\mathrm{fun} \cdot \mathrm{coNP}$ are incomparable with respect to set inclusion unless $\mathrm{NP} = \mathrm{coNP}$, their counterparts containing only total functions, $\mathrm{fun}_t \cdot \mathrm{NP}$ and $\mathrm{fun}_t \cdot \mathrm{coNP}$, satisfy $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP}$. In Section 3.7 we generalize an idea from [HNOS96] and obtain some structural consequences for inclusions for which the operator method fails.

3.2 Basic Definitions

A relation r over Σ^* is a subset of Σ^*, i. e. x and y are in relation r if and only if $\langle x, y\rangle \in r$. The domain of r is $\mathrm{dom}(r) = \{x \in \Sigma^* : (\exists y \in \Sigma^*)[\langle x, y\rangle \in r]\}$ and the range of r is $\mathrm{range}(r) = \{y \in \Sigma^* : (\exists x \in \Sigma^*)[\langle x, y\rangle \in r]\}$. For all $x \in \Sigma^*$, let $r(x) = \{y \in \Sigma^* : \langle x, y\rangle \in r\}$.

For two relations r_1 and r_2 we define the concatenation $r_1 \cdot r_2$ as follows

$$r_1 \cdot r_2 = \{\langle x, uv\rangle : \langle x, u\rangle \in r_1 \ \wedge \ \langle x, v\rangle \in r_2\}.$$

We define the concatenation of two classes of relations R_1 and R_2 as well,

$$R_1 \cdot R_2 = \{r_1 \cdot r_2 : r_1 \in R_1 \ \wedge \ r_2 \in R_2\}.$$

For relations r_1 and r_2, r_1 is called a refinement of r_2 if and only if $\mathrm{dom}(r_1) = \mathrm{dom}(r_2)$ and $r_1 \subseteq r_2$. If r_1 is a refinement of r_2 and r_1 is a function we write $r_1 \preceq_{ref} r_2$. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be classes of relations, we define $\mathcal{R}_2 \subseteq_c \mathcal{R}_1$ if and only if every relation $r_2 \in \mathcal{R}_2$ has a refinement $r_1 \in \mathcal{R}_1$.

Following [VW93] the operator U is defined as follows: $A \in \mathrm{U} \cdot \mathcal{C}$ if and only if there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

(a) $||\{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| \leq 1$ and

(b) $x \in A \iff ||\{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1.$

It is not hard to see that $\mathrm{U} \cdot \mathrm{P} = \mathrm{UP}$ and $\mathrm{U} \cdot \mathrm{NP} = \mathrm{NP}$.

The following classes of functions and relations will be of interest.

Definition 3.2.1

(1) The function class FP *is the set of all partial functions computed by deterministic polynomial-time Turing machines.*

For any complexity class $\mathcal{C}$ let

(2) $\mathrm{FP}^{\mathcal{C}}$ *(*$\mathrm{FP}_{\|}^{\mathcal{C}}$*) be the set of all functions that can be computed by deterministic polynomial-time oracle Turing machines with adaptive (nonadaptive/parallel) oracle queries to an oracle from* $\mathcal{C}$*,*

(3) [Wec00] $r \in \mathrm{rel} \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$[r(x) = \{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}]$,

(4) [Wec00] $f \in \mathrm{fun} \cdot \mathcal{C} \iff f \in \mathrm{rel} \cdot \mathcal{C} \wedge (\forall x \in \Sigma^*)[||f(x)|| \leq 1]$,

(5) [HW00] $f \in \max \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$[f(x) = \max\{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}]$,

(6) [HW00] $f \in \min \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$[f(x) = \min\{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}]$,

(7) [WT92] $f \in \# \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$[f(x) = ||\{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}||]$.

Note that the classes $\min \cdot \mathcal{C}$ and $\max \cdot \mathcal{C}$ may contain partial functions in contrast to the original definition in [HW00]. This is due to the fact that we use the definition that the minimum and the maximum of the empty set is not defined.

Clearly, for all classes $\mathcal{C}$ closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions, $\mathrm{rel} \cdot \mathcal{C}$ and $\mathrm{fun} \cdot \mathcal{C}$ are in fact subsets of $\mathcal{C}$, $\mathrm{rel} \cdot \mathcal{C}$ being a set of (polynomially length-bounded) relations and $\mathrm{fun} \cdot \mathcal{C}$ a set of (polynomially length-bounded) functions. For any relation class $\mathcal{R}$ defined above, the subset of all total functions or relations will be denoted with the additional subscript t, $\mathcal{R}_t$.

In regard to computing a relation r, we want to point out that instead of deciding membership to r, we are interested in computing $r(x)$ for any given x.

Note that by definition FP, $\mathrm{fun} \cdot \mathcal{C}$, $\max \cdot \mathcal{C}$, and $\min \cdot \mathcal{C}$ are sets of functions mapping from Σ^* to Σ^*, whereas in contrast $\# \cdot \mathcal{C}$ is a set of functions mapping from Σ^* to $\mathbb{N}$. In order to study the inclusion structure between $\mathrm{fun} \cdot \mathcal{C}$ and $\mathrm{rel} \cdot \mathcal{C}$ on the one hand and $\# \cdot \mathcal{C}$ on the other hand, we have to look at the "mapping-from-Σ^*-to-$\mathbb{N}$" version of $\mathrm{fun} \cdot \mathcal{C}$ and $\mathrm{rel} \cdot \mathcal{C}$. Of course this does not pose a serious problem since there exist easily, i. e., polynomial-time, computable and invertible bijections between Σ^* and $\mathbb{N}$ allowing us to take either view at the objects in $\mathrm{fun} \cdot \mathcal{C}$ or $\mathrm{rel} \cdot \mathcal{C}$ for complexity classes $\mathcal{C}$ having nice closure properties. In light of this comment, recall that $\max \cdot \mathcal{C}$ and $\min \cdot \mathcal{C}$ have originally been defined as sets of functions mapping from Σ^* to $\mathbb{N}$ [HW00].

For some complexity classes $\mathcal{C}$, $\mathrm{fun} \cdot \mathcal{C}$ and $\mathrm{rel} \cdot \mathcal{C}$ are well-known classes and have been studied in the literature before.

Proposition 3.2.2

(1) $\mathrm{rel} \cdot \mathrm{P} = \mathrm{NPMV}_g$.

(2) $\mathrm{rel} \cdot \mathrm{NP} = \mathrm{NPMV}$.

(3) $\mathrm{rel} \cdot \mathrm{coNP} = \mathrm{coNPMV}$.

(4) $\mathrm{fun} \cdot \mathrm{P} = \mathrm{NPSV}_g$.

(5) $\mathrm{fun} \cdot \mathrm{UP} = \mathrm{UPF}$.

(6) $\mathrm{fun} \cdot \mathrm{NP} = \mathrm{NPSV}$.

For instance NPMV, NPSV, NPMV_g, and NPSV_g have been defined and studied in [Sel96], coNPMV was defined in [FHOS97] and UPF can be found in [BGH90]. A different framework for defining and generalizing function classes has been considered in [KSV98].

We define the following operators on classes of relations. This is a generalization of the definition in [Hem03], where some of these operators were defined on classes of functions.

Definition 3.2.3 *For any class $\mathcal{R}$ of relations let*

(1) *(see also [VW93])* $A \in \mathcal{U} \cdot \mathcal{R} \iff c_A \in \mathcal{R}$,

(2) $A \in \mathrm{Sig} \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\forall f \preceq_{ref} r)(\forall x \in \Sigma^*)$
$[x \in A \iff f(x) \in \Sigma^* - \{\varepsilon\}]$,

(3) $A \in \mathrm{SIG} \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\forall f \preceq_{ref} r)(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$
$\Big[\big(f(x) \leq_{lex} 1^{p(|x|)}\big) \wedge \big(x \in A \iff f(x) <_{lex} 1^{p(|x|)}\big)\Big]$,

(4) $A \in \mathrm{C}_{\geq} \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\exists g \in \mathrm{FP}_t)(\forall f \preceq_{ref} r)(\forall x \in \Sigma^*)$
$[x \in A \iff f(x) \geq_{lex} g(x)]$,

(5) $A \in \mathrm{C}_{=} \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\exists g \in \mathrm{FP}_t)(\forall f \preceq_{ref} r)(\forall x \in \Sigma^*)$
$[x \in A \iff f(x) = g(x)]$,

(6) $A \in \mathrm{C}_{\leq} \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\exists g \in \mathrm{FP}_t)(\forall f \preceq_{ref} r)(\forall x \in \Sigma^*)$
$[x \in A \iff f(x) \leq_{lex} g(x)]$,

(7) $A \in \oplus \cdot \mathcal{R} \iff (\exists r \in \mathcal{R})(\forall f \preceq_{ref} r)(\forall x \in \Sigma^*)$
$[x \in A \iff$ *the least significant bit of* $f(x)$ *is* $1]$.

The for-all-refinements quantifier allows us to state Theorem 3.6.2 which will be the key lemma for the operator method. We would not be able to prove Theorem 3.6.2 if we used the existence quantifier instead. If $\mathcal{R}$ is a class of functions, the for-all-refinements quantifier is superfluous. Note that the operators defined above can easily be modified to apply to classes of functions that map to $\mathbb{N}$. For instance, in the definition of Sig one has to change "$f(x) \in \Sigma^* - \{\varepsilon\}$" to "$f(x) > 0$" or in the definition of $\oplus$ one has to change "the least significant bit of $f(x)$ is 1" to "$f(x) \equiv 1 \bmod 2$" (see [HW00]). Note that in general $\mathrm{U} \cdot \mathcal{C} = \mathcal{U} \cdot \# \cdot \mathcal{C}$ (see also [HVW95]). It follows, for instance, $\mathrm{U} \cdot \mathrm{coNP} = \mathrm{U} \cdot \mathrm{P}^{\mathrm{NP}}$ or equivalently $\mathrm{U} \cdot \mathrm{coNP} = \mathrm{UP}^{\mathrm{NP}}$, since it is known that $\# \cdot \mathrm{coNP} = \# \cdot \mathrm{P}^{\mathrm{NP}}$ [KST89].

3.3 Relations as Oracles

We mention classes of relations computed in polynomial time with access to an oracle. If the oracle is a relation, we use the oracle in a different way from the case of a standard set oracle. Let f be a function. For a Turing machine M with access to f as an oracle, we write $M^{(f)}$. This is like a common oracle Turing machine with the following difference. If the machine asks the oracle about a word x then it receives the value $f(x)$ instead of a "Yes/No" answer. If $x \notin \mathrm{dom}(f)$, the machine receives the special symbol $\perp$.

Using this, we can define the classes $\mathrm{FP}^{\mathcal{R}}$, $\mathrm{P}^{\mathcal{R}}$ and $\mathrm{NP}^{\mathcal{R}}$.

Definition 3.3.1 *Let r be a relation, and $\mathcal{R}$ be a class of relations.*

(1) $\mathrm{FP}^r = \{g : (\exists M^{()} : M^{()}$ *is a DPOM*$)(\forall f \preceq_{ref} r)[M^{(f)}$ *computes* $g]\}$.

(2) $\mathrm{FP}^{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \mathrm{FP}^r$.

(3) $\mathrm{P}^r = \{L : (\exists M^{()} : M^{()}$ *is a DPOM*$)(\forall f \preceq_{ref} r)[L(M^{(f)}) = L]\}$.

(4) $\mathrm{P}^{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \mathrm{P}^r$.

(5) $\mathrm{NP}^r = \{L : (\exists M^{()} : M^{()}$ *is a NPOM*$)(\forall f \preceq_{ref} r)[L(M^{(f)}) = L]\}$.

(6) $\mathrm{NP}^{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \mathrm{P}^r$.

Note that this definition involves classes for which the oracle is from a class of functions, since every function f has a unique refinement, namely f itself.

The above definition for $\mathrm{FP}^{\mathcal{R}}$ and $\mathrm{P}^{\mathcal{R}}$, respectively, differs from that definition given in [FHOS97].

The authors gave the following definitions:

$$\mathrm{FP}^r = \{s : (\exists g \preceq_{ref} s)(\exists M^{()} : M^{()} \text{ is a DPOM})(\forall f \preceq_{ref} r)[M^{(f)} \text{ computes } g]\}.$$

$$\mathrm{FP}^{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \mathrm{FP}^r.$$

This implies that noncomputable relations are contained in $\mathrm{FP}^{\mathrm{FP}}$.

Let K an arbitrary nondecidable set, for instance the Halting problem. We define the following relation

$$r = c_K \cup \Sigma^* \times \{2\}.$$

The constant function $f(x) \equiv 2$ is obviously a refinement of r and of course contained in $\mathrm{FP}^{\mathrm{FP}}$. But the relation r is noncomputable, at least in the following sense.

A relation r is called computable, if and only if there exists a Turing machine M which for every input x outputs the set $r(x)$.

Note that every relation r satisfying $||\{x : ||r(x)|| \geq 2\}|| = \infty$ contains uncountably many refinements, thus some of them are noncomputable.

For these reasons, we use Definition 3.3.1 to avoid such problems.

A third possibility to define such classes would be to replace the for-all-refinement quantifier by the existence quantifier in Definition 3.3.1.

3.4 General Results

As already mentioned, our definition of the operators fun and rel captures a number of well-known function and relation classes. We will now state quite general results regarding the operators fun and rel.

Clearly fun and rel (and also fun_t and rel_t) are monotone (with respect to set inclusion) operators mapping complexity classes to relation or function classes. Moreover, the two operators rel and fun preserve the inclusion structure of the complexity classes they are applied to.

Theorem 3.4.1 *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ complexity classes both being closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions. The following statements are equivalent:*

(1) $\mathcal{C}_1 \subseteq \mathcal{C}_2$.

(2) $\mathrm{rel} \cdot \mathcal{C}_1 \subseteq \mathrm{rel} \cdot \mathcal{C}_2$.

(3) $\mathrm{fun} \cdot \mathcal{C}_1 \subseteq \mathrm{fun} \cdot \mathcal{C}_2$.

Proof The implications (1) $\rightarrow$ (2) and (2) $\rightarrow$ (3) are obvious. We show (3) $\rightarrow$ (1):

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be complexity classes such that $\mathcal{C}_1$ is closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions. Suppose $\mathrm{fun}\cdot\mathcal{C}_1 \subseteq \mathrm{fun}\cdot\mathcal{C}_2$. Let $A \in \mathcal{C}_1$. Define a function f to be $f = \{\langle x, 1\rangle : x \in A\}$ and note that $f \in \mathcal{C}_1$ since $\mathcal{C}_1$ is closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions. Clearly $f \in \mathrm{fun}\cdot\mathcal{C}_1$. By our assumption $\mathrm{fun}\cdot\mathcal{C}_1 \subseteq \mathrm{fun}\cdot\mathcal{C}_2$ it follows that $f \in \mathrm{fun}\cdot\mathcal{C}_2$.

Hence there exist a set B in $\mathcal{C}_2$ and a polynomial p such that for all $x \in \Sigma^*$,

$$f(x) = \{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

It follows that for all $x \in \Sigma^*$, $x \in A \iff \langle x, 1\rangle \in B$. In other words, $A \leq^{\mathrm{p}}_{\mathrm{m}} B$ and thus, since $\mathcal{C}_2$ is closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions, $A \in \mathcal{C}_2$. ❑

It follows from Theorem 3.4.1 that $\mathrm{rel}\cdot\mathrm{P} \subseteq \mathrm{rel}\cdot\mathrm{NP} \cap \mathrm{rel}\cdot\mathrm{coNP}$ and that $\mathrm{rel}\cdot\mathrm{NP}$ and $\mathrm{rel}\cdot\mathrm{coNP}$ are incomparable with respect to set inclusion unless $\mathrm{NP} = \mathrm{coNP}$. Note that when replacing fun and rel by fun_t and rel_t, respectively, in the above theorem only the implications (1) $\rightarrow$ (2) and (2) $\rightarrow$ (3) hold. See Corollary 3.4.7 for example.

Observation 3.4.2

For classes of relations $\mathcal{R}$ and $\mathcal{S}$ it holds that $\mathcal{R} \subseteq_{\mathrm{c}} \mathcal{S} \implies \mathcal{R}_t \subseteq_{\mathrm{c}} \mathcal{S}_t$.

Thus all inclusions that hold between classes of partial relations do also hold between the corresponding classes of total relations. However, some inclusions between classes of total functions do not carry over to their partial counterparts unless some unlikely complexity class collapses occur. For instance we will see that $\mathrm{fun}_t\cdot\mathrm{NP} \subseteq \mathrm{fun}_t\cdot\mathrm{coNP}$, yet $\mathrm{fun}\cdot\mathrm{NP} \not\subseteq \mathrm{fun}\cdot\mathrm{coNP}$ unless $\mathrm{NP} = \mathrm{coNP}$.

A first link between classes of relations on the one side and classes of sets on the other side is given in the following theorem.

Theorem 3.4.3 *For any complexity class $\mathcal{C}$ being closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions and any set A,*

(1) $A \in \exists\cdot\mathcal{C}$ if and only if $A = dom(r)$ for some relation $r \in \mathrm{rel}\cdot\mathcal{C}$.

(2) $A \in \mathrm{U}\cdot\mathcal{C}$ if and only if $A = dom(f)$ for some function $f \in \mathrm{fun}\cdot\mathcal{C}$.

Proof (1) Let r be a binary relation over Σ^*, $\mathcal{C}$ be a complexity class being closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$ reductions, and $A \in \exists\cdot\mathcal{C}$.

Suppose $r \in \mathrm{rel}\cdot\mathcal{C}$. Hence, there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

By definition, $dom(r) = \{x : (\exists y)[\langle x, y\rangle \in r]\}$. But note that for all $x, y \in \Sigma^*$,

$$\langle x, y\rangle \in r \iff |y| \leq p(|x|) \wedge \langle x, y\rangle \in B.$$

It follows that $dom(r) \in \exists \cdot \mathcal{C}$.

Assume $A \in \exists \cdot \mathcal{C}$. Hence, there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$x \in A \iff (\exists y \in \Sigma^*)[|y| \leq p(|x|) \wedge \langle x, y\rangle \in B].$$

Define

$$r = \{\langle x, y\rangle : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}$$

and observe that $r \in \mathrm{rel} \cdot \mathcal{C}$ and also $dom(r) = A$.

The proof of (2) is the same as above and thus omitted. ❏

In [FGH+96], the authors noted that $\mathrm{rel} \cdot \mathrm{coNP}$ is surprisingly powerful since $\mathrm{rel} \cdot \mathrm{coNP}$ relations are almost as powerful as relations from $\mathrm{rel} \cdot \Sigma_2^{\mathrm{p}}$.

We strengthen this to the claim that the well-known projection theorem carry over to classes of relations.

Theorem 3.4.4

If a complexity class $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions then $\mathrm{rel} \cdot \exists \cdot \mathcal{C} \subseteq \pi_1^2 \cdot \mathrm{rel} \cdot \mathcal{C}$.

Proof Let $r \in \mathrm{rel} \cdot \exists \cdot \mathcal{C}$. Hence there exist a set $A \in \exists \cdot \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : \ |y| \leq p(|x|) \wedge \langle x, y\rangle \in A\}.$$

It follows that there also exist a set $B \in \mathcal{C}$ and a polynomial q such that for all $x, y \in \Sigma^*$,

$$\langle x, y\rangle \in A \iff (\exists z : |z| \leq q(|\langle x, y\rangle|))[\langle x, y, z\rangle \in B].$$

Hence, for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \wedge (\exists z : |z| \leq q(|\langle x, y\rangle|))[\langle x, y, z\rangle \in B]\}.$$

Define

$$B' = B \cap \{\langle x, y, z\rangle : |y| \leq p(|x|) \wedge |z| \leq q(|\langle x, y\rangle|)\}.$$

Clearly, $B' \in \mathcal{C}$. Let q' be a polynomial such that for all $x, y, z \in \Sigma^*$ satisfying $|y| \leq p(|x|)$ and $|z| \leq q(|\langle x, y\rangle|)$ it holds that $|\langle y, z\rangle| \leq q'(|x|)$.

Define a relation s such that for all $x \in \Sigma^*$,

$$s(x) = \{\langle y, z\rangle : |\langle y, z\rangle| \leq q'(|x|) \ \wedge \ \langle x, y, z\rangle \in B'\}.$$

Note that $s \in \mathrm{rel} \cdot \mathcal{C}$. However, for all $x \in \Sigma^*$, $r(x) = \mathrm{proj}_1^2(s(x))$. It follows that $r \in \pi_1^2 \cdot \mathrm{rel} \cdot \mathcal{C}$. ❏

Corollary 3.4.5

(1) $\mathrm{rel} \cdot \mathrm{NP} \subseteq \pi_1^2 \cdot \mathrm{rel} \cdot \mathrm{P}$.

(2) *[FGH+96]* $\mathrm{rel} \cdot \Sigma_2^{\mathrm{p}} \subseteq \pi_1^2 \cdot \mathrm{rel} \cdot \mathrm{coNP}$.

Theorem 3.4.6 *Let $\mathcal{C}$ be a complexity class being closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions.*

$$\mathrm{fun}_t \cdot \mathcal{C} \subseteq \mathrm{fun}_t \cdot \mathrm{co}(\mathrm{U} \cdot \mathcal{C}).$$

Proof Let $f \in \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exist a set $A \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$\langle x, y\rangle \in f \iff |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in A.$$

Or equivalently, since f is a total function we have:

$$\langle x, y\rangle \notin f \iff (\exists y' : y \neq y' \ \wedge \ y' \leq p(|x|))[\langle x, y'\rangle \in A].$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ and f is a total function the right side of the last equivalence is an $\mathrm{U} \cdot \mathcal{C}$ predicate. So we have that $f \in \mathrm{fun}_t \cdot \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$ ❏

Corollary 3.4.7

(1) $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP}$

(2) $\mathrm{fun}_t \cdot \Sigma_2^{\mathrm{p}} \subseteq \mathrm{fun}_t \cdot \Pi_2^{\mathrm{p}}$

Note that in contrast $\mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{fun} \cdot \mathrm{coNP} \iff \mathrm{NP} = \mathrm{coNP}$.

Historically, classes like FP and in general $\mathrm{F}\Delta_k^{\mathrm{p}} = \mathrm{FP}^{\Sigma_{k-1}^{\mathrm{p}}}$, $k \geq 1$, have been among the first function classes studied in complexity theory. We will now see how these classes relate to classes $\mathrm{fun} \cdot \mathcal{C}$ and $\mathrm{rel} \cdot \mathcal{C}$.

Theorem 3.4.8 *Let $\mathcal{C}$ be a complexity class being closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions.*

(1) $\mathrm{fun}_t \cdot \mathcal{C} \subseteq (\mathrm{FP}_t)_{\|}^{\mathrm{U}\cdot\mathcal{C}\cap\mathrm{co}(\mathrm{U}\cdot\mathcal{C})}$.

(2) $\mathrm{fun} \cdot \mathcal{C} \subseteq \mathrm{rel} \cdot \mathcal{C} \subseteq_{\mathrm{c}} \mathrm{FP}^{\exists\cdot\mathcal{C}}$.

Proof (1) Let $f \in \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exist a set $A \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$, $\langle x, y\rangle \in f \iff |y| \leq p(|x|) \wedge \langle x, y\rangle \in A$.

We define a set B as follows:

$$B = \{\langle\langle x, 0^i\rangle, a\rangle : x \in \Sigma^* \wedge a \in \{0,1\} \wedge (\exists y : y \in \Sigma^* \wedge |y| \leq p(|x|) \wedge \mathrm{bit}_i(y) = a)[\langle x, y\rangle \in A]\}.$$

Since f is a total function it holds that $B \in \mathrm{U}\cdot\mathcal{C}$. From

$$\langle\langle x, 0^i\rangle, a\rangle \notin B \iff (\exists y : y \in \Sigma^* \wedge |y| \leq p(|x|) \wedge |y| < i)[\langle x, y\rangle \in A] \oplus (\exists y : y \in \Sigma^* \wedge |y| \leq p(|x|) \wedge \mathrm{bit}_i(y) \neq a)[\langle x, y\rangle \in A]$$

it follows that $\overline{B} \in \mathrm{U}\cdot\mathcal{C}$ too and so $B \in \mathrm{co}(\mathrm{U}\cdot\mathcal{C})$.

We can compute $f(x)$ in polynomial-time by submitting the following queries

$$\langle\langle x, 0^1\rangle, 0\rangle, \langle\langle x, 0^1\rangle, 1\rangle, \langle\langle x, 0^2\rangle, 0\rangle, \langle\langle x, 0^2\rangle, 1\rangle, \ldots, \langle\langle x, 0^{p(|x|)}\rangle, 0\rangle, \langle\langle x, 0^{p(|x|)}\rangle, 1\rangle$$

in parallel to the oracle B. This shows $\mathrm{fun}_t \cdot \mathcal{C} \subseteq (\mathrm{FP}_t)_{\|}^{\mathrm{U}\cdot\mathcal{C}\cap\mathrm{co}(\mathrm{U}\cdot\mathcal{C})}$.

(2) The inclusion $\mathrm{fun} \cdot \mathcal{C} \subseteq \mathrm{rel} \cdot \mathcal{C}$ is obvious. It remains to show $\mathrm{rel} \cdot \mathcal{C} \subseteq_{\mathrm{c}} \mathrm{FP}^{\exists\cdot\mathcal{C}}$.

Let $r \in \mathrm{rel} \cdot \mathcal{C}$. Hence there exist a set $A \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \wedge \langle x, y\rangle \in A\}.$$

We define

$$B = \{\langle x, y\rangle : (\exists z \in \Sigma^*)[\langle x, z\rangle \in r \wedge z \leq_{\mathrm{lex}} y]\}.$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions, $B \in \exists \cdot \mathcal{C}$. Now we can compute a refinement of r. For a given x we query B in a binary search manner to find the lexicographically smallest string $y \in r(x)$. ❑

Theorem 3.4.9

(1) For all $k \in \mathbb{N}$, $\mathrm{F}\Delta_k^{\mathrm{p}} \subseteq \mathrm{fun} \cdot \Delta_k^{\mathrm{p}}$

(2) [HHN+ 93] $\mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}} \subseteq \mathrm{fun} \cdot \mathrm{NP}$

Proof (1) is obvious.

(2) Let $f \in \mathrm{FP}^A$ with $A \in \mathrm{NP} \cap \mathrm{coNP}$. Hence we have NPTMs M_1 and M_2 for A and $\overline{A}$, respectively. In a Turing machine M that computes f with oracle A we can substitute a question to the oracle by running the machines M_1 and M_2 in parallel. On accepting paths of M_1 we continue in the same way as with a "Yes" answer from the oracle and on accepting paths of M_2 we continue in the same way as with a "No" answer from the oracle. This Turing machine computes f nondeterministically and shows $f \in \mathrm{fun} \cdot \mathrm{NP}$. ❏

In [FHOS97] the power of $\mathrm{rel} \cdot \mathrm{NP}$ and $\mathrm{fun} \cdot \mathrm{NP}$ oracles has been studied. We prove some generalized results.

Theorem 3.4.10 *Let $\mathcal{C}$ be a complexity class.*

(1) $\mathrm{FP}^{\mathcal{C}} \subseteq \mathrm{FP}^{\mathrm{fun}\cdot\mathcal{C}} \subseteq \mathrm{FP}^{\mathrm{rel}\cdot\mathcal{C}} = \mathrm{FP}^{\exists\cdot\mathcal{C}}$.

(2) $\mathrm{FP}_{||}^{\mathcal{C}} \subseteq \mathrm{FP}_{||}^{\mathrm{fun}\cdot\mathcal{C}} \subseteq \mathrm{FP}_{||}^{\mathrm{rel}\cdot\mathcal{C}}$

Proof (1) We will show the inclusions and equalities from left to right. Let $f \in \mathrm{FP}^{\mathcal{C}}$ via a DPOM M and an oracle $B \in \mathcal{C}$.

Define a function $g = \{\langle x, 1\rangle : x \in B\}$. Note that $g \in \mathrm{fun} \cdot \mathcal{C}$ and for all $x \in \Sigma^*$, $x \in B$ if and only if $g(x) = 1$. By modifying M in the obvious way it is clear that a DPOM with oracle g can compute f.

The inclusion $\mathrm{FP}^{\mathrm{fun}\cdot\mathcal{C}} \subseteq \mathrm{FP}^{\mathrm{rel}\cdot\mathcal{C}}$ is obvious.

It remains to show $\mathrm{FP}^{\mathrm{rel}\cdot\mathcal{C}} = \mathrm{FP}^{\exists\cdot\mathcal{C}}$. Let $f \in \mathrm{FP}^{\mathrm{rel}\cdot\mathcal{C}}$. Hence there exist a DPOM M and a relation $r \in \mathrm{rel} \cdot \mathcal{C}$ such that M with oracle r computes f. Note that for all inputs x, and all queries q generated by $M(x)$, and for all of the possibly different answers the oracle may give to a query "$? \in r(q)$", $M(x)$ computes the same value $f(x)$.

Informally put, by Theorem 3.4.12 we know that r has a refinement g, that is even a function in $\min \cdot \mathcal{C}$. Recall that $M^{(g)}$ by definition computes f. We use a binary search strategy to find $g(q)$ for any query q (generated by $M(x)$) with the help of an $\exists \cdot \mathcal{C}$ oracle. More formally, let $B \in \mathcal{C}$ and p be a polynomial such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \ \wedge\ \langle x, y\rangle \in B\}.$$

Define the set

$$D = \{\langle x, w\rangle : (\exists z : |wz| \leq p(|x|))[\langle x, wz\rangle \in B]\}.$$

Obviously, $D \in \exists \cdot \mathcal{C}$. Observe that any query "$f(q) =?$" made during a computation by M can be replaced by a series of queries to D, where we query D in a binary search manner to find the lexicographically smallest string ω such that $\omega \in r(q)$. It is not difficult to see that M can be modified in a way to query D instead of r and still compute the same function f.

Now suppose that $f \in \mathrm{FP}^{\exists\cdot\mathcal{C}}$ via a DPOM M and a set $D \in \exists\cdot\mathcal{C}$. Hence there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$x \in D \iff (\exists y : |y| \leq p(|x|))[\langle x, y\rangle \in B].$$

We define a relation r such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \;\wedge\; \langle x, y\rangle \in B\}.$$

Clearly, any query q made by M to the oracle D can be replaced by a query "$? \in r(q)$". If the latter returns a string then $q \in D$, and if the latter returns the special symbol that signals $r(x) = \varnothing$ then $q \notin D$.

(2) It is not difficult to see that the first two proofs above also work for the case that all queries are made in parallel. ❑

Theorem 3.4.11 *Let $\mathcal{C}$ be a complexity class.*

(1) $\mathrm{rel}\cdot(\mathrm{P}^{\mathrm{rel}\cdot\mathcal{C}}) = \mathrm{rel}\cdot(\mathrm{P}^{\exists\cdot\mathcal{C}})$.

(2) $\mathrm{rel}\cdot(\mathrm{NP}^{\mathrm{rel}\cdot\mathcal{C}}) = \mathrm{rel}\cdot(\mathrm{NP}^{\exists\cdot\mathcal{C}})$.

Proof The proof is analogous to the proof of Theorem 3.4.10. ❑

Other types of well-studied classes of functions are classes of optimization and counting functions.

Theorem 3.4.12 *Let $\mathcal{C}$ be a complexity class being closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions and intersection.*

(1) $\max\cdot\mathcal{C} \cap \min\cdot\mathcal{C} = \mathrm{fun}\cdot\mathcal{C} \subseteq \mathrm{rel}\cdot\mathcal{C}$.

(2) $\mathrm{rel}\cdot\mathcal{C} \subseteq_{\mathrm{c}} \min\cdot\mathcal{C} \subseteq \mathrm{fun}\cdot(\mathcal{C} \;\wedge\; \forall\cdot\mathrm{co}\mathcal{C})$.

(3) $\mathrm{rel}\cdot\mathcal{C} \subseteq_{\mathrm{c}} \max\cdot\mathcal{C} \subseteq \mathrm{fun}\cdot(\mathcal{C} \;\wedge\; \forall\cdot\mathrm{co}\mathcal{C})$

Proof (a) $\max\cdot\mathcal{C} \cap \min\cdot\mathcal{C} = \mathrm{fun}\cdot\mathcal{C}$:

Let $\mathcal{C}$ be a complexity class being closed under intersection. Let $f \in \mathrm{fun}\cdot\mathcal{C}$. Hence there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \mathrm{dom}(f)$,

$$||\{y : |y| \leq p(|x|) \;\wedge\; \langle x, y\rangle \in B\}|| = 1$$

and $f(x)$ is the unique string y, $|y| \leq p(|x|)$, such that $\langle x, y\rangle \in B$. Obviously, for every $x \in \mathrm{dom}(f)$,

$$f(x) = \max\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\} \qquad \text{and}$$
$$f(x) = \min\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

For all $x \notin \mathrm{dom}(f)$, we have

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 0.$$

Note that the maximum and the minimum of the empty set are not defined.

It follows that $f \in \max \cdot \mathcal{C} \cap \min \cdot \mathcal{C}$.

Now suppose $f \in \max \cdot \mathcal{C} \cap \min \cdot \mathcal{C}$. Hence there exist sets $C_1, C_2 \in \mathcal{C}$ and polynomials p_1, p_2 such that for all $x \in \Sigma^*$,

$$f(x) = \max\{y : |y| \leq p_1(|x|) \ \wedge \ \langle x, y\rangle \in C_1\} \qquad \text{and}$$
$$f(x) = \min\{y : |y| \leq p_2(|x|) \ \wedge \ \langle x, y\rangle \in C_2\}.$$

Define the set B to be

$$B = \{\langle x, y\rangle : \langle x, y\rangle \in C_1 \cap C_2\} \cap \{\langle x, y\rangle : |y| \leq \min\{p_1(|x|), p_2(|x|)\}\}$$

and let p be a polynomial satisfying for all n, $p(n) \geq \max\{p_1(n), p_2(n)\}$. Observe that $B \in \mathcal{C}$ and that for all $x \in \mathrm{dom}(f)$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1$$

and $f(x)$ is the unique string y, $|y| \leq p(|x|)$, such that $\langle x, y\rangle \in B$. For all $x \notin \mathrm{dom}(f)$ we have

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 0.$$

It follows that

$$f(x) = \{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

Hence $f \in \mathrm{fun} \cdot \mathcal{C}$.

(b) The statements $\mathrm{fun} \cdot \mathcal{C} \subseteq \mathrm{rel} \cdot \mathcal{C}$, $\mathrm{rel} \cdot \mathcal{C} \subseteq_{\mathrm{c}} \max \cdot \mathcal{C}$ and $\mathrm{rel} \cdot \mathcal{C} \subseteq_{\mathrm{c}} \min \cdot \mathcal{C}$ are obvious.

(c) $\max \cdot \mathcal{C} \cup \min \cdot \mathcal{C} \subseteq \mathrm{fun} \cdot (\mathcal{C} \ \wedge \ \forall \cdot \mathrm{co}\mathcal{C})$:

Let $f \in \max \cdot \mathcal{C}$ (the case $f \in \min \cdot \mathcal{C}$ is analogous). Hence there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$f(x) = \max\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

Equivalently we can state for all $x \in \Sigma^*$,

$$f(x) = y \iff \langle x, y\rangle \in B \ \wedge\ (\forall z : y <_{\text{lex}} z \ \wedge\ |z| \leq p(|x|))[\langle x, z\rangle \notin B].$$

The right hand side of the above equivalence clearly describes a predicate from $\mathcal{C} \ \wedge\ \forall \cdot \text{co}\mathcal{C}$ and thus $f \in \text{fun} \cdot (\mathcal{C} \ \wedge\ \forall \cdot \text{co}\mathcal{C})$. ❑

At the end we will take a quick look at the connection between fun-rel classes and classes of counting functions. Note that classes $\# \cdot \mathcal{C}$ are by definition classes of total functions. Since $\# \cdot \mathcal{C}$ contains functions mapping from Σ^* to $\mathbb{N}$ we now look at the mapping-from-Σ^*-to-$\mathbb{N}$ version of $\text{fun} \cdot \mathcal{C}$ and $\text{rel} \cdot \mathcal{C}$.

Theorem 3.4.13 *Let $\mathcal{C}$ be a complexity class being closed under $\leq_{\text{m}}^{\text{p}}$ reductions.*

(1) $\text{fun}_t \cdot \mathcal{C} \subseteq \# \cdot \mathcal{C}$.

(2) $\text{rel}_t \cdot \mathcal{C} \subseteq_{\text{c}} \# \cdot \exists \cdot \mathcal{C}$.

Proof (1) Let $f \in \text{fun}_t \cdot \mathcal{C}$ and let $B \in \mathcal{C}$ be a set and p be a polynomial such that

$$f = \{\langle x, y\rangle : y \leq 2^{p(|x|)} \ \wedge\ \langle x, y\rangle \in B\}.$$

Define a set D to be

$$D = \{\langle x, y, z\rangle : \langle x, y\rangle \in B \ \wedge\ y \leq 2^{p(|x|)} \ \wedge\ 0 \leq z < y\}.$$

Since $\mathcal{C}$ is closed under $\leq_{\text{m}}^{\text{p}}$ reductions we conclude $D \in \mathcal{C}$. It follows that there exists a polynomial r such that for all $x \in \Sigma^*$,

$$f(x) = ||\{w : w \leq 2^{r(|x|)} \ \wedge\ \langle x, w\rangle \in D\}||.$$

Hence $f \in \# \cdot \mathcal{C}$.

(2) According to Theorem 3.4.12 we have $\text{rel}_t \cdot \mathcal{C} \subseteq_{\text{c}} \max \cdot \mathcal{C}$ and hence $\text{rel}_t \cdot \mathcal{C} \subseteq_{\text{c}} \max \cdot \exists \cdot \mathcal{C}$. It was shown in [HW00] that $\max \cdot \exists \cdot \mathcal{C} \subseteq \# \cdot \exists \cdot \mathcal{C}$. ❑

Theorem 3.4.14 *For any complexity classes $\mathcal{C}, \mathcal{K}$ closed under $\leq_{\text{m}}^{\text{p}}$ reductions,*

$$\# \cdot \mathcal{K} \subseteq \text{fun}_t \cdot \mathcal{C} \iff \# \cdot co\mathcal{K} \subseteq \text{fun}_t \cdot \mathcal{C}.$$

Proof We have to prove only one direction. Let $f \in \# \cdot \mathcal{K}$. Hence there exists a polynomial p and a set $B \in \mathcal{K}$ such that

$$f(x) = ||\{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge\ \langle x, y\rangle \in B\}||.$$

We define

$$g(x) = ||\{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge \ \langle x, y \rangle \notin B\}||.$$

Obviously, $g \in \# \cdot co\mathcal{K}$ and $f(x) = 2^{p(|x|)} - g(x)$. By our assumption we have $g \in \mathrm{fun}_t \cdot \mathcal{C}$ and there exist a polynomial q and a set $D \in \mathcal{C}$ such that

$$g(x) = z \iff z \leq 2^{q(|x|)} \ \wedge \ \langle x, z \rangle \in D.$$

The set

$$D' = \{\langle x, z \rangle : z \leq 2^{q(|x|)} \ \wedge \ \langle x, 2^{p(|x|)} - z \rangle \in D\}$$

is also from $\mathcal{C}$, since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. So we have

$$f(x) = z \iff 2^{p(|x|)} - 2^{q(|x|)} \leq z \leq 2^{p(|x|)} \ \wedge \ \langle x, z \rangle \in D'.$$

Without loss of generality, $p(n) < q(n)$ for all n and since $z \in \mathbb{N}$:

$$f(x) = z \iff z \leq 2^{p(|x|)} \ \wedge \ \langle x, y \rangle \in D'.$$

It follows that $f \in \mathrm{fun}_t \cdot \mathcal{C}$. ❑

3.5 Operators on Function and Relation Classes

In this section our focus is on the interaction of various operators with classes of the form $\mathrm{fun} \cdot \mathcal{C}$ or $\mathrm{rel} \cdot \mathcal{C}$ where $\mathcal{C}$ is a complexity class.

Theorem 3.5.1 *Let $\mathcal{C}$, $\mathcal{C}_1$, and $\mathcal{C}_2$ be complexity classes. Let $\mathcal{C}$ be closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$.*

(1) $\mathrm{rel} \cdot (\mathcal{C}_1 \ \wedge \ \mathcal{C}_2) = \mathrm{rel} \cdot \mathcal{C}_1 \ \wedge \ \mathrm{rel} \cdot \mathcal{C}_2$.

(2) $\mathrm{rel} \cdot (\mathcal{C}_1 \ \vee \ \mathcal{C}_2) = \mathrm{rel} \cdot \mathcal{C}_1 \ \vee \ \mathrm{rel} \cdot \mathcal{C}_2$.

(3) $\mathrm{rel} \cdot (\mathcal{C}_1 \cap \mathcal{C}_2) = \mathrm{rel} \cdot \mathcal{C}_1 \cap \mathrm{rel} \cdot \mathcal{C}_2$.

(4) $\mathrm{rel} \cdot (\mathcal{C}_1 \cup \mathcal{C}_2) = \mathrm{rel} \cdot \mathcal{C}_1 \cup \mathrm{rel} \cdot \mathcal{C}_2$.

(5) $\mathrm{rel} \cdot (\mathrm{co}\mathcal{C}) = \mathrm{co}\,(\mathrm{rel} \cdot \mathcal{C})$.

Proof (1). Suppose $r \in \mathrm{rel} \cdot (\mathcal{C}_1 \wedge \mathcal{C}_2)$. Hence there exist a set $B \in \mathcal{C}_1 \wedge \mathcal{C}_2$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le p(|x|) \wedge \langle x, y\rangle \in B\}.$$

Thus there also exist sets $C_1 \in \mathcal{C}_1$ and $C_2 \in \mathcal{C}_2$ such that $B = C_1 \cap C_2$. Define relations r_1 and r_2 such that for all $x \in \Sigma^*$,

$$r_1(x) = \{y : |y| \le p(|x|) \wedge \langle x, y\rangle \in C_1\}$$

and

$$r_2(x) = \{y : |y| \le p(|x|) \wedge \langle x, y\rangle \in C_2\}.$$

Clearly $r_1 \in \mathrm{rel} \cdot \mathcal{C}_1$ and $r_2 \in \mathrm{rel} \cdot \mathcal{C}_2$.

It follows that for all $x \in \Sigma^*$, $r(x) = r_1(x) \cap r_2(x)$ and thus $r = r_1 \cap r_2$. This shows $r \in \mathrm{rel} \cdot \mathcal{C}_1 \wedge \mathrm{rel} \cdot \mathcal{C}_2$.

Now let $r \in \mathrm{rel} \cdot \mathcal{C}_1 \wedge \mathrm{rel} \cdot \mathcal{C}_2$. Hence there exist relations $s_1 \in \mathrm{rel} \cdot \mathcal{C}_1$ and $s_2 \in \mathrm{rel} \cdot \mathcal{C}_2$ such that $r = s_1 \cap s_2$. Let $D_1 \in \mathcal{C}_1$, $D_2 \in \mathcal{C}_2$, and $p_1, p_2 \in \mathrm{Pol}$ such that for all $x \in \Sigma^*$,

$$s_1(x) = \{y : |y| \le p_1(|x|) \wedge \langle x, y\rangle \in D_1\}$$

and

$$s_2(x) = \{y : |y| \le p_2(|x|) \wedge \langle x, y\rangle \in D_2\}.$$

Define

$$D_1' = \{\langle x, y\rangle : |y| \le \min\{p_1(|x|), p_2(|x|)\} \wedge \langle x, y\rangle \in D_1\}.$$

Since $\mathcal{C}_1$ is closed under $\le_{\mathrm{m}}^{\mathrm{p}}$ reductions we have $D_1' \in \mathcal{C}_1$. Let q be a polynomial such that $q(n) \ge \max\{p_1(n), p_2(n)\}$. Note that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le q(|x|) \wedge \langle x, y\rangle \in D_1' \cap D_2\}.$$

Hence $r \in \mathrm{rel} \cdot (\mathcal{C}_1 \wedge \mathcal{C}_2)$.

(2) can be shown quite similarly to (1).

(3). Let $r \in \mathrm{rel} \cdot (\mathcal{C}_1 \cap \mathcal{C}_2)$. Hence there exist a set $B \in \mathcal{C}_1 \cap \mathcal{C}_2$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le p(|x|) \wedge \langle x, y\rangle \in B\}.$$

It follows that $r \in \mathrm{rel} \cdot \mathcal{C}_1$ and $r \in \mathrm{rel} \cdot \mathcal{C}_2$ via B and p.

Now let $r \in \mathrm{rel} \cdot \mathcal{C}_1 \cap \mathrm{rel} \cdot \mathcal{C}_2$. Let $C_1 \in \mathcal{C}_1$, $C_2 \in \mathcal{C}_2$, and p_1, p_2 be polynomials such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le p_1(|x|) \ \wedge \ \langle x, y\rangle \in C_1\}$$

and

$$r(x) = \{y : |y| \le p_2(|x|) \ \wedge \ \langle x, y\rangle \in C_2\}.$$

Define

$$B = C_1 \cap \{\langle x, y\rangle : |y| \le \min\{p_1(|x|), p_2(|x|)\}\}.$$

Note that

$$B = C_2 \cap \{\langle x, y\rangle : |y| \le \min\{p_1(|x|), p_2(|x|)\}\}.$$

Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed under $\le_{\mathrm{m}}^{\mathrm{p}}$ reductions we conclude $B \in \mathcal{C}_1 \cap \mathcal{C}_2$. Let p be a polynomial such that $p(n) \ge \min\{p_1(n), p_2(n)\}$ for all n. It follows that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

This shows $r \in \mathrm{rel} \cdot (\mathcal{C}_1 \cap \mathcal{C}_2)$.

(4) can be shown quite similar to (3).

(5). Let $r \in \mathrm{rel} \cdot (\mathrm{co}\mathcal{C})$. Hence there exist a set $D \in \mathrm{co}\mathcal{C}$ and a polynomial q such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \le q(|x|) \ \wedge \ \langle x, y\rangle \in D\}.$$

Hence, for all $x \in \Sigma^*$,

$$r(x) = \Sigma^{\le q(|x|)} - \{y : |y| \le q(|x|) \ \wedge \ \langle x, y\rangle \in \overline{D}\}.$$

Since $\overline{D} \in \mathcal{C}$ we obtain $r \in \mathrm{co}(\mathrm{rel} \cdot \mathcal{C})$.

Now suppose $r \in \mathrm{co}(\mathrm{rel} \cdot \mathcal{C})$. Hence there exist a relation $s \in \mathrm{rel} \cdot \mathcal{C}$ and a polynomial q such that for all $x \in \Sigma^*$,

$$r(x) = \Sigma^{\le q(|x|)} - s(x).$$

It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$s(x) = \{y : |y| \le p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

Hence, for all $x \in \Sigma^*$,

$$r(x) = \Sigma^{\leq q(|x|)} - \{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}.$$

Define

$$D = \{\langle x, y\rangle : |y| \leq \min\{p(|x|), q(|x|)\} \ \wedge \ \langle x, y\rangle \notin B\} \cup \{\langle x, y\rangle : p(|x|) \leq |y| \leq q(|x|)\}.$$

Note that $D \in \mathrm{co}\mathcal{C}$ since $\mathcal{C}$ and thus also $\mathrm{co}\mathcal{C}$ are closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. It follows that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq q(|x|) \ \wedge \ \langle x, y\rangle \in D\}.$$

Hence $r \in \mathrm{rel} \cdot (\mathrm{co}\mathcal{C})$. ❑

The above theorem shows that set theoretic operators and the operator rel can be interchanged. It follows that the difference hierarchy over NPMV as defined in [FHOS97] is nothing but the “rel” equivalent of the boolean hierarchy over NP.

Corollary 3.5.2 *For all* $k \in \mathbb{N}^+, \mathrm{NPMV}(k) = \mathrm{rel} \cdot (\mathrm{BH}_k)$.

Applying Theorem 3.4.1 we obtain:

Corollary 3.5.3

(1) [FHOS97] For all $k \in \mathbb{N}^+$, $\mathrm{rel} \cdot (\mathrm{BH}_k) = \mathrm{rel} \cdot (\mathrm{BH}_{k+1})$ *if and only if* $\mathrm{BH}_k = \mathrm{BH}_{k+1}$.

(2) [FHOS97] $\mathrm{co}(\mathrm{rel} \cdot \mathrm{coNP}) = \mathrm{rel} \cdot \mathrm{NP}$.

We will now turn to the operators $\mathcal{U}$, Sig, SIG, $\mathrm{C}_{\geq}$, $\mathrm{C}_{=}$, $\mathrm{C}_{\leq}$ and $\oplus$.

Proposition 3.5.4 *For* $k \in \mathbb{N}^+$ *and for every* $\mathrm{op} \in \{\mathcal{U}, \mathrm{Sig}, \mathrm{SIG}, \mathrm{C}_{\geq}, \mathrm{C}_{=}, \mathrm{C}_{\leq}, \oplus\}$,

$$\mathrm{op} \cdot \mathrm{F}\Delta_k^{\mathrm{p}} = \Delta_k^{\mathrm{p}}.$$

The proof is obvious and thus omitted.

The following results can be found in [Hem03].

Theorem 3.5.5 *[Hem03] Let $\mathcal{C}$ be a complexity class being closed under $\leq_m^p$ and $\leq_{ctt}^p$ reductions.*

(1) $\mathcal{U} \cdot \min \cdot \mathcal{C} = \mathcal{U} \cdot \min_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C}$.

(2) $\mathrm{C}_{\geq} \cdot \min_t \cdot \mathcal{C} = \forall \cdot \mathrm{co}\mathcal{C}$.

(3) $\mathrm{C}_{\geq} \cdot \min \cdot \mathcal{C} = \forall \cdot \mathrm{co}\mathcal{C} \wedge \exists \cdot \mathcal{C}$.

(4) $\mathrm{Sig} \cdot \min_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C}$.

(5) $\mathrm{Sig} \cdot \min \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \wedge \exists \cdot \mathcal{C}$.

(6) $\mathrm{C}_{=} \cdot \min_t \cdot \mathcal{C} = \mathrm{C}_{=} \cdot \min \cdot \mathcal{C} = \mathcal{C} \wedge \forall \cdot \mathrm{co}\mathcal{C}$.

(7) $\oplus \cdot \min \cdot \mathcal{C} = \oplus \cdot \min_t \cdot \mathcal{C} = \mathrm{P}^{\exists \cdot \mathcal{C}}$.

(8) $\mathcal{U} \cdot \max \cdot \mathcal{C} = \mathcal{U} \cdot \max_t \cdot \mathcal{C} = \mathcal{C}$.

(9) $\mathrm{C}_{\geq} \cdot \max \cdot \mathcal{C} = \mathrm{C}_{\geq} \cdot \max_t \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(10) $\mathrm{Sig} \cdot \max \cdot \mathcal{C} = \mathrm{Sig} \cdot \max_t \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(11) $\mathrm{C}_{=} \cdot \max \cdot \mathcal{C} = \mathrm{C}_{=} \cdot \max_t \cdot \mathcal{C} = \mathcal{C} \wedge \forall \cdot \mathrm{co}\mathcal{C}$.

(12) $\oplus \cdot \max \cdot \mathcal{C} = \oplus \cdot \max_t \cdot \mathcal{C} = \mathrm{P}^{\exists \cdot \mathcal{C}}$.

The results for the operators SIG and $\mathrm{C}_{\leq}$ on max-classes are the same as for the operators Sig and $\mathrm{C}_{\geq}$ on min-classes, respectively, and vice versa.

Lemma 3.5.6

(1) $\mathrm{SIG} \cdot \min_t \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(2) $\mathrm{SIG} \cdot \min \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(3) $\mathrm{SIG} \cdot \max_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C}$.

(4) $\mathrm{SIG} \cdot \max \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \wedge \exists \cdot \mathcal{C}$.

(5) $\mathrm{C}_{\leq} \cdot \min_t \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(6) $\mathrm{C}_{\leq} \cdot \min \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(7) $\mathrm{C}_{\leq} \cdot \max_t \cdot \mathcal{C} = \forall \cdot \mathrm{co}\mathcal{C}$.

(8) $\mathrm{C}_{\leq} \cdot \max \cdot \mathcal{C} = \exists \cdot \mathcal{C} \wedge \forall \cdot \mathrm{co}\mathcal{C}$.

The proof is analogous to the proof of Theorem 3.5.5 which can be found in [Hem03] and is thus omitted.

If we apply operators to classes of total functions we get the following results:

Theorem 3.5.7 *Let $\mathcal{C}$ be a complexity class closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions and union.*

(1) $\mathcal{U} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathcal{U} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathcal{C} \cap \mathrm{co}\mathcal{C}$.

(2) $\mathrm{C}_{\geq} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{C}_{\leq} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

(3) $\mathrm{Sig} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{SIG} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \cap \mathrm{U} \cdot \mathcal{C}$.

(4) $\mathrm{C}_{=} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

(5) $\oplus \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

Proof (1). Let $A \in \mathcal{U} \cdot \mathrm{fun} \cdot \mathcal{C}$. Hence $c_A \in \mathrm{fun} \cdot \mathcal{C}$ or equivalently $c_A \in \mathrm{fun}_t \cdot \mathcal{C}$. It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$\begin{aligned} ||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| &\leq 1, \\ x \in A \iff \langle x, 1\rangle &\in B, \qquad \text{and} \\ x \notin A \iff \langle x, 0\rangle &\in B. \end{aligned}$$

It follows $A \in \mathcal{C} \cap \mathrm{co}\mathcal{C}$.

Now let $A \in \mathcal{C} \cap \mathrm{co}\mathcal{C}$. Define

$$B = \{\langle x, 1\rangle : x \in A\} \cup \{\langle x, 0\rangle : x \in \overline{A}\}.$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions and under union, it follows $c_A \in \mathrm{fun}_t \cdot \mathcal{C}$ via the set $B \in \mathcal{C}$ (and the polynomial $p(n) \equiv 1$).

(2). We prove only the equation $\mathrm{C}_{\geq} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$. The proof of the other one is analogous.

Let $A \in \mathrm{C}_{\geq} \cdot \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exist functions $f \in \mathrm{fun}_t \cdot \mathcal{C}$ and $g \in \mathrm{FP}_t$ such that for all $x \in \Sigma^*$, $x \in A \iff f(x) \geq_{\mathrm{lex}} g(x)$.

Define

$$D = \{\langle x, y\rangle : y \geq_{\mathrm{lex}} g(x) \ \wedge \ \langle x, y\rangle \in f\}$$

and

$$E = \{\langle x, y\rangle : y <_{\mathrm{lex}} g(x) \ \wedge \ \langle x, y\rangle \in f\}.$$

Note that for all $x \in \Sigma^*$, on the one hand,

$$\begin{aligned} ||\{y : \langle x, y\rangle \in D\}|| &\leq 1, \qquad \text{and} \\ ||\{y : \langle x, y\rangle \in E\}|| &\leq 1, \end{aligned}$$

and on the other hand,

$$x \in A \iff ||\{y : \langle x, y\rangle \in D\}|| = 1, \qquad \text{and}$$
$$x \notin A \iff ||\{y : \langle x, y\rangle \in E\}|| = 1.$$

Clearly, $D, E \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. It follows that $A \in \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

Now suppose that $A \in \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$. Hence there exist sets $B, D \in \mathcal{C}$ and polynomials p and q such that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| \leq 1,$$
$$||\{y : |y| \leq q(|x|) \ \wedge \ \langle x, y\rangle \in D\}|| \leq 1,$$

and

$$x \in A \iff ||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1, \qquad \text{and}$$
$$x \notin A \iff ||\{y : |y| \leq q(|x|) \ \wedge \ \langle x, y\rangle \in D\}|| = 1.$$

Define

$$B' = B \cap \{\langle x, y\rangle : |y| \leq p(|x|)\}, \qquad \text{and}$$
$$D' = D \cap \{\langle x, y\rangle : |y| \leq q(|x|)\}.$$

Note that $B', D' \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$. Let

$$B'' = \{\langle x, y1^{q(|x|)+1}\rangle : \langle x, y\rangle \in B'\}$$

and observe that $B'' \in \mathcal{C}$.

Let r be a polynomial such that $r(n) \geq \max\{p(n), q(n)\}$ for all n. Define $E = B'' \cup D'$ and note that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq r(|x|) \ \wedge \ \langle x, y\rangle \in E\}|| = 1.$$

Define a function f such that $f(x)$ is the unique string y such that $|y| \leq r(|x|)$ and $\langle x, y\rangle \in E$. Clearly, $E \in \mathcal{C}$ and $f \in \mathrm{fun}_t \cdot \mathcal{C}$. Now observe that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \geq_{\mathrm{lex}} 1^{q(|x|)+1}.$$

It follows that $A \in \mathrm{C}_{\geq} \cdot \mathrm{fun}_t \cdot \mathcal{C}$.

(3). We prove only the equation $\mathrm{Sig} \cdot \mathrm{fun}_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \cap \mathrm{U} \cdot \mathcal{C}$. The proof of the other one is analogous.

Let $A \in \mathrm{Sig} \cdot \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exists a function $f \in \mathrm{fun}_t \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \in \Sigma^* - \{\varepsilon\}.$$

It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$\begin{aligned} ||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1, \\ x \in A \implies \langle x, \varepsilon\rangle \notin B, \qquad \text{and} \\ x \notin A \implies \langle x, \varepsilon\rangle \in B. \end{aligned}$$

Hence, for all $x \in \Sigma^*$,

$$\begin{aligned} x \in A &\iff \langle x, \varepsilon\rangle \notin B, \qquad \text{and also} \\ x \in A &\iff (\exists y : y \neq \varepsilon)[\langle x, y\rangle \in B]. \end{aligned}$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions we have $A \in \mathrm{co}\mathcal{C}$. From the fact that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1,$$

it follows that $A \in \mathrm{U} \cdot \mathcal{C}$.

Now let $A \in \mathrm{co}\mathcal{C} \cap \mathrm{U} \cdot \mathcal{C}$. Hence there exist a set $B \in \mathcal{C}$ and a polynomial p witnessing $A \in \mathrm{U} \cdot \mathcal{C}$. Define

$$B' = \{\langle x, 1y\rangle : \langle x, y\rangle \in B\}.$$

Clearly, $B' \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. Define

$$B'' = \{\langle x, \varepsilon\rangle : x \notin A\} \cup B'.$$

Note that $B'' \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions and union. Observe that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B''\}|| = 1.$$

Define a function f such that

$$f = \{\langle x, y\rangle : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B''\}.$$

Clearly, f is polynomially length-bounded and thus $f \in \mathrm{fun}_t \cdot \mathcal{C}$. It follows that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \in \Sigma^* - \{\varepsilon\},$$

and thus $A \in \mathrm{Sig} \cdot \mathrm{fun}_t \cdot \mathcal{C}$.

(4). Let $A \in \mathrm{C}_= \cdot \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exist functions $g \in \mathrm{FP}_t$ and $f \in \mathrm{fun}_t \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in A \iff f(x) = g(x)$. It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$
\begin{aligned}
&||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1, \\
&x \in A \iff \langle x, g(x)\rangle \in B \qquad \text{and} \\
&x \in A \iff \neg(\exists y : |y| \leq p(|x|) \ \wedge \ y \neq g(x))[\langle x, y\rangle \in B].
\end{aligned}
$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions we have $A \in \mathcal{C}$. Since for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1,$$

we have $A \in \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

Now let $A \in \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$. Hence $\overline{A} \in \mathrm{co}\mathcal{C} \cap \mathrm{U} \cdot \mathcal{C}$. By Claim 2 we have $\overline{A} \in \mathrm{Sig} \cdot \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exists a function $f \in \mathrm{fun}_t \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$,

$$
\begin{aligned}
&x \in \overline{A} \iff f(x) >_{\mathrm{lex}} \varepsilon \qquad \text{or equivalently} \\
&x \in A \iff f(x) = \varepsilon.
\end{aligned}
$$

This shows $A \in \mathrm{C}_= \cdot \mathrm{fun}_t \cdot \mathcal{C}$.

(5). Let $A \in \oplus \cdot \mathrm{fun}_t \cdot \mathcal{C}$. Hence there exist functions $f \in \mathrm{fun}_t \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in A \iff \mathrm{lsb}(f(x)) = 1$.

Define

$$
\begin{aligned}
D &= \{\langle x, y\rangle : \mathrm{lsb}(y) = 1 \ \wedge \ \langle x, y\rangle \in f\} \qquad \text{and} \\
E &= \{\langle x, y\rangle : \mathrm{lsb}(y) = 0 \ \wedge \ \langle x, y\rangle \in f\}.
\end{aligned}
$$

Note that for all $x \in \Sigma^*$,

$$
\begin{aligned}
&||\{y : \langle x, y\rangle \in D\}|| \leq 1, \\
&||\{y : \langle x, y\rangle \in E\}|| \leq 1,
\end{aligned}
$$

and

$$
\begin{aligned}
x \in A &\iff ||\{y : \langle x, y\rangle \in D\}|| = 1, \\
x \notin A &\iff ||\{y : \langle x, y\rangle \in E\}|| = 1.
\end{aligned}
$$

Clearly, $D, E \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions. It follows that $A \in \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$.

Now suppose that $A \in \mathrm{U} \cdot \mathcal{C} \cap \mathrm{co}(\mathrm{U} \cdot \mathcal{C})$. Hence there exist sets $B, D \in \mathcal{C}$ and polynomials p and q such that for all $x \in \Sigma^*$,

$$\begin{aligned} &||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| \leq 1, \\ &||\{y : |y| \leq q(|x|) \ \wedge \ \langle x, y\rangle \in D\}|| \leq 1, \end{aligned}$$

and

$$\begin{aligned} x \in A &\iff ||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y\rangle \in B\}|| = 1, \\ x \notin A &\iff ||\{y : |y| \leq q(|x|) \ \wedge \ \langle x, y\rangle \in D\}|| = 1. \end{aligned}$$

Define

$$\begin{aligned} B' &= B \cap \{\langle x, y\rangle : |y| \leq p(|x|)\} \qquad \text{and} \\ D' &= D \cap \{\langle x, y\rangle : |y| \leq q(|x|)\}. \end{aligned}$$

Note that $B', D' \in \mathcal{C}$ since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$. Let

$$\begin{aligned} B'' &= \{\langle x, y1\rangle : \langle x, y\rangle \in B'\} \qquad \text{and} \\ D'' &= \{\langle x, y0\rangle : \langle x, y\rangle \in D'\}. \end{aligned}$$

Observe that $B'', D'' \in \mathcal{C}$.

Let r be a polynomial such that $r(n) \geq \max\{p(n), q(n)\}$ for all n. Define

$$E = B'' \cup D''$$

and note that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq r(|x|) \ \wedge \ \langle x, y\rangle \in E\}|| = 1.$$

Define a function f such that $f(x)$ is the unique string y such that $|y| \leq r(|x|)$ and $\langle x, y\rangle \in E$. Clearly, $E \in \mathcal{C}$ and $f \in \mathrm{fun}_t \cdot \mathcal{C}$. Now observe that for all $x \in \Sigma^*$,

$$x \in A \iff \mathrm{lsb}(f(x)) = 1.$$

It follows that $A \in \oplus \cdot \mathrm{fun}_t \cdot \mathcal{C}$. ❑

Similar results can be shown for classes of partial functions.

Theorem 3.5.8 *Let $\mathcal{C}$ be a complexity class closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions and union.*

(1) $\mathrm{C}_{\geq} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathrm{C}_{\leq} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C}$.

(2) $\mathrm{Sig} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathrm{SIG} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C}$.

(3) $\oplus \cdot \mathrm{fun} \cdot \mathcal{C} = \mathrm{U} \cdot \mathcal{C}$.

(4) $\mathrm{C}_{=} \cdot \mathrm{fun} \cdot \mathcal{C} = \mathcal{C}$.

Proof (1). Let $A \in \mathrm{C}_{\geq} \cdot \mathrm{fun} \cdot \mathcal{C}$. Hence there exist functions $g \in \mathrm{FP}_t$ and $f \in \mathrm{fun} \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \geq_{\mathrm{lex}} g(x).$$

It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}|| \leq 1 \quad \text{and}$$
$$x \in A \iff (\exists y : y \geq_{\mathrm{lex}} g(x) \wedge |y| \leq p(|x|))[\langle x, y\rangle \in B].$$

This shows that $A \in \mathrm{U} \cdot \mathcal{C}$.

Now let $A \in \mathrm{U} \cdot \mathcal{C}$. Hence there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}|| \leq 1 \quad \text{and}$$
$$x \in A \iff (\exists y : |y| \leq p(|x|))[\langle x, y\rangle \in B].$$

Define

$$f = \{\langle x, y\rangle : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}.$$

Clearly, f is polynomially length bounded and thus $f \in \mathrm{fun} \cdot \mathcal{C}$. Furthermore, for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \geq_{\mathrm{lex}} \varepsilon.$$

Hence $A \in \mathrm{C}_{\geq} \cdot \mathrm{fun} \cdot \mathcal{C}$.

The other equality from (1) and the equalities (2) and (3) can be shown quite similarly.

(4). Let $A \in \mathrm{C}_{=} \cdot \mathrm{fun} \cdot \mathcal{C}$. Hence there exist functions $g \in \mathrm{FP}_t$ and $f \in \mathrm{fun} \cdot \mathcal{C}$ such that for all $x \in \Sigma^*$,

$$x \in A \iff f(x) = g(x).$$

It follows that there exist a set $B \in \mathcal{C}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$||\{y : |y| \leq p(|x|) \ \wedge \ \langle x, y \rangle \in B\}|| \leq 1 \qquad \text{and}$$
$$x \in A \iff \langle x, g(x) \rangle \in B.$$

Since $\mathcal{C}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions we have $A \in \mathcal{C}$.

Now let $A \in \mathcal{C}$. Define

$$f = \{\langle x, 1 \rangle : x \in A\}.$$

Clearly, $f \in \mathcal{C}$ and f is polynomially length bounded via the polynomial $p(n) \equiv 1$. Hence $f \in \mathrm{fun} \cdot \mathcal{C}$ and for all $x \in \Sigma^*$,

$$x \in A \iff f(x) = 1.$$

Thus $A \in \mathrm{C}_= \cdot \mathrm{fun} \cdot \mathcal{C}$. ❑

Similar results can be shown for classes of relations.

Theorem 3.5.9 *Let $\mathcal{C}$ be a complexity class closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ reductions.*

(1) $\mathcal{U} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \mathcal{C} \cap \mathrm{co}\mathcal{C}$.

(2) $\mathrm{C}_{\geq} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \exists \cdot \mathcal{C} \cap \forall \cdot \mathrm{co}\mathcal{C}$.

(3) $\mathrm{C}_{=} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \mathcal{C} \cap \forall \cdot \mathrm{co}\mathcal{C}$.

(4) $\mathrm{C}_{\leq} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \exists \cdot \mathcal{C} \cap \forall \cdot \mathrm{co}\mathcal{C}$.

(5) $\mathrm{Sig} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \cap \exists \cdot \mathcal{C}$.

(6) $\mathrm{SIG} \cdot \mathrm{rel}_t \cdot \mathcal{C} = \mathrm{co}\mathcal{C} \cap \exists \cdot \mathcal{C}$.

(7) $\oplus \cdot \mathrm{rel}_t \cdot \mathcal{C} = \exists \cdot \mathcal{C} \cap \forall \cdot \mathrm{co}\mathcal{C}$.

(7) $\mathcal{U} \cdot \mathrm{rel} \cdot \mathcal{C} = \mathcal{C} \cap \mathrm{co}\mathcal{C}$.

(8) $\mathrm{C}_{\geq} \cdot \mathrm{rel} \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(9) $\mathrm{C}_{=} \cdot \mathrm{rel} \cdot \mathcal{C} = \mathcal{C}$.

(10) $\mathrm{C}_{\leq} \cdot \mathrm{rel} \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(11) $\mathrm{Sig} \cdot \mathrm{rel} \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(12) $\mathrm{SIG} \cdot \mathrm{rel} \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

(13) $\oplus \cdot \mathrm{rel} \cdot \mathcal{C} = \exists \cdot \mathcal{C}$.

The proof is similar to the last two proofs and thus omitted.

3.6 The Inclusion Structure and Structural Consequences

In this section we show that we can use the results of the previous section to derive structural consequences for unlikely inclusions between classes of functions or relations.

Observation 3.6.1 *For any two relation classes $\mathcal{R}_1$ and $\mathcal{R}_2$ and any operator $op \in \{\mathcal{U}, \mathrm{Sig}, \mathrm{SIG}, \mathrm{C}_{\geq}, \mathrm{C}_{=}, \mathrm{C}_{\leq}, \oplus\}$ we have $\mathcal{R}_1 \subseteq \mathcal{R}_2 \implies op \cdot \mathcal{R}_1 \subseteq op \cdot \mathcal{R}_2$.*

While this observation is immediate from the fact that all operators $\mathcal{U}$, $\mathrm{C}_{\geq}$, $\mathrm{C}_{=}$, $\mathrm{C}_{\leq}$, Sig, SIG, and $\oplus$ are monotone with respect to set inclusion, we are able to apply the operator method to derive structural consequences for hypotheses like $\mathcal{R}_1 \subseteq_{\mathrm{c}} \mathcal{R}_2$ instead of $\mathcal{R}_1 \subseteq \mathcal{R}_2$ as well.

Theorem 3.6.2 *For any two relation classes $\mathcal{R}_1$ and $\mathcal{R}_2$ and any operator $op \in \{\mathcal{U}, \mathrm{Sig}, \mathrm{SIG}, \mathrm{C}_{\geq}, \mathrm{C}_{=}, \mathrm{C}_{\leq}, \oplus\}$ we have $\mathcal{R}_1 \subseteq_{\mathrm{c}} \mathcal{R}_2 \implies op \cdot \mathcal{R}_1 \subseteq op \cdot \mathcal{R}_2$.*

Proof Let $\mathcal{R}_1$ and $\mathcal{R}_2$ are two classes of relations, and $op = \mathrm{C}_{\geq}$. (The proof for the other operators is similar.) Suppose that $\mathcal{R}_1 \subseteq_{\mathrm{c}} \mathcal{R}_2$ and let $A \in \mathrm{C}_{\geq} \cdot \mathcal{R}_1$. Hence there exist a relation $r_1 \in \mathcal{R}_1$ and a function $g \in \mathrm{FP}_t$ such that for all refinements f of r_1 where f is a function, and for all $x \in \Sigma^*$,

$$x \in A \iff f(x) \geq_{\mathrm{lex}} g(x).$$

By our assumption $\mathcal{R}_1 \subseteq_{\mathrm{c}} \mathcal{R}_2$ we know that r_1 has a refinement r_2 in $\mathcal{R}_2$. Obviously, all refinements of r_2 are refinements of r_1. It follows that every refinement f' of r_2 where f' is a function is a refinement of r_1. Hence for all refinements f' of r_2 where f' is a function, and for all $x \in \Sigma^*$,

$$x \in A \iff f'(x) \geq_{\mathrm{lex}} g(x),$$

and thus $A \in \mathrm{C}_{\geq} \cdot \mathcal{R}_2$. ❏

Now we will make extensive use of the results from Sections 3.4 and 3.5 to completely reveal the inclusion structure of function classes that are based on the complexity classes P, NP and coNP.

Note that Figures 3.1 and 3.2 present the inclusion structure in form of Hasse-diagrams of the partial orders $\subseteq$ and $\subseteq_{\mathrm{c}}$. A few of the given results have been shown previously, $\mathrm{fun}_t \cdot \mathrm{NP} = \mathrm{FP}_t^{\mathrm{NP} \cap \mathrm{coNP}}$ was mentioned in [HHN+93], $\mathrm{rel} \cdot \mathrm{NP} \subseteq_{\mathrm{c}} \mathrm{FP}^{\mathrm{NP}}$ is already contained in [Sel96].

First we state straightforward corollaries that follow from the theorems proven in Section 3.4. Note that these corollaries contain only a partial list of consequences that follow from the theorems proven in Section 3.4. See Table 3.1 on page 51 for a complete summary.

Figure 3.1: The left part shows the inclusion structure of classes of total functions relative to each other and relative to classes of deterministically polynomial-time computable functions.
The right part shows the inclusion structure of classes of relations relative to each other and relative to classes of deterministically polynomial-time computable functions. The structure remains unchanged if every rel is replaced by fun or if every rel is replaced by rel_t.

Corollary 3.6.3

(1) $\mathrm{rel} \cdot \mathrm{P} \subseteq \mathrm{rel} \cdot \mathrm{NP} \cap \mathrm{rel} \cdot \mathrm{coNP} \subseteq \mathrm{rel} \cdot \mathrm{NP} \cup \mathrm{rel} \cdot \mathrm{coNP} \subseteq \mathrm{rel} \cdot \mathrm{DP} \subseteq \mathrm{rel} \cdot \mathrm{P}^{\mathrm{NP}}$.

(2) *[FHOS97]* $\mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{coNP} \iff \mathrm{NP} = \mathrm{coNP}$.

(3) $\mathrm{fun} \cdot \mathrm{P} \subseteq \mathrm{fun} \cdot \mathrm{NP} \cap \mathrm{fun} \cdot \mathrm{coNP} \subseteq \mathrm{fun} \cdot \mathrm{NP} \cup \mathrm{fun} \cdot \mathrm{coNP} \subseteq \mathrm{fun} \cdot \mathrm{P}^{\mathrm{NP}}$.

(4) $\mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{fun} \cdot \mathrm{coNP} \iff \mathrm{NP} = \mathrm{coNP}$.

(5) $\mathrm{fun} \cdot \mathrm{DP} \subseteq \mathrm{fun} \cdot \mathrm{coNP} \iff \mathrm{NP} = \mathrm{coNP}$.

(6) $\mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{fun} \cdot \mathrm{P} \iff \mathrm{P} = \mathrm{NP}$.

(7) $\mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{P} \iff \mathrm{P} = \mathrm{NP}$.

The corollary follows from Theorem 3.4.1.

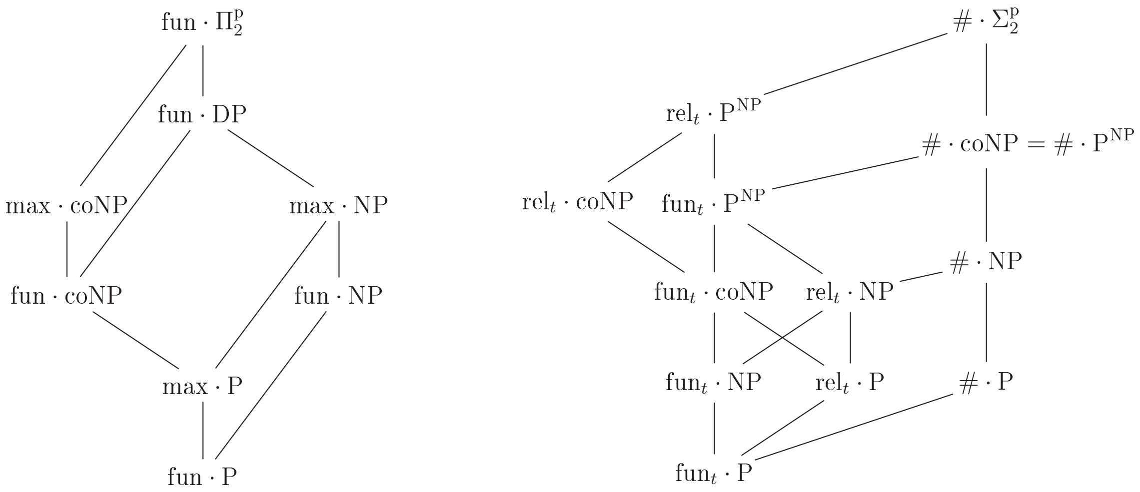


Figure 3.2: The left part shows the inclusion structure of classes of functions relative to each other and relative to classes of maximization functions. The structure remains unchanged if every max is replaced by min or every fun and every max is replaced by fun_t and $\max_t$, respectively.
The right part shows the inclusion structure of classes of total functions and relations relative to each other and relative to classes of counting functions.

Corollary 3.6.4

(1) $\mathrm{FP}_t \subseteq \mathrm{fun}_t \cdot \mathrm{P} \subseteq (\mathrm{FP}_t)_{||}^{\mathrm{UP}\cap\mathrm{coUP}}$.

(2) *[HHN+93]* $\mathrm{fun}_t \cdot \mathrm{NP} = \mathrm{FP}_t^{\mathrm{NP}\cap\mathrm{coNP}}$.

(3) $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP} \subseteq (\mathrm{FP}_t)_{||}^{\mathrm{UP}^{\mathrm{NP}}\cap\mathrm{coUP}^{\mathrm{NP}}}$.

(4) *[Sel96]* $\mathrm{rel} \cdot \mathrm{NP} \subseteq_c \mathrm{FP}^{\mathrm{NP}}$.

(5) $\mathrm{rel} \cdot \mathrm{coNP} \subseteq_c \mathrm{FP}^{\Sigma_2^p}$.

The claims of this corollary are straightforward consequences of Theorem 3.4.6, Theorem 3.4.8 and Theorem 3.4.9. For item (3) we use the previous mentioned fact that $\mathrm{U} \cdot \mathrm{coNP} = \mathrm{UP}^{\mathrm{NP}}$.

Corollary 3.6.5

(1) $\mathrm{fun} \cdot \mathrm{P} = \max \cdot \mathrm{P} \cap \min \cdot \mathrm{P}$.

(2) $\mathrm{fun} \cdot \mathrm{NP} = \max \cdot \mathrm{NP} \cap \min \cdot \mathrm{NP}$.

(3) $\mathrm{fun} \cdot \mathrm{coNP} = \max \cdot \mathrm{coNP} \cap \min \cdot \mathrm{coNP}$.

(4) $\max \cdot \mathrm{P} \subseteq \mathrm{fun} \cdot \mathrm{coNP}$.

(5) $\max \cdot \mathrm{NP} \subseteq \mathrm{fun} \cdot \mathrm{DP}$.

(6) $\max \cdot \mathrm{coNP} \subseteq \mathrm{fun} \cdot \Sigma_2^{\mathrm{p}}$.

(7) $\max \cdot \mathrm{DP} \subseteq \mathrm{fun} \cdot \Pi_2^{\mathrm{p}}$.

The claims follow immediately from Theorem 3.4.12. Item (4)–(7) remains true, if we replace the operator max by the operator min. Analogous results hold for the total versions of max and fun.

Corollary 3.6.6

(1) $\mathrm{fun}_t \cdot \mathrm{P} \subseteq \# \cdot \mathrm{P}$.

(2) $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \# \cdot \mathrm{NP}$.

(3) $\mathrm{fun}_t \cdot \mathrm{coNP} \subseteq \# \cdot \mathrm{P}^{\mathrm{NP}}$.

(4) $\mathrm{rel}_t \cdot \mathrm{NP} \subseteq \# \cdot \mathrm{NP}$

The corollary follows from Theorem 3.4.13.

The known inclusions as given in the previous corollaries are depicted in Figures 3.1 and 3.2.

All inclusions given are optimal unless some very unlikely complexity classes collapses occur. As examples we will state a few such structural consequences in the theorems below. Note that almost all results are immediate consequences of the Theorems 3.5.7, 3.5.8 and 3.5.9 obtained by applying the so-called operator method. (Observation 3.6.1 and Theorem 3.6.2)

A large number of inclusions hold if and only if NP = coNP. Note again, only some examples will be shown here. For a complete summary see Table 3.1 on page 51.

Theorem 3.6.7 *The following statements are pairwise equivalent:*

(1) $\mathrm{NP} = \mathrm{coNP}$

(2) $\mathrm{fun}_t \cdot \mathrm{coNP} \subseteq \mathrm{rel} \cdot \mathrm{NP}$

(3) $\mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{coNP}$

(4) $\mathrm{fun} \cdot \mathrm{DP} \subseteq \max \cdot \mathrm{NP}$

(5) $\mathrm{rel}_t \cdot \mathrm{NP} \subseteq \mathrm{rel}_t \cdot \mathrm{coNP}$

(6) $\mathrm{fun} \cdot \mathrm{coNP} \subseteq \max \cdot \mathrm{NP}$

(7) $\mathrm{rel} \cdot \mathrm{coNP} \subseteq_c \min \cdot \mathrm{NP}$

(8) *[FGH+96]* $\max \cdot \mathrm{NP} \subseteq \mathrm{fun} \cdot \mathrm{NP}$

(9) *[Sel94]* $\mathrm{FP}^{\mathrm{NP}} \subseteq \mathrm{rel} \cdot \mathrm{NP}$

(10) *[FGH+96]* $\mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{coNP}$

Proof To see that items (2), (3) and (10) imply NP = coNP we use Observation 3.6.1 with the operator $\mathrm{C}_=$. For item (4) we use Observation 3.6.1 with the operator $\mathcal{U}$. Item (5) can be seen as follows:

Suppose $\mathrm{rel}_t \cdot \mathrm{NP} \subseteq \mathrm{rel}_t \cdot \mathrm{coNP}$ and let $A \in \mathrm{NP}$ be a $\leq_m^p$-complete set. Define

$$r = \{\langle x, 1\rangle : x \in A\} \cup \{\langle x, 0\rangle : x \in \Sigma^*\}$$

and observe that $r \in \mathrm{rel}_t \cdot \mathrm{NP}$. Note that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in A &\implies r(x) = \{0,1\} \qquad \text{and} \\ x \notin A &\implies r(x) = \{0\}. \end{aligned}$$

By our assumption we conclude $r \in \mathrm{rel}_t \cdot \mathrm{coNP}$ and thus there exist a set $B \in \mathrm{NP}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$r(x) = \{y : |y| \leq p(|x|) \wedge \langle x, y\rangle \notin B\}.$$

It follows that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in A &\implies \langle x, 1\rangle \notin B \qquad \text{and} \\ x \notin A &\implies \langle x, 1\rangle \in B. \end{aligned}$$

Hence $A \leq_m^p \overline{B}$, implying NP = coNP. For item (6) we use Observation 3.6.1 with the operator $\mathrm{C}_{\geq}$, for item (7) the operator SIG and for item (8) and item (9) the operator $\oplus$.

It is not hard to see that NP = coNP implies all items. ❏

Theorem 3.6.8 *The following statements are pairwise equivalent:*

(1) $\mathrm{P} = \mathrm{NP}$

(2) *[Sel94]* $\mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{P}$

(3) $\mathrm{fun} \cdot \mathrm{NP} \subseteq \min \cdot \mathrm{P}$

(4) $\max \cdot \mathrm{P} \subseteq \mathrm{rel} \cdot \mathrm{P}$

(5) $\max \cdot \mathrm{P} \subseteq \mathrm{fun} \cdot \mathrm{P}$

(6) *[Sel94]* $\mathrm{rel} \cdot \mathrm{P} \subseteq_c \mathrm{FP}$

(7) *[Sel94]* $\mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{rel} \cdot \mathrm{P}$

(8) $\mathrm{rel} \cdot \mathrm{coNP} \subseteq \mathrm{rel} \cdot \mathrm{P}$

Proof To see that the inclusions (2), (4), (5), (7) and (8) imply $\mathrm{P} = \mathrm{NP}$ we use Observation 3.6.1 and Theorem 3.6.2, respectively with the operator $\mathrm{C}_=$.

Item (6) is done using the operator Sig.

For item (3) we have to use two operators. Using operator $\mathcal{U}$ we get the consequence $\mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{P}$. Using operator $\mathrm{C}_=$ we get the consequence $\mathrm{NP} = \mathrm{coNP}$. Both together give the consequence $\mathrm{P} = \mathrm{NP}$.

The other direction is easy to be seen. ❑

We can prove that some previously known results can be relativized using the operator method, at least in one direction.

Theorem 3.6.9

(1) *[GS88]* $\mathrm{fun} \cdot \mathrm{P} \subseteq \mathrm{FP} \iff \mathrm{P} = \mathrm{UP}$.

(2) $\mathrm{fun} \cdot \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{FP}^{\mathrm{NP}} \implies \mathrm{P}^{\mathrm{NP}} = \mathrm{UP}^{\mathrm{NP}}$.

(3) *[Sel94]* $\mathrm{rel} \cdot \mathrm{P} \subseteq_c \mathrm{FP} \iff \mathrm{P} = \mathrm{NP}$.

(4) $\mathrm{rel} \cdot \mathrm{P}^{\mathrm{NP}} \subseteq_c \mathrm{FP}^{\mathrm{NP}} \implies \mathrm{P}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}}$.

(5) *[Sel94]* $\mathrm{rel} \cdot \mathrm{P} \subseteq_c \mathrm{fun} \cdot \mathrm{P} \implies \mathrm{UP} = \mathrm{NP}$.

(6) $\mathrm{rel} \cdot \mathrm{P}^{\mathrm{NP}} \subseteq_c \mathrm{fun} \cdot \mathrm{P}^{\mathrm{NP}} \implies \mathrm{UP}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}}$.

Proof For the left-to-right implications we use the operator method applying the operator Sig.

The other directions of (1) and (3) are easy to be seen. ❑

Items (2), (4) and (6) can be strengthened as the following theorem shows.

Theorem 3.6.10

(1) $\mathrm{fun}\cdot\mathrm{coNP} \subseteq \mathrm{FP}^{\mathrm{NP}} \implies \mathrm{P}^{\mathrm{NP}} = \mathrm{UP}^{\mathrm{NP}}$.

(2) $\mathrm{rel}\cdot\mathrm{coNP} \subseteq_{\mathrm{c}} \mathrm{FP}^{\mathrm{NP}} \implies \mathrm{P}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}}$.

(3) $\mathrm{rel}\cdot\mathrm{coNP} \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\mathrm{P}^{\mathrm{NP}} \implies \mathrm{UP}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}}$.

Proof

All claims follow by applying the operator method, by applying the operator $\mathrm{C}_{\geq}$. ❑

Again, for a complete summary of such results see Table 3.1 on page 51.

3.7 Beyond the Operator Method

The operator method fails at some structural consequences for hypotheses like $\mathrm{rel}\cdot\mathrm{NP} \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\mathrm{NP}$. Selman proved that this is equivalent to $\mathrm{rel}\cdot\mathrm{P} \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\mathrm{NP}$. But we will obtain some consequences for such inclusions if we generalize an idea from [HNOS96]. They showed that $\mathrm{rel}\cdot\mathrm{NP} \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\mathrm{NP}$ implies a collapse of the polynomial hierarchy to the class $\mathrm{ZPP}^{\mathrm{NP}}$. This result was strengthened in [CCHO03] to a collapse to $\mathrm{S}_2^{\mathrm{NP}\cap\mathrm{coNP}}$. Theorem 3.7.1 remains true if we replace $\mathrm{ZPP}^{\Sigma^{\mathrm{p}}_{k+1}}$ by $\mathrm{S}_2^{\Sigma^{\mathrm{p}}_{k+1}\cap\Pi^{\mathrm{p}}_{k+1}}$ and $\mathrm{ZPP}^{\Sigma^{\mathrm{p}}_{k}}$ by $\mathrm{S}_2^{\Sigma^{\mathrm{p}}_{k}\cap\Pi^{\mathrm{p}}_{k}}$, respectively.

Theorem 3.7.1 *For all $k \in \mathbb{N}^+$,*

(1) $\mathrm{rel}\cdot\Pi^{\mathrm{p}}_k \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\Pi^{\mathrm{p}}_k \implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma^{\mathrm{p}}_{k+1}}$.

(2) $\mathrm{rel}\cdot\Sigma^{\mathrm{p}}_k \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\Sigma^{\mathrm{p}}_k \implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma^{\mathrm{p}}_{k}}$.

Proof

We show

$$\mathrm{rel}\cdot\Pi^{\mathrm{p}}_k \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\Pi^{\mathrm{p}}_k \implies \Sigma^{\mathrm{p}}_{k+1} \subseteq (\Sigma^{\mathrm{p}}_{k+1}\cap\Pi^{\mathrm{p}}_{k+1})/\mathrm{poly}.$$

Köbler and Watanabe [KW98] proved

$$\Sigma^{\mathrm{p}}_{k+1} \subseteq (\Sigma^{\mathrm{p}}_{k+1}\cap\Pi^{\mathrm{p}}_{k+1})/\mathrm{poly} \implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma^{\mathrm{p}}_{k+1}}.$$

Let $\mathrm{rel}\cdot\Pi^{\mathrm{p}}_k \subseteq_{\mathrm{c}} \mathrm{fun}\cdot\Pi^{\mathrm{p}}_k$ and $A \in \Sigma^{\mathrm{p}}_{k+1}$. We define a relation r by

$$r(\langle x,y\rangle) = \{z : (z = x \ \vee\ z = y) \ \wedge\ z \in A\}.$$

Obviously, it holds that $r \in \mathrm{rel} \cdot \Sigma^{\mathrm{p}}_{k+1}$. It follows that there exist a set $B \in \Pi^{\mathrm{p}}_k$ and a polynomial $p \in \mathrm{Pol}$ with

$$z \in r(\langle x, y\rangle) \iff (\exists u \in \Sigma^* : |u| \leq p(|\langle x, y\rangle|))[\langle z, u, x, y\rangle \in B].$$

We define another relation s by

$$s(\langle x, y\rangle) = \{z\#u : |u| \leq p(|\langle x, y\rangle|) \ \wedge\ \langle z, u, x, y\rangle \in B\}.$$

Since $B \in \Pi^{\mathrm{p}}_k$, it holds that $s \in \mathrm{rel} \cdot \Pi^{\mathrm{p}}_k$. Hence we have a refinement $f \in \mathrm{fun} \cdot \Pi^{\mathrm{p}}_k$ of the relation s. This function could be called a quasi-selector of A. We can define a graph $G = (\Sigma^n, E)$ for every $n \in \mathbb{N}$. A pair $(x, y) \in \Sigma^n \times \Sigma^n$ is an edge if and only if $f(\langle x, y\rangle)$ starts with x:

$$(x, y) \in E \iff (\exists u \in \Sigma^* : |u| \leq p(|\langle x, y\rangle|))[f(\langle x, y\rangle) = x\#u].$$

As in the case of Ko's proof that the P-selective sets are in P/poly [Ko83], we use a well-known theorem about tournament graphs. These graphs always have a dominating set of size logarithmic in the number of nodes. Hence there exists a set $D_{|x|} \subseteq A^{=|x|}$ with at most n elements satisfying

$$x \in A \iff (\exists y \in D_{|x|})(\exists u \in \Sigma^* : |u| \leq p(|\langle x, y\rangle|))[f(\langle x, y\rangle) = x\#u].$$

From $A \in \Sigma^{\mathrm{p}}_{k+1}$ it follows that there exists a set $C \in \Pi^{\mathrm{p}}_k$ and a polynomial $q \in \mathrm{Pol}$ satisfying

$$y \in A \iff (\exists z \in \Sigma^* : |z| \leq q(|y|))[\langle y, z\rangle \in C].$$

We define the set D as follows

$$\begin{aligned} B = \{\langle x, W, U\rangle : W \subseteq \Sigma^{|x|} \ \wedge\ ||W|| \leq |x| \ \wedge\ (\forall w \in W)(\exists z \in U)[\langle w, z\rangle \in C] \wedge \\ (\exists y \in W)(\exists u \in \Sigma^* : |u| \leq p(|\langle x, y\rangle|))[f(\langle x, y\rangle) = x\#u]\}. \end{aligned}$$

Since W has only $|x|$ elements, the for-all quantifier is harmless and we get $B \in \Sigma^{\mathrm{p}}_{k+1}$. Moreover, we will show that $\overline{B} \in \Sigma^{\mathrm{p}}_{k+1}$.

We can write $\overline{B}$ in the form

$$\begin{aligned} \overline{B} = \{\langle x, W, U\rangle : W \subseteq \Sigma^{|x|} \ \wedge\ ||W|| \leq |x| \ \wedge\ (\forall w \in W)(\exists z \in U)[\langle w, z\rangle \in C] \implies \\ (\forall y \in W)(\forall u \in \Sigma^* : |u| \leq p(|\langle x, y\rangle|))[f(\langle x, y\rangle) \neq x\#u]\}. \end{aligned} \tag{3.1}$$

If the hypothesis in the above inclusion is true, then it holds that $W \subseteq A$. But the quasi-selector f has the following property

$$\{x, y\} \cap A \neq \varnothing \implies f(\langle x, y\rangle) = x\#u \ \vee\ f(\langle x, y\rangle) = y\#v \text{ for some } u \text{ and } v.$$

So we can write equation (3.1) in the form

$$\overline{B} = \{\langle x, W, U\rangle : W \subseteq \Sigma^{|x|} \ \wedge \ ||W|| \leq |x| \ \wedge \ (\forall w \in W)(\exists z \in U)[\langle w, z\rangle \in C] \implies (\forall y \in W)(\exists v \in \Sigma^* : |v| \leq p(|\langle x, y\rangle|))[f(\langle x, y\rangle) = y\#v]\}.$$

This shows that $\overline{B} \in \Sigma^{\mathrm{p}}_{k+1}$ and hence $B \in \Sigma^{\mathrm{p}}_{k+1} \cap \Pi^{\mathrm{p}}_{k+1}$. Let $U_{|x|}$ be a set such that for every $w \in D_{|x|}$ there exists some z with $\langle w, z\rangle \in C$. We define the function h as $h(|x|) = (D_{|x|}, U_{|x|})$ and get the equivalence

$$x \in A \iff (x, h(|x|)) \in B.$$

This shows $A \in (\Sigma^{\mathrm{p}}_{k+1} \cap \Pi^{\mathrm{p}}_{k+1})/\mathrm{poly}$ and hence $\Sigma^{\mathrm{p}}_{k+1} \subseteq (\Sigma^{\mathrm{p}}_{k+1} \cap \Pi^{\mathrm{p}}_{k+1})/\mathrm{poly}$. This completes the proof of the first item.

The proof of the second item is analogous to this proof. ❏

3.8 Open Problems

We would like to find a structural consequence that follows from $\mathrm{fun}_t \cdot \mathrm{coNP} \subseteq \# \cdot \mathrm{NP}$. Note that $\min_t \cdot \mathrm{P} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP}$ follows from Theorem 3.4.12 part 2. Hence any structural consequence that follows from $\min_t \cdot \mathrm{P} \subseteq \# \cdot \mathrm{NP}$ immediately yields a structural consequence that follows from $\mathrm{fun}_t \cdot \mathrm{coNP} \subseteq \# \cdot \mathrm{NP}$. However no structural consequence that follows from $\min_t \cdot \mathrm{P} \subseteq \# \cdot \mathrm{NP}$ is known today. So proving a structural consequence that follows from $\mathrm{fun}_t \cdot \mathrm{coNP} \subseteq \# \cdot \mathrm{NP}$ is potentially easier.

Furthermore we want to fill out Table 3.1 completely, since there are some cells for which we have not been able to find structural equivalences.

	FP	fun · P	rel · P	max · P	min · P	fun · NP	rel · NP	max · NP	min · NP	fun · coNP	rel · coNP	max · coNP	min · coNP	FP^{NP}	fun · P^{NP}	rel · P^{NP}
FP	=	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
fun · P	P = UP	=	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
rel · P	P = NP	NP = UP	=	$\subseteq_c$	$\subseteq_c$	rel·NP $\subseteq_c$ fun·NP	$\subseteq$	$\subseteq_c$	$\subseteq_c$	$\subseteq_c$	$\subseteq$	$\subseteq_c$	$\subseteq_c$	$\subseteq_c$	$\subseteq_c$	$\subseteq$
max · P	P = NP	P = NP	P = NP	=	NP = coNP	NP = coNP	NP = coNP	$\subseteq$	NP = coNP	$\subseteq$	$\subseteq$	$\subseteq$	②	$\subseteq$	$\subseteq$	$\subseteq$
min · P	P = NP	P = NP	P = NP	NP = coNP	=	NP = coNP	NP = coNP	NP = coNP	$\subseteq$	$\subseteq$	$\subseteq$	②	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
fun · NP	P = NP	P = NP	P = NP	P = NP	P = NP	=	$\subseteq$	$\subseteq$	$\subseteq$	NP = coNP	NP = coNP	①	①	$\subseteq$	$\subseteq$	$\subseteq$
rel · NP	P = NP	P = NP	P = NP	P = NP	P = NP	$PH = ZPP^{NP}$	=	$\subseteq_c$	$\subseteq_c$	NP = coNP	NP = coNP	①	①	$\subseteq_c$	$\subseteq_c$	$\subseteq$
max · NP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	=	NP = coNP	NP = coNP	NP = coNP	NP = coNP	②	$\subseteq$	$\subseteq$	$\subseteq$
min · NP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	=	NP = coNP	NP = coNP	②	NP = coNP	$\subseteq$	$\subseteq$	$\subseteq$
fun · coNP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	=	$\subseteq$	$\subseteq$	$\subseteq$	$\Delta_2^P = UP^{NP}$	$\subseteq$	$\subseteq$
rel · coNP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	$PH = ZPP^{\Sigma_2^P}$	=	$\subseteq_c$	$\subseteq_c$	$\Delta_2^P = \Sigma_2^P$	$UP^{NP} = \Sigma_2^P$	$\subseteq$
max · coNP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	=	NP = coNP	$\Delta_2^P = \Sigma_2^P$	$UP^{NP} = \Sigma_2^P$	$\Delta_2^P = \Sigma_2^P$
min · coNP	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	=	$\Delta_2^P = \Sigma_2^P$	$UP^{NP} = \Sigma_2^P$	$\Delta_2^P = \Sigma_2^P$
FP^{NP}	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	=	$\subseteq$	$\subseteq$
fun · P^{NP}	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	$\Delta_2^P = UP^{NP}$	=	$\subseteq$
rel · P^{NP}	P = NP	P = NP	P = NP	P = NP	P = NP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	NP = coNP	$\Delta_2^P = \Sigma_2^P$	$UP^{NP} = \Sigma_2^P$	=

Table 3.1: The complete structure of the observed relation classes.

If a cell is filled by $\subseteq$ ($\subseteq_c$), the left-hand class is contained in the upper one (with respect to refinements). Otherwise the contents of a cell is equivalent to the condition that the left-hand class is contained in the upper one (perhaps with respect to refinements). Some cells have a gray background, which means that the contents is only a consequence of the fact that the left-hand class is contained in the upper one.

Note that the four inclusions marked by ① follow from NP = coNP. The inclusions marked by ② follow from P = NP. So do all other inclusions, too.

Chapter 4

Solution Relations

In this chapter we study so-called *easy*-languages. These are two kinds of languages: One is required to have easily computable solution relations for at least one corresponding NPTM, and the other must have easily computable solution relations for all corresponding NPTMs. If we speak of solution relation, we mean a relation that computes accepting paths of the corresponding NPTM. We analyze whether it makes a difference to have a solution relation or a relation that computes only one bit of a solution.

Furthermore we examine which languages can be accepted for a given class of solution relations. For this purpose we study the power of solution relations from classes as rel · NP, fun · NP or fun · UP.

4.1 Introduction

The analysis of the class NP is motivated by so-called projection and search problems.

The initial point is a *problem* $a \subseteq \Sigma^*$. For a given pair $\langle x, y \rangle$ we want to know if it is in a. If $\langle x, y \rangle \in a$ then y is called a *certificate* for x.

For example, let

$$\text{ham} = \{\langle G, p \rangle : p \text{ is a hamiltonian path in the finite graph } G\}.$$

In this case, certificates are hamiltonian paths.

We have a one-to-one relation between such problems and nondeterministic Turing machines in the following way:

$$\langle x, y \rangle \in a \iff M \text{ accepts } x \text{ along the path } y.$$

The question whether a given graph has a hamiltonian path is more frequent than the above one. It can be described using the concept of projection.

We call the language $A = \mathrm{proj}_1^2(a)$ the *projection problem* related to the problem a.

In our example: does a given graph have a hamiltonian path? Obviously every decision problem a has a related projection problem A. But there are many decision problems related to one and the same projection problem.

For practical applications the most interesting task is to compute a certificate y for a given x. This is the so-called *search problem*. To solve a search problem means to compute a *solution relation*.

Definition 4.1.1 *Let a be a problem, M a related nondeterministic Turing machine and $A = \mathrm{proj}_1^2(a)$ the projection of a.*
A relation r is called a weak solution relation *for A with respect to M if and only if*

$$x \in A \implies \varnothing \neq \; r(x) \subseteq \mathrm{acc}_M(x).$$
$$x \notin A \implies r(x) = \varnothing.$$

A relation r is called a strong solution relation *for A with respect to M if and only if*

$$x \in A \implies \varnothing \neq r(x) \subseteq \{1y : y \in \mathrm{acc}_M(x)\},$$
$$x \notin A \implies \varnothing \neq r(x) \subseteq \{0y : y \in \Sigma^*\}.$$

Note that a weak solution relation r for A with respect to M is a refinement of acc_M and evidently $\mathrm{dom}(r) = A$.

In the negative case of a strong solution relation, that is $x \notin A$, $r(x) = 0$ would be enough. For technical reasons we allow an arbitrary word starting with 0.

It is possible that there are uncountably many solution relations for one problem a, namely if there are infinitely many x with $||\{y : \langle x, y\rangle \in a\}|| \geq 2$.

We intuitively know that solving a projection problem is easier than solving the corresponding search problem, because knowing that a given graph has a hamiltonian path does not automatically yield a construction of such a path? On the other hand – if we have an algorithm to compute a solution relation, then we can solve the projection problem, too.

In practical applications, the computation of a solution relation is much more interesting than solving the projection problem. So it is an interesting question what the relationship between the complexity of solving the search problem and the projection problem is.

It is known that for self-reducible problems the corresponding search problem is Turing-reducible to the decision problem in polynomial time[BD76, Sch79]. This property is known as *search reduces to decision* (see for instance [HNOS93]).

But there is a negative result, too.

Borodin and Demers [BD76] proved the following result.

Theorem 4.1.2 *[BD76] If* $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP}$*, then there exists a set* A *such that*

(1) $A \in \mathrm{P}$

(2) $A \subseteq \mathrm{SAT}$*, and*

(3) there exists no function $f \in \mathrm{FP}$ *that computes a satisfying assignment for all* $F \in A$.

This can be rephrased as follows. Under the above hypothesis which most complexity theoreticians would assume to be true, it follows that there exist easily decidable sets, yet it is hard to compute why, i. e. it is hard to compute the corresponding solution relation.

This chapter is organized as follows. We analyze languages for which it is easy to compute (partial) certificates in Section 4.2. For this reason we distinguish between languages for which every or at least one NPTM has easy (partial) certificates. Further we examine which languages we get if we use solution relations from a given class of relations. For this purpose we define the operators wsol and ssol in Section 4.3 and study some of their properties. In Section 4.4 we apply these operators to relation classes as $\mathrm{rel} \cdot \mathrm{NP}$ and $\mathrm{fun} \cdot \mathrm{NP}$.

4.2 Easy Languages

In [HRW97] complexity classes of the following form were studied. They contain languages that have easily computable solution relations for either at least one or for all corresponding NPTMs.

We are as well interested in solution relations for which only a part – e.g. one bit – can easily be computed.

4.2.1 $\mathrm{Easy}_\forall$

We start with languages for which *every* related NPTM allows for an easy computation of the solution relation or parts of it. Therefor we define the following notations.

As introduced in [HRW97], we will say that an NPTM M *has easy certificates*, if for each $x \in L(M)$ some accepting path of $M(x)$ can be computed by a function $f \in \mathrm{FP}_t$. The class $\mathrm{Easy}_\forall$ is the set of all languages for which every accepting NPTM has easy certificates. If the n-th bit of an accepting path can be computed in polynomial time then the language is in $\mathrm{Easy}_\forall^{(n)}$.

This is stated more formally in Definition 4.2.1.

Definition 4.2.1 *Let $L \subseteq \Sigma^*$ be a set and $n \in \mathbb{N}^+$.*

(1) [HRW97] $L \in \mathrm{Easy}_\forall$ if and only if

(a) $L \in \mathrm{NP}$, and

(b) $(\forall NPTM\ M : L(M) = L)(\exists f \in \mathrm{FP}_t)(\forall x \in L)[f(x) \in \mathrm{acc}_M(x)]$

(2) $L \in \mathrm{Easy}_\forall^{(n)}$ if and only if

(a) $L \in \mathrm{NP}$, and

(b)

$$(\forall NPTM\ M : L(M) = L)(\exists f \in \mathrm{FP}_t)(\forall x \in L)(\exists u, v \in \Sigma^*) \Big[\Big(f(x) \in \{0,1\} \ \wedge\ |uf(x)| = n \ \wedge\ uf(x)v \in \mathrm{acc}_M(x)\Big) \vee \Big((\forall y \in \mathrm{acc}_M(x))[|y| < n]\Big)\Big]$$

Functions as used in Definition 4.2.1 are called solution functions. Traditionally, in the context of the class $\mathrm{Easy}_\forall$, the solution functions are considered to be total. This implies that $f(x)$ is an arbitrary path if $x \notin L$. The same note holds for Definition 4.2.7.

The following observation is a direct consequence of Definition 4.2.1.

Observation 4.2.2 *For all $n \in \mathbb{N}^+$ we have $\mathrm{FINITE} \subseteq \mathrm{Easy}_\forall \subseteq \mathrm{Easy}_\forall^{(n)} \subseteq \mathrm{NP}$.*

It is easy to see that $\mathrm{Easy}_\forall \subseteq \mathrm{P}$.

Obviously one of the inclusions in Observation 4.2.2 has to be a proper inclusion, since we know that $\mathrm{FINITE} \neq \mathrm{NP}$. For the class $\mathrm{Easy}_\forall$, many properties are known.

(1) [HRW97] $\mathrm{P} \neq \mathrm{NP} \iff \mathrm{Easy}_\forall \neq \mathrm{NP}$.

(2) [BD76] $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP} \implies \mathrm{Easy}_\forall \neq \mathrm{P}$.

(3) [FFNR96]

$$\begin{aligned} \mathrm{Easy}_\forall = \mathrm{P} &\iff \Sigma^* \in \mathrm{Easy}_\forall \\ &\iff \mathrm{rel}_t \cdot \mathrm{NP} \subseteq_c \mathrm{FP} \\ &\iff \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP} \ \wedge\ \mathrm{rel}_t \cdot \mathrm{NP} \subseteq_c \mathrm{fun}_t \cdot \mathrm{NP} \end{aligned}$$

Let us first to concentrate on the $\mathrm{Easy}_\forall^{(n)}$ classes. Is there in fact a difference between computing the first or the second bit of a solution? We can show that there is no difference.

Theorem 4.2.3 *For all* $n \in \mathbb{N}^+$, $\text{Easy}_\forall^{(1)} = \text{Easy}_\forall^{(n)}$.

Proof First we show that for all $n \in \mathbb{N}^+$ we have $\text{Easy}_\forall^{(n+1)} \subseteq \text{Easy}_\forall^{(n)}$.

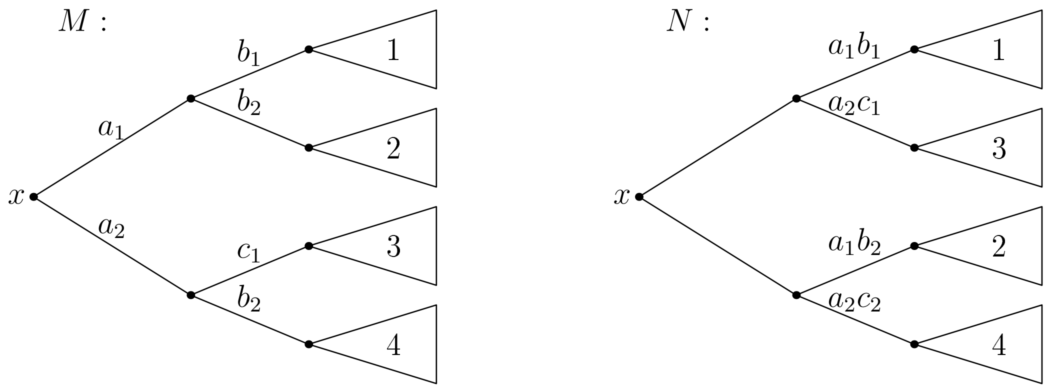
Let $L \in \text{Easy}_\forall^{(n+1)}$ and N be an NPTM with $L(N) = L$. We define an NPTM M so that on input x the machine M nondeterministically guesses one bit, and on each of the two branches it continues simulating N on input x. Then $L(M) = L$ and hence there exists a function $f \in \text{FP}_t$ that computes the $(n+1)$-th bit of an accepting path of M. Obviously this function computes the n-th bit of an accepting path of N. It follows that $L \in \text{Easy}_\forall^{(n)}$.

For the other direction we only show the case $\text{Easy}_\forall^{(1)} \subseteq \text{Easy}_\forall^{(2)}$. The general case is analogous.

Let $L \in \text{Easy}_\forall^{(1)}$ and M be an arbitrary NPTM with $L(M) = L$. We will now describe an NPTM N with the following properties:

(1) $L(N) = L$, and

(2) if there exists an accepting path in $M(x)$ whose second bit is 0 (or 1) then there is an accepting path in $N(x)$ whose first bit is 0 (or 1).



In its first step, the machine N guesses the second bit of an accepting path of the machine M. In the second step, N simulates the first and the second step of M. This is done in one step. (See the sketch above.) From the third step on, the machine N works as the machine M.

Obviously, $L(N) = L$ holds.

Since $L \in \text{Easy}_\forall^{(1)}$, there is a function $f \in \text{FP}_t$, that for every $x \in L$ computes the first bit of an accepting path of N. The rearrangement and the slight modifications of the computation tree of M ensure that this bit is the second bit of an accepting path of M. ❑

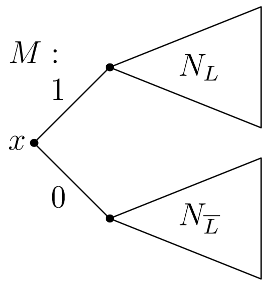
Since all classes $\mathrm{Easy}_\forall^{(n)}$ are equal, we denote these classes with $\mathrm{Easy}'_\forall$.

Obviously, all finite sets are in $\mathrm{Easy}'_\forall$, but are there infinite sets in $\mathrm{Easy}'_\forall$? The next lemma shows that even such simple sets as Σ^* are probably not in $\mathrm{Easy}'_\forall$, otherwise we would have the unlikely equality $\mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$. It also strengthens the implication $\Sigma^* \in \mathrm{Easy}_\forall \implies \mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$ that follows from the above mentioned results from [BD76] and [FFNR96].

Lemma 4.2.4 $\Sigma^* \in \mathrm{Easy}'_\forall \implies \mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$

Proof Suppose $\Sigma^* \in \mathrm{Easy}'_\forall$. It remains to show $\mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{P}$.

Let $L \in \mathrm{NP} \cap \mathrm{coNP}$ via NPTMs N_L and $N_{\overline{L}}$, that is, $L(N_L) = L$ and $L(N_{\overline{L}}) = \overline{L}$. We define an NPTM M so that on input x, M guesses which NPTM is simulated. To that effect M simulates N_L or $N_{\overline{L}}$ on input x. Then $L(M) = \Sigma^*$ and hence there is a function $f \in \mathrm{FP}_t$ that computes the first bit of an accepting path of M.



We have $(\forall x \in \Sigma^*)[x \in L \iff f(x) = 1]$ and hence $L \in \mathrm{P}$. ❑

We can conclude a simple corollary.

Corollary 4.2.5

$$\mathrm{P} \subseteq \mathrm{Easy}'_\forall \implies \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$$

From this corollary it follows that

$$\mathrm{NP} = \mathrm{Easy}'_\forall \implies \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}. \tag{4.1}$$

This strengthens the implication

$$\mathrm{NP} = \mathrm{Easy}_\forall \implies \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$$

from [BD76]. Implication (4.1) should also be compared to the equivalence

$$\mathrm{NP} = \mathrm{Easy}_\forall \iff \mathrm{P} = \mathrm{NP}$$

from [HRW97] as stated above.

Since such easy sets as Σ^* are probably not in $\mathrm{Easy}'_\forall$ we ask: Are there only finite sets in $\mathrm{Easy}'_\forall$? We do not know the answer. But we know the answer is as difficult as the question whether $\mathrm{P} \neq \mathrm{NP}$.

Lemma 4.2.6 $\mathrm{Easy}'_\forall = \mathrm{FINITE} \implies \mathrm{P} \neq \mathrm{NP}$

Proof From $\mathrm{Easy}'_\forall = \mathrm{FINITE}$ it follows that $\mathrm{Easy}_\forall = \mathrm{FINITE}$ and further we can conclude that $\mathrm{P} \not\subseteq \mathrm{Easy}_\forall$ and so we get $\mathrm{NP} \neq \mathrm{Easy}_\forall$. In [HRW97] it was shown that $\mathrm{NP} \neq \mathrm{Easy}_\forall$ is equivalent to $\mathrm{P} \neq \mathrm{NP}$. ❑

4.2.2 $\mathrm{Easy}_\exists$

For languages in $\mathrm{Easy}_\forall$, every corresponding NPTM must have easy certificates. We want to weaken this condition and now claim that only at least one corresponding NPTM has easy certificates. For this reason we define classes $\mathrm{Easy}_\exists$ and $\mathrm{Easy}_\exists^{(n)}$, analogously to the classes $\mathrm{Easy}_\forall$ and $\mathrm{Easy}_\forall^{(n)}$, respectively.

Definition 4.2.7 *Let $L \subseteq \Sigma^*$ be a set and $n \in \mathbb{N}^+$.*

(1) [HRW97] $L \in \mathrm{Easy}_\exists$ if and only if

(a) $L \in \mathrm{NP}$, and

(b) $(\exists NPTM\ M : L(M) = L)(\exists f \in \mathrm{FP}_t)(\forall x \in L)[f(x) \in \mathrm{acc}_M(x)]$

(2) $L \in \mathrm{Easy}_\exists^{(n)}$ if and only if

(a) $L \in \mathrm{NP}$, and

(b)
$$(\exists NPTM\ M : L(M) = L)(\exists f \in \mathrm{FP}_t)(\forall x \in L)(\exists u, v \in \Sigma^*)$$
$$\Big[\Big(f(x) \in \{0,1\} \ \wedge\ |uf(x)| = n \ \wedge\ uf(x)v \in \mathrm{acc}_M(x)\Big) \vee \Big((\forall y \in \mathrm{acc}_M(x))[|y| < n]\Big)\Big]$$

Obviously, the following inclusions hold:

Observation 4.2.8

(1) $\mathrm{P} \subseteq \mathrm{Easy}_\exists$,

(2) $\mathrm{Easy}_\forall \subseteq \mathrm{Easy}_\exists$,

(3) For all $n \in \mathbb{N}^+$, $\mathrm{Easy}'_\forall \subseteq \mathrm{Easy}_\exists^{(n)}$, and

(4) $\mathrm{Easy}_\exists \subseteq \mathrm{NP}$.

Recall Theorem 4.2.3. For the $\mathrm{Easy}_\exists^{(n)}$ classes, we can show that they are equal to each other, too.

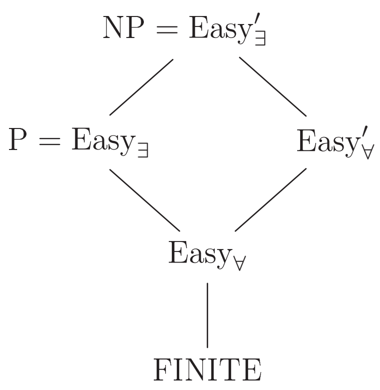


Figure 4.1: The Classes Easy$_\exists$ and Easy$_\forall$

Theorem 4.2.9

(1) $\text{Easy}_\exists = \text{P}$

(2) For all $n \in \mathbb{N}^+$, $\text{Easy}_\exists^{(n)} = \text{NP}$

Proof

(1) The inclusion $\text{P} \subseteq \text{Easy}_\exists$ holds by definition.

For the other direction, let $L \in \text{Easy}_\exists$ via the NPTM N and the function $f_N \in \text{FP}_t$. Then there exists a DPTM M that recognizes L as follows. On input x, M simulates the computation of $N(x)$ along the path $f_N(x)$. If $x \in L$ we have $f_N(x) \in \text{acc}_N(x)$ and M accepts x. If $x \notin L$ then $f_N(x)$ cannot be an accepting path of $N(x)$ and thus M rejects x.

(2) The inclusion $\text{Easy}_\exists^{(n)} \subseteq \text{NP}$ holds for all n by definition.

For the other direction, let $n \in \mathbb{N}^+$ and $L \in \text{NP}$. Then there exists an NPTM M with $L(M) = L$. We construct a new NPTM N in the following way. On input x, the machine N makes n irrelevant guesses and simulates M. Obviously $L(N) = L$ holds. But for all $x \in L$ we have $1^n v \in \text{acc}_N(x)$ with $v \in \text{acc}_M(x)$. It follows that $L \in \text{Easy}_\exists^{(n)}$ via the function $f(x) = 1$ for all x.

❑

Since all classes $\text{Easy}_\exists^{(n)}$ are equal we denote these classes with $\text{Easy}'_\exists$. The inclusion structure of the Easy-classes is shown in Figure 4.1.

4.3 The Operators wsol and ssol

A good starting point for the analysis of the complexity of search problems is the class $\mathrm{Easy}_\exists$. This class can be seen as a special case of a more general concept.

Definition 4.3.1 *Let $\mathcal{C}$ be a set of NTMs and $\mathcal{R}$ be a set of relations. We define*

$$\begin{aligned}\mathrm{Easy}_\exists(\mathcal{C},\mathcal{R}) =&\{L \subseteq \Sigma^* : (\exists M \in \mathcal{C})(\exists r_M \in \mathcal{R})[L = L(M) \wedge \\ & r_M \textit{ is a weak solution relation for } L \textit{ with respect to } M]\}.\end{aligned}$$

In particular, we have for instance $\mathrm{Easy}_\exists = \mathrm{Easy}_\exists(\mathrm{NP}, \mathrm{FP}_t)$.

In this definition we are interested in those $\mathcal{C}$-machine which possess weak solution relations in $\mathcal{R}$. Now we move the emphasis from $\mathcal{C}$ to $\mathcal{R}$. For this purpose we raise the question: In which way is the outcome influenced by $\mathcal{R}$, independently of the constraint given by $\mathcal{C}$. Under this aspect we define the operators wsol and ssol.

Definition 4.3.2 *For a class $\mathcal{R}$ of relations we define*

$$\begin{aligned}\mathrm{wsol}\cdot\mathcal{R} = \{L(M) : M \textit{ is an NTM } \wedge (\exists r \in \mathcal{R})[&r \textit{ is a weak solution} \\ &\textit{relation for } L \textit{ with respect to } M]\}\end{aligned}$$

and

$$\begin{aligned}\mathrm{ssol}\cdot\mathcal{R} = \{L(M) : M \textit{ is an NTM } \wedge (\exists r \in \mathcal{R})[&r \textit{ is a strong solution} \\ &\textit{relation for } L \textit{ with respect of } M]\}.\end{aligned}$$

Obviously $\mathrm{Easy}_\exists(\mathcal{C},\mathcal{R}) \subseteq \mathrm{wsol}\cdot\mathcal{R}$ holds for all $\mathcal{C}$. Definition 4.3.2 is a generalization of $\mathrm{Easy}_\exists$. Theorem 4.4.1 shows for instance $\mathrm{Easy}_\exists = \mathrm{wsol}\cdot\mathrm{FP}$.

An important aspect is the question which problems of a given complexity class can be solved by which solution relations. That is, can we arrange the wsol classes in known complexity classes?

We start with some elementary properties.

The first theorem shows that the operators wsol and ssol are monotone operators.

Theorem 4.3.3 *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two classes of relations.*

$$\begin{aligned}\mathcal{R}_1 \subseteq_c \mathcal{R}_2 &\implies \mathrm{wsol}\cdot\mathcal{R}_1 \subseteq \mathrm{wsol}\cdot\mathcal{R}_2\\ \mathcal{R}_1 \subseteq_c \mathcal{R}_2 &\implies \mathrm{ssol}\cdot\mathcal{R}_1 \subseteq \mathrm{ssol}\cdot\mathcal{R}_2\end{aligned}$$

Proof Let $L \in \mathrm{wsol}\cdot\mathcal{R}_1$. Hence there exists an NTM M with $L(M) = L$ and a relation $r_1 \in \mathcal{R}_1$ which is a weak solution relation for L with respect to M. This means that for all $x \in L$ we have $\varnothing \subset r_1(x) \subseteq \mathrm{acc}_M(x)$. By our assumption there exists a refinement $r_2 \in \mathcal{R}_2$ of r_1. So we get for all $x \in \Sigma^*$,

$$\begin{aligned}x \in L &\implies \varnothing \subset r_2(x) \subseteq r_1(x) \subseteq \mathrm{acc}_M(x),\\ x \notin L &\implies r_2(x) \subseteq r_1(x) = \varnothing.\end{aligned}$$

It follows that r_2 is a weak solution relation for L with respect to M and thus $L \in \mathrm{wsol} \cdot \mathcal{R}_2$.

The second proof is analogous to the first. ❑

The next theorem shows that some closure properties in the original relation class carry over to the corresponding wsol and ssol class.

Theorem 4.3.4

(1) If for a class $\mathcal{R}$ of relations it holds that $\mathrm{FP}_t \cdot \mathcal{R} \subseteq \mathcal{R}$, then the classes $\mathrm{ssol} \cdot \mathcal{R}$ and $\mathrm{wsol} \cdot \mathcal{R}$ are closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$-reductions.

(2) If a class $\mathcal{R}$ of relations is closed under concatenation, that is $\mathcal{R} \cdot \mathcal{R} \subseteq \mathcal{R}$, then the classes $\mathrm{ssol} \cdot \mathcal{R}$ and $\mathrm{wsol} \cdot \mathcal{R}$ are closed under intersection.

Proof We will prove the wsol statements. The proofs for the ssol statements are analogous.

(1) Let $\mathcal{R}$ be a class of relations satisfying $\mathrm{FP}_t \cdot \mathcal{R} \subseteq \mathcal{R}$. Let $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ via the function $f \in \mathrm{FP}_t$ and let $B \in \mathrm{wsol} \cdot \mathcal{R}$. We have to show that $A \in \mathrm{wsol} \cdot \mathcal{R}$.

Since $B \in \mathrm{wsol} \cdot \mathcal{R}$ we have an NTM M for B. We describe a new NTM M' for A. On input x, the machine M' computes $f(x)$ on all paths. Without loss of generality all paths have length $p(|x|)$ with $p \in \mathrm{Pol}$. Afterwards M' simulates the machine M with input $f(x)$. Obviously, it holds that $L(M') = A$.

If $r \in \mathcal{R}$ is a weak solution relation for B with respect to M then

$$r' = \{\langle x, 0^{p(|x|)}y\rangle : y \in r(x)\}$$

is a weak solution relation for A with respect to M'. Observe that $\mathrm{dom}(r') = \mathrm{dom}(r)$, hence we have $A \in \mathrm{wsol} \cdot \mathcal{R}$.

(2) Now let $\mathcal{R}$ be a class of relations which is closed under concatenation and let $A, B \in \mathrm{wsol} \cdot \mathcal{R}$ via the NTMs M_A, M_B and the weak solution relations $r_A, r_B \in \mathcal{R}$.

We construct an NTM M' for $A \cap B$.

On input x, the machine M' simulates $M_A(x)$. Afterwards, on every accepting path of $M_A(x)$, M' simulates M_B on input x. The machine M' accepts on some path if both simulations were successful, and the output on this path is the concatenation of both outputs of the machines M_A and M_B.

Obviously, it holds that $L(M) = A \cap B$ and the relation $r = r_A \cdot r_B$ is a weak solution relation for $A \cap B$ with respect to M'. ❑

In general, the wsol classes are not closed under union. In Section 4.4 we will show that $\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} = \mathrm{UP}$. It is known that UP being closed under union is improbable.

4.4 Some Special wsol and ssol Classes

Now we investigate which languages are obtained for a given class of solution relations. In particular, we study the power of solution relations from classes like $\mathrm{rel} \cdot \mathrm{NP}$, $\mathrm{fun} \cdot \mathrm{NP}$ or $\mathrm{fun} \cdot \mathrm{UP}$.

Theorem 4.4.1

$$\mathrm{wsol} \cdot \mathrm{FP} = \mathrm{P}$$

Proof First we show $\mathrm{wsol} \cdot \mathrm{FP} \subseteq \mathrm{P}$. Let $L \in \mathrm{wsol} \cdot \mathrm{FP}$, hence there exists an NTM M with $L(M) = L$ and a function $f_M \in \mathrm{FP}$ which is a weak solution function for L with respect to M. Let $p \in \mathrm{Pol}$ be the polynomial time-bound for f_M.

Now we describe a DPTM N for L. On input x, the machine N computes $f_M(x)$. Simultaneously, N counts the number of steps it carries out. The input x is rejected if after $p(|x|)$ steps N has no result for $f_M(x)$. Otherwise, $f_M(x)$ is defined and hence the path under consideration is an accepting path of M. In this case, the machine N accepts the input x. This shows $L(N) = L$ and hence $L \in \mathrm{P}$.

For the other direction let $L \in \mathrm{P}$. Hence there is a DPTM M with $L(M) = L$. Obviously, there exists an NPTM N for L. The machine N behaves exactly as M on all paths. The accepting behavior of N is as follows. If $x \in L$ and hence the machine M halts and accepts, then all paths of N are accepting paths. If $x \notin L$ and hence the machine M halts and accepts, then all paths of N are nonaccepting path. So if $p \in \mathrm{Pol}$ is the time function of M (and so of N) then

$$f(x) = \begin{cases} 0^{p(|x|)} & \text{if } x \in L, \\ \text{n. d.} & \text{if } x \notin L, \end{cases}$$

is a weak solution function for N. Clearly it holds that $f \in \mathrm{FP}$. ❏

The next theorem shows that solution relations from $\mathrm{rel} \cdot \mathrm{P}$ are strong enough for languages from NP.

Theorem 4.4.2

(1) $\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{P} = \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{UP} = \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP} = \mathrm{NP}$

(2) $\mathrm{wsol} \cdot \max \cdot \mathrm{P} = \mathrm{wsol} \cdot \min \cdot \mathrm{P} = \mathrm{NP}$

Proof (1). From Theorem 4.3.3 it follows that

$$\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{P} \subseteq \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{UP} \subseteq \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP}.$$

So it remains to show that $\mathrm{NP} \subseteq \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{P}$ and $\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{NP}$.

Let $L \in \mathrm{NP}$. Hence there exists a set $B \in \mathrm{P}$ and a polynomial $p \in \mathrm{Pol}$ with

$$x \in L \iff (\exists y \in \Sigma^* : |y| \leq p(|x|))[\langle x, y \rangle \in B].$$

We define a relation r as follows

$$r(x) = \{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge \ \langle x, y \rangle \in B\}.$$

Since $B \in \mathrm{P}$, it obviously holds that $r \in \mathrm{rel} \cdot \mathrm{P}$. As mentioned earlier, there is an NPTM M with $\langle x, y \rangle \in B \iff M$ accepts x along path y. From $L(M) = L$ and $\mathrm{acc}_M(x) = r(x)$ it follows that r is a weak solution relation for L with respect to M. So we get $L \in \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{P}$ and can conclude $\mathrm{NP} \subseteq \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{P}$.

Let $L \in \mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP}$. Hence there exists an NTM M and a weak solution relation $r \in \mathrm{rel} \cdot \mathrm{NP}$ for L with respect to M. Since $r \in \mathrm{rel} \cdot \mathrm{NP}$ we have an NPTM M_r which on input $x \in L$ accepts and outputs at least one accepting path of $M(x)$. For $x \notin L$ the machine M_r does not accept the input x. Obviously it holds that $L(M_r) = L$, and M_r is an NPTM. It follows that $\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{NP}$.

(2). We have to show two directions.

Let $L \in \mathrm{NP}$ and $\mathrm{op} \in \{\min, \max\}$. Hence we have an NPTM M with $L(M) = L$. We define

$$r(x) = \mathrm{op}\{y : M \text{ accepts } x \text{ along path } y\}.$$

Since M *accepts* x *along path* y is a P-predicate, we have $r(x) \in \mathrm{op} \cdot \mathrm{P}$ and obviously r is a weak solution relation for L with respect to M. Remember that the maximum and the minimum of the empty set are not defined. It follows that $L \in \mathrm{wsol} \cdot \mathrm{op} \cdot \mathrm{P}$.

Let $L \in \mathrm{wsol} \cdot \mathrm{op} \cdot \mathrm{P}$ for some $\mathrm{op} \in \{\min, \max\}$. Hence we have an NTM M with $L(M) = L$ and a weak solution relation $r \in \mathrm{op} \cdot \mathrm{P}$. Since r is polynomially bounded, so is M. Hence M is an NPTM and we can conclude $L \in \mathrm{NP}$. ❏

Now we restrict the solution relations to functions and expect that we obtain a (proper) subset of the languages from the previous case. This happens for solution functions from $\mathrm{fun} \cdot \mathrm{P}$ and $\mathrm{fun} \cdot \mathrm{UP}$.

Theorem 4.4.3

$$\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} = \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP} = \mathrm{UP}$$

Proof From Theorem 4.3.3 it follows that $\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} \subseteq \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP}$.

It remains to show $\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP} \subseteq \mathrm{UP}$ and $\mathrm{UP} \subseteq \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P}$.

Let $L \in \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP}$. Hence we have an NTM M with $L(M) = L$ and a weak solution function $f \in \mathrm{fun} \cdot \mathrm{UP}$ for L with respect to M. Since $f \in \mathrm{fun} \cdot \mathrm{UP}$, we have a UP-machine M_f which on input $x \in L$ accepts on exactly one path and outputs an accepting path of $M(x)$. For $x \notin L$ the machine M_f does not accept the input x. Obviously it holds that $L(M_f) = L$, and M_f is a UP-machine. It follows $\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{UP}$.

Let $L \in \mathrm{UP}$. Hence there exists a set $B \in \mathrm{P}$ and a polynomial $p \in \mathrm{Pol}$ with

$$x \in L \iff (\exists y \in \Sigma^* : |y| \leq p(|x|))[\langle x, y\rangle \in B] \quad \text{and}$$
$$||\{y : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}|| \leq 1.$$

We define a function f as follows

$$f(x) = \{y \in \Sigma^* : |y| \leq p(|x|) \wedge \langle x, y\rangle \in B\}.$$

It follows that $f \in \mathrm{fun} \cdot \mathrm{UP}$. As in the proof of 4.4.2, for the decision problem B we have an NTM M such that $\langle x, y\rangle \in B \iff M$ accepts x along path y. So we have an NTM M with $L(M) = L$ and $\mathrm{acc}_M(x) = f(x)$, hence f is a weak solution relation for L with respect to M. It follows that $L \in \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP}$, and we can conclude $\mathrm{UP} \subseteq \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP}$. ❑

Interestingly enough, the solution functions from $\mathrm{fun} \cdot \mathrm{NP}$ are as powerful as the solution relations from $\mathrm{rel} \cdot \mathrm{NP}$. (See Theorem 4.4.2)

Theorem 4.4.4

$$\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} = \mathrm{NP}$$

In order o prove Theorem 4.4.4 we need the concept of the UP-m-closure of NP.

Definition 4.4.5 *A language L is in $\mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$ if and only if there exists a language $A \in \mathrm{NP}$ and an NPTM M with:*

$$x \in L \implies M(x) \text{ has exactly one accepting path } \alpha \text{ whose output is } y\text{, and } y \in A.$$
$$x \notin L \implies M(x) \text{ does not accept.}$$

The following lemma is well-known.

Lemma 4.4.6

$$\mathrm{R}_m^{\mathrm{P}}(\mathrm{NP}) = \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP}) = \mathrm{R}_m^{\mathrm{NP}}(\mathrm{NP}) = \mathrm{NP}$$

The classes $\mathrm{R}_m^{\mathrm{P}}(\mathrm{NP})$ and $\mathrm{R}_m^{\mathrm{NP}}(\mathrm{NP})$ are defined similarly as $\mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$.

Proof of Theorem 4.4.4 To show Theorem 4.4.4, due to Lemma 4.4.6 it suffices to show that $\mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP}) = \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP}$. We start with $\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$.

Let $L \in \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP}$. Then there exists an NPTM M and a function $f \in \mathrm{fun} \cdot \mathrm{NP}$ with $\mathrm{dom}(f) = L$ and

$$\begin{aligned} x \in L &\implies M(x) \text{ accepts along path } f(x). \\ x \notin L &\implies M(x) \text{ does not accept.} \end{aligned}$$

If N is an NPTM computing the function f, we get

$$\begin{aligned} x \in L &\implies (\exists!!y)\underbrace{[N(x) \text{ has the output } y \wedge M(x) \text{ accepts along path } y.]}_{(*)} \\ x \notin L &\implies N(x) \text{ does not accept.} \end{aligned}$$

Since $(*)$ is an NP-predicate we have $L \in \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$.

To prove the other direction, let $L \in \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$.
Then there exists an NPTM M and a language $A \in \mathrm{NP}$ with

$$\begin{aligned} x \in L &\implies (\exists!!\alpha)(\exists!!y)[M(x) \text{ outputs } y \text{ (and accepts) along } \alpha \text{ and } y \in A]. \\ x \notin L &\implies \neg(\exists\alpha)(\exists y)[M(x) \text{ outputs } y \text{ (and accepts) along } \alpha]. \end{aligned}$$

We construct a new NPTM M' according to

$$M'(x) \text{ accepts along } \alpha\#y \iff M(x) \text{ outputs } y \text{ along } \alpha.$$

Then we can write

$$\begin{aligned} x \in L &\implies (\exists!!\alpha)(\exists!!y)[(x, \alpha\#y) \in \Sigma^* \times (\Sigma^*\#A) \wedge M'(x) \text{ accepts along } \alpha\#y]. \\ x \notin L &\implies \neg(\exists\alpha)(\exists y)[M'(x) \text{ accepts along } \alpha\#y]. \end{aligned}$$

We define a function g for all $x \in \Sigma^*$ as follows:

$$g(x) = \begin{cases} \alpha\#y & \text{if } (x, \alpha\#y) \in \Sigma^* \times (\Sigma^*\#A) \text{ and } M'(x) \text{ accepts along } \alpha\#y, \\ \text{n. d.} & \text{otherwise.} \end{cases}$$

Obviously, it holds that $g \in \mathrm{NP}$, and since g is a function, even $g \in \mathrm{fun} \cdot \mathrm{NP}$. Furthermore g is a weak solution function for L with respect to M'. It follows that $L \in \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP}$. ❏

Note that the equation $\mathrm{wsol} \cdot \mathrm{rel} \cdot \mathrm{NP} = \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP}$ does not imply that $\mathrm{rel} \cdot \mathrm{NP} \subseteq_c \mathrm{fun} \cdot \mathrm{NP}$, since a solution relation from $\mathrm{rel} \cdot \mathrm{NP}$ and a solution function from $\mathrm{fun} \cdot \mathrm{NP}$ for one and the same language can belong to different Turing machines. Of course they have the same domain.

The equality $\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} = \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP})$ is not an isolated result. The following table shows the results of the Theorems 4.4.1 and 4.4.3 in another light.

$$\begin{aligned} \mathrm{wsol} \cdot \mathrm{FP} &= \mathrm{P} = \mathrm{R}_m^{\mathrm{P}}(\mathrm{P}) \\ \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} &= \mathrm{UP} = \mathrm{R}_m^{\mathrm{UP}}(\mathrm{P}) \\ \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP} &= \mathrm{UP} = \mathrm{R}_m^{\mathrm{UP}}(\mathrm{UP}) \\ \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} &= \mathrm{NP} = \mathrm{R}_m^{\mathrm{UP}}(\mathrm{NP}) \end{aligned}$$

The situation for the ssol-classes is slightly simpler.

Theorem 4.4.7

$$\mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{P} = \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{UP} = \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{NP} = \mathrm{NP} \cap \mathrm{coNP}$$

Proof From Theorem 4.3.3 it follows

$$\mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{P} \subseteq \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{UP} \subseteq \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{NP}.$$

At first we show $\mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{NP} \subseteq \mathrm{NP} \cap \mathrm{coNP}$. Let $L \in \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{NP}$, hence we have an NTM M with $L(M) = L$ and a strong solution relation $r \in \mathrm{rel} \cdot \mathrm{NP}$. Since $r \in \mathrm{rel} \cdot \mathrm{NP}$ we have an NPTM M_r with $acc_{M_r}(x) = r(x)$ for all x. We build two NPTMs M_L with $L(M_L) = L$ and $M_{\overline{L}}$ with $L(M_{\overline{L}}) = \overline{L}$. Both M_L and $M_{\overline{L}}$ work in the same way as M_r but have different accepting behavior. The machine M_L accepts exactly on those paths on which M_r accepts and outputs a string beginning with 1. The machine $M_{\overline{L}}$ accepts exactly on those paths on which M_r accepts and outputs a string beginning with 0. So we have $L, \overline{L} \in \mathrm{NP}$ and hence $L \in \mathrm{NP} \cap \mathrm{coNP}$.

It remains to show $\mathrm{NP} \cap \mathrm{coNP} \in \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{P}$.

Let $L \in \mathrm{NP} \cap \mathrm{coNP}$, hence we have an NPTM M_L with $L(M_L) = L$ and an NPTM $M_{\overline{L}}$ with $L(M_{\overline{L}}) = \overline{L}$. We define for all $x \in \Sigma^*$:

$$\begin{aligned} r_L &= \{\langle x, 1y\rangle : y \in \mathrm{acc}_{M_L}(x)\}, \\ r_{\overline{L}} &= \{\langle x, 0y\rangle : y \in \mathrm{acc}_{M_{\overline{L}}}(x)\}, \\ r &= r_L \cup r_{\overline{L}}. \end{aligned}$$

Since $r_L, r_{\overline{L}} \in \mathrm{P}$ and P is closed under union, we get $r \in \mathrm{P}$. Obviously, r is a relation and hence we obtain $r \in \mathrm{rel} \cdot \mathrm{P}$. But r is a strong solution relation for L with respect to M_L and hence we have $L \in \mathrm{ssol} \cdot \mathrm{rel} \cdot \mathrm{P}$. ❑

For classes of functions we present the following result.

Theorem 4.4.8

(1) $\mathrm{ssol} \cdot \mathrm{FP} = \mathrm{P}$

(2) $\mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{P} = \mathrm{ssol}(\mathrm{fun} \cdot \mathrm{UP}) = \mathrm{UP} \cap \mathrm{coUP}$

(3) $\mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{NP} = \mathrm{NP} \cap \mathrm{coNP}$

Proof

(1), (2). These proofs are analogous to the proof of Theorem 4.4.1 and to the proof of Theorem 4.4.7, respectively, and thus are omitted.

(3). From Theorem 4.3.3 and Theorem 4.4.7 it follows that $\mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{NP} \subseteq \mathrm{NP} \cap \mathrm{coNP}$.

Now we show that $\mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{NP}$.

Let $L \in \mathrm{NP} \cap \mathrm{coNP}$. From Theorem 4.4.4 it follows that $L \in \mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP}$. Hence there exists an NPTM M and a weak solution function $f \in \mathrm{fun} \cdot \mathrm{NP}$ for L with respect to M. We define $f_1(x) = 1f(x)$ for all $x \in L$. Note that for all $x \in \Sigma^*$ the function $f_1(x)$ is defined if and only if $f(x)$ is defined. Obviously, f_1 is a function from $\mathrm{fun} \cdot \mathrm{NP}$.

Define $f_2 = \{\langle x, 0\rangle : x \notin L\}$. Since L is a coNP language and f_2 is a function, we have $f_2 \in \mathrm{fun} \cdot \mathrm{NP}$.

Let $g = f_1 \cup f_2$. Since f_1 and f_2 are disjoint sets from NP, it follows that $g \in \mathrm{NP}$, and since g is a function, we obtain $g \in \mathrm{fun} \cdot \mathrm{NP}$. Obviously, g is a strong solution function for L with respect to M. ❑

4.5 Open Problems

From Corollary 4.2.5 we know that

$$\mathrm{P} \subseteq \mathrm{Easy}'_{\forall} \implies \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}.$$

We do not know whether the converse direction holds. Furthermore we are interested in structural consequences that follow from $\mathrm{Easy}'_{\forall} \subseteq \mathrm{P}$ and $\mathrm{Easy}'_{\forall} \subseteq \mathrm{Easy}_{\forall}$.

One starting point for a proceeding research might be wsol and ssol operators based on $\mathrm{Easy}_{\forall}$ instead of $\mathrm{Easy}_{\exists}$. Another possible alternative are modifications of the concept of weak solution functions. We could require them to be total, such as the solution functions at the $\mathrm{Easy}_{\exists}$ and the $\mathrm{Easy}_{\forall}$ classes.

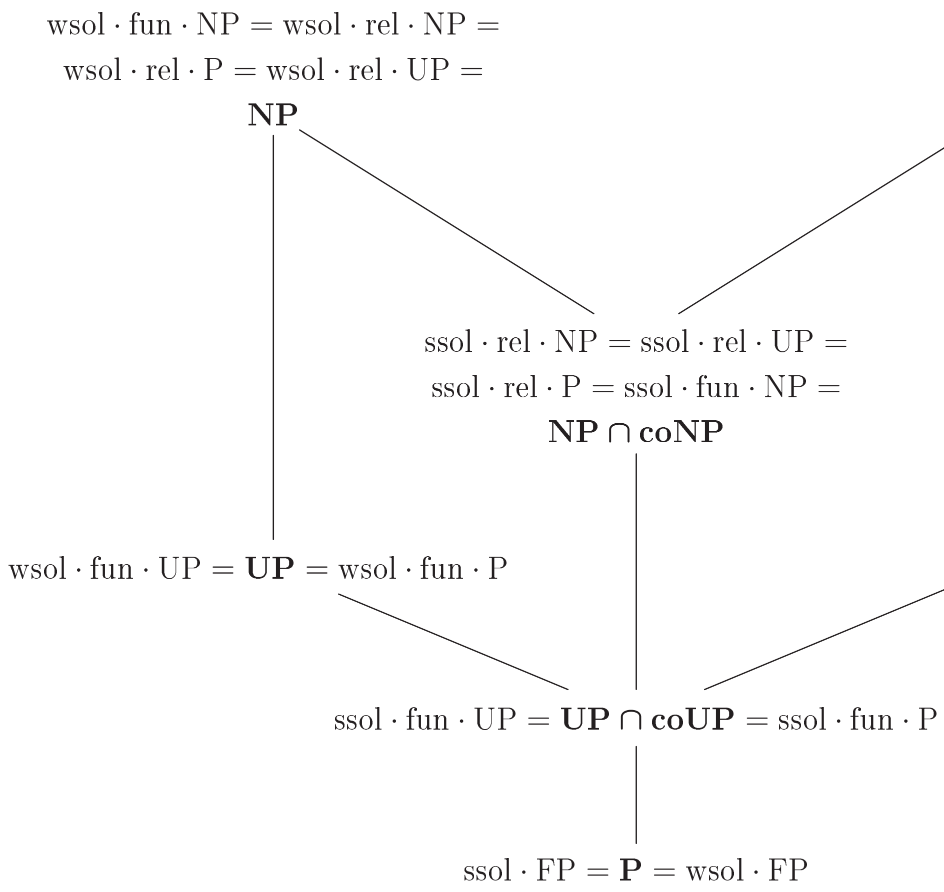


Figure 4.2: The wsol and ssol Classes

List of Symbols

Subject Index

T

W

Bibliography

[AKS02] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P, August 2002. preprint.

[BD76] A. Borodin and A. Demers. Some comments on functional self-reducibility and the NP hierarchy. Technical Report TR 76-284, Cornell Department of Computer Science, Ithaca, NY, July 1976.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1988. 2nd edition 1995.

[BDG90] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II.* EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1990.

[BGH90] R. Beigel, J. Gill, and U. Hertrampf. Counting classes: Thresholds, parity, mods, and fewness. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, pages 49–57. Springer-Verlag *Lecture Notes in Computer Science #415*, February 1990.

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

[BLS85] R. Book, T. Long, and A. Selman. Qualitative relativizations of complexity classes. *Journal of Computer and System Sciences*, 30:395–413, 1985.

[CCHO03] J. Cai, V. Chakaravarthy, L. Hemaspaandra, and M. Ogihara. Competing provers yield improved karp-lipton collapse results. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 535–546. Springer-Verlag *Lecture Notes in Computer Science #2607*, 2003.

[CGH+88] 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.

[CGH+89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.

[CH86] J. Cai and L. Hemachandra. The boolean hierarchy: Hardware over NP. In *Proceedings of the 1st Structure in Complexity Theory Conference*, pages 105–124. Springer-Verlag *Lecture Notes in Computer Science #223*, June 1986.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[Edm65] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[FFNR96] S. Fenner, L. Fortnow, A. Naik, and J. Rogers. Inverting onto functions. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, pages 213–222. IEEE Computer Society Press, May 1996.

[FGH+96] S. Fenner, F. Green, S. Homer, A. Selman, T. Thierauf, and H. Vollmer. Complements of multivalued functions. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 260–269. IEEE Computer Society Press, June 1996.

[FHOS97] S. Fenner, S. Homer, M. Ogiwara, and A. Selman. Oracles that compute values. *SIAM Journal on Computing*, 26(4):1043–1065, 1997.

[GH03] A. Große and H. Hempel. On functions and relations. Technical Report Math/Inf/01/03, Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany, January 2003.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988. plib.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre*. Leipzig, 1914.

[Hem03] H. Hempel. Functions in complexity theory. Habilitation Thesis, Friedrich-Schiller-Universität Jena, Jena, Germany, January 2003.

[HHN+93] L. Hemaspaandra, A. Hoene, A. Naik, M. Ogiwara, A. Selman, T. Thierauf, and J. Wang. Selectivity: Reductions, nondeterminism, and function classes. Technical Report TR-469, University of Rochester, Department of Computer Science, Rochester, NY, August 1993.

[HNOS93] E. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. Technical Report 93-21, State University of New York at Buffalo, Department of Computer Science, Buffalo, NY, 1993.

[HNOS96] L. Hemaspaandra, A. Naik, M. Ogihara, and A. Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25(4):697–708, 1996.

[HRW97] L. Hemaspaandra, J. Rothe, and G. Wechsung. Easy sets and hard certificate schemes. *Acta Informatica*, 34:859–879, 1997.

[HV95] L. Hemaspaandra and H. Vollmer. The satanic notations: Counting classes beyond $\# \cdot \mathrm{P}$ and other definitional adventures. *SIGACT News*, 26(1):2–13, 1995.

[HVW95] U. Hertrampf, H. Vollmer, and K. Wagner. On the power of number-theoretic operations with respect to counting. In *Proceedings of the 10th Structure in Complexity Theory Conference*, pages 299–314. IEEE Computer Society Press, June 1995.

[HW00] H. Hempel and G. Wechsung. The operators min and max on the polynomial hierarchy. *International Journal of Foundations of Computer Science*, 11(2):315–342, 2000.

[Kar72] R. Karp. Reducibilities among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, 1972.

[Kha79] L. G. Khachiyan. A polynomial algorithm for linear programming. *Soviet Math. Dokl.*, 20:191–194, 1979.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, April 1980. An extended version has also appeared as: Turing machines that take advice, *L'Enseignement Mathématique*, 2nd series 28, 1982, pages 191–209.

[Ko83] K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26:209–221, 1983.

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

[KSV98] S. Kosub, H. Schmitz, and H. Vollmer. Uniformly defining complexity classes of functions. In *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, pages 607–617. Springer-Verlag *Lecture Notes in Computer Science #1373*, 1998.

[KSW87] J. Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *R.A.I.R.O. Informatique théorique et Applications*, 21:419–435, 1987.

[KW98] J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. *SIAM Journal on Computing*, 28(1):311–324, 1998.

[MS73] A. Meyer and L. Stockmeyer. Word problems requiring exponential time. In *Proceedings of the 5th ACM Symposium on Theory of Computing*, pages 1–9, 1973.

[Pap94] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[PY84] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[Sch76] C. P. Schnorr. The network complexity and the turing machine complexity of finite functions. *AI*, 7:95–107, 1976.

[Sch79] C. P. Schnorr. In self-transformable combinatorial problems. Preprint, 1979.

[Sel94] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*, 48(2):357–381, 1994.

[Sel96] A. Selman. Much ado about functions. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 198–212. IEEE Computer Society Press, June 1996.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Tur36] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, ser. 2*, 42:230–265, 1936. Correction, *ibid*, vol. 43, pp. 544–546, 1937.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5:20–23, 1976.

[VW93] H. Vollmer and K. Wagner. The complexity of finding middle elements. *International Journal of Foundations of Computer Science*, 4:293–307, 1993.

[Wec85] 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).

[Wec00] G. Wechsung. *Vorlesungen zur Komplexitätstheorie*. Teubner-Texte zur Informatik. B.G. Teubner, Stuttgart, 2000. in German.

[Wra77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.

[WT92] O. Watanabe and S. Toda. Polynomial time 1-Turing reductions from #PH to #P. *Theoretical Computer Science*, 100(1):205–221, 1992.

[WW86] K. Wagner and G. Wechsung. *Computational Complexity*. D. Reidel Publishing Company, 1986. Distributors for the U.S.A. and Canada: Kluwer Academic Publishers.

Zusammenfassung

Die Klassifizierung von Problemen bezüglich ihrer Komplexität ist der Kernpunkt der Komplexitätstheorie. Natürlich müssen wir erklären, was wir unter Problemen und deren Komplexität verstehen. Probleme sind für uns üblicherweise Entscheidungsprobleme, d. h. die Frage, ob ein gegebenes Objekt zu einer bestimmten Menge gehört oder nicht. Ein solches Entscheidungsproblem zu lösen, bedeutet, einen Algorithmus anzugeben, welcher zu einer gegebenen Eingabe x die Zugehörigkeit zu der entsprechenden Menge entscheidet.

Damit haben wir auch eine Möglichkeit, die Komplexität eines Problems zu messen, allerdings nur in Bezug auf den verwendeten Algorithmus. Wir messen, wieviele Ressourcen der Algorithmus für seine Entscheidung benötigt, in Abhängigkeit von der Eingabelänge.

Um Algorithmen exakt formulieren zu können, benötigen wir ein Berechnungsmodell. Das Standardmodell in der Komplexitätstheorie ist das der Turingmaschine. Dieses universelle Modell wurde 1936 von Turing [Tur36] entwickelt. Wir unterscheiden dabei eine deterministische und eine nichtdeterministische Variante.

Bei der Frage nach dem Ressourcenverbrauch gibt es verschiedene Varianten. Eine Möglichkeit ist die Frage nach der benötigten Zeit. Dazu zählen wir die Anzahl der Schritte, die eine passende Turingmaschine für ihre Entscheidung benötigt hat. Dies erlaubt uns eine Klassifizierung von Problemen nach der Laufzeit entsprechender Lösungsalgorithmen. Man beachte, daß man durch die Angabe eines Lösungsalgorithmus nur eine obere Schranke erhält. Der Beweis einer scharfen unteren Schranke gestaltet sich oft schwierig und ist mitunter nicht möglich.

Als Beispiel betrachten wir die durch Edmonds [Edm65] eingeführte Komplexitätsklasse P. Diese Klasse enthält alle Mengen, welche durch eine deterministische polynomialzeitbeschränkte Turingmaschine entschieden werden können. Das heißt, für jede Menge A in P existiert eine entsprechende Turingmaschine, welche für jede Eingabe x nach $p(|x|)$ Takten ihre Entscheidung fällt. Dabei ist p ein zu A passende Polynom. Die Probleme in P werden üblicherweise als praktisch-machbare Probleme betrachtet. Viele natürliche und nicht triviale Probleme gehören zu der Klasse P, z. B. das Finden eines maximalen Matchings in Graphen [Edm65], lineare Optimierung [Kha79] und der Test, ob eine gegebene natürliche Zahl eine Primzahl ist [AKS02].

Eine weitere grundlegende Komplexitätsklasse ist NP. Die Klasse aller Mengen, die durch eine nichtdeterministische polynomialzeitbeschränkte Turingmaschine akzeptiert werden können. Offensichtlich sind alle Probleme aus P auch in NP. Die Frage, ob es ein Problem aus NP gibt, welches nicht in P liegt, konnte trotz intensiver Forschung noch nicht beantwortet werden, obwohl es viele Kandidaten dafür gibt.
Viele dieser Kandidaten besitzen die Eigenschaft, daß sie die schwersten Probleme in NP sind. Damit ist gemeint, daß allein aus der Tatsache, daß eines dieser Probleme in P liegt, folgen würde, daß P = NP gilt. Ein Beispiel eines solchen Problems ist das Handlungsreisendenproblem. Ein Händler möchte einige vorgegebene Städte besuchen – existiert eine Route, die eine vorgegebene Länge unterschreitet?
Die Frage, ob P = NP gilt, war der Ausgangspunkt eines ganzen Forschungsgebietes. Viele neue Fragen wurden untersucht und gelöst. Eine große Anzahl weiterer Komplexitätsklassen wurden definiert und studiert und erlaubten einen tiefen Einblick in dieses Forschungsgebiet. Es gab viele Ansätze, die Frage, ob P = NP gilt, zu lösen, aber bis heute ist die Antwort darauf unbekannt.
Neben Entscheidungsproblemen spielen Relationen eine wichtige Rolle in der Komplexitätstheorie, nicht nur als Werkzeug, sondern auch als Forschungsobjekte selbst. Diese noch sehr junge Forschungsrichtung wurde wesentlich durch die Arbeiten von Selman [Sel94, Sel96] in den frühen neunziger Jahren beeinflußt. Viele verschiedene Klassen von Relationen und – als Spezialfall – Klassen von Funktionen wurden untersucht. Um nur zwei zu nennen: FP – die Klasse aller in deterministischer Polynomialzeit berechenbarer Funktionen, NPMV – die Relationen, welche durch eine nichtdeterministische Polynomialzeit-Turingmaschine berechnet werden können [BLS84, BLS85]. Genaue Definitionen enthält das Kapitel 3.
Bei der Definition von Relationenklassen folgen wir [Wec00] und [HW00]. Der Kernpunkt dieses systematischen Zugangs ist die Definition von Relationenklassen basierend auf gut bekannten Komplexitätsklassen anstatt auf der Berechnung von Turingmaschinen. Dieser Zugang führt nicht nur zu natürlichen Bezeichnungen, er erlaubt auch Beweise für sehr allgemeine Aussagen.
Zum Beispiel definieren wir [Wec00] folgend:

- $r \in \mathrm{rel} \cdot \mathcal{C} \iff (\exists B \in \mathcal{C})(\exists p \in \mathrm{Pol})(\forall x \in \Sigma^*)$ $[r(x) = \{y \in \Sigma^* : |y| \leq p(|x|) \ \wedge \ \langle x, y \rangle \in B\}]$,
- $f \in \mathrm{fun} \cdot \mathcal{C} \iff f \in \mathrm{rel} \cdot \mathcal{C} \ \wedge \ (\forall x \in \Sigma^*)[||f(x)|| \leq 1]$.

Zunächst beweisen wir einige allgemeine Resultate. Zum Beispiel überträgt sich der bekannte Projektionssatz von den Komplexitätsklassen auf Klassen von Relationen. Bekannt war, daß eine Vergleichbarkeit der Klassen $\mathrm{fun} \cdot \mathrm{NP}$ und $\mathrm{fun} \cdot \mathrm{coNP}$ bezüglich der Inklusion wahrscheinlich ist. Es konnte gezeigt werden, daß bei der Beschränkung auf totale Funktionen, $\mathrm{fun}_t \cdot \mathrm{NP}$ und $\mathrm{fun}_t \cdot \mathrm{coNP}$, die Inklusion $\mathrm{fun}_t \cdot \mathrm{NP} \subseteq \mathrm{fun}_t \cdot \mathrm{coNP}$ gilt.

Wir zeigen auch eine Möglichkeit, wie man Relationen als Orakel verwenden kann. Ein Frage x an ein Relationenorakel r liefert in diesem Fall ein Element der Menge $r(x)$. Zu klären ist dabei, wie man mit der Tatsache umgeht, daß bei gleichen Fragen verschiedene Antworten geliefert werden können.
Um Eigenschaften der Komplexitätsklassen auf die Relationenklassen übertragen zu können, nutzen wir die sogenannte Operatorenmethode, welche auch schon in anderen Gebieten erfolgreich angewandt wurde [VW93, HW00]. Damit gelingt es für fast alle Inklusionen, die wir nicht zeigen können, unwahrscheinliche Folgerungen zu beweisen.
Zwei Beispiele:

$$\begin{aligned} \mathrm{rel} \cdot \mathrm{P} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \mathrm{P} &\implies \mathrm{NP} = \mathrm{UP} \\ \mathrm{rel} \cdot \mathrm{P}^{\mathrm{NP}} \subseteq_{\mathrm{c}} \mathrm{FP}^{\mathrm{NP}} &\implies \mathrm{P}^{\mathrm{NP}} = \mathrm{NP}^{\mathrm{NP}} \end{aligned}$$

Ein Typ von Inklusionen, bei dem die Operatorenmethode keine Ergebnisse erzielt, wird durch die Verwendung nicht uniformer Komplexitätsklassen behandelt. Dies erlaubt den Beweis des folgenden Resultats:

$$\begin{aligned} \mathrm{rel} \cdot \Pi_k^{\mathrm{p}} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \Pi_k^{\mathrm{p}} &\implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma_{k+1}^{\mathrm{p}}}, \\ \mathrm{rel} \cdot \Sigma_k^{\mathrm{p}} \subseteq_{\mathrm{c}} \mathrm{fun} \cdot \Sigma_k^{\mathrm{p}} &\implies \mathrm{PH} = \mathrm{ZPP}^{\Sigma_k^{\mathrm{p}}}. \end{aligned}$$

Im zweiten Teil der vorliegenden Dissertation studieren wir sogenannte „*easy*-languages“, also in irgendeiner Form einfache Sprachen. Dabei handelt es sich um Sprachen mit einfach zu berechnenden Lösungsrelationen. Das heißt, es gibt zu einer solchen Sprache eine Relation, die akzeptierende Pfade einer entsprechenden Turingmaschine berechnet.
Ein Resultat von Borodin und Demers [BD76] ist dabei der Ausgangspunkt dieser Forschung. Sie zeigten, daß unter Annahme einer allgemein anerkannten Vermutung eine Menge existiert, welche einfach zu entscheiden ist, für die es aber schwer zu bestimmen ist, warum ein Element dazugehört. Dies bedeutet, daß es schwer ist eine entsprechende Lösungsrelation zu berechnen.
Wie in [HRW97] eingeführt, definieren wir die zwei Komplexitätsklassen $\mathrm{Easy}_\forall$ und $\mathrm{Easy}_\exists$. Die Klasse $\mathrm{Easy}_\forall$ enthält alle Sprachen, für die *jede* nichtdeterministische Turingmaschine, die eine solche Sprache akzeptiert, eine Lösungsfunktion aus FP_t besitzt. Für $\mathrm{Easy}_\exists$ genügt es, wenn jeweils *eine* Turingmaschine eine solche leichte Lösungsfunktion besitzt.
Zunächst interessieren wir uns dafür, was passiert, wenn wir nicht eine Lösungsfunktion fordern, sondern eine Funktion, die nur ein Bit eines akzeptierenden Pfades berechnet. Weiterhin untersuchen wir ob es einen Unterschied macht, um welches Bit es sich dabei handelt. Dabei stellt sich heraus, daß es keine Rolle spielt.
Danach stellen wir uns die Frage, welche Sprachen wir erhalten, wenn wir die Definition von $\mathrm{Easy}_\exists$ abwandeln und andere Relationenklassen anstelle von FP_t erlauben.

Dazu führen wir die Operatoren wsol und ssol ein. Die Klassen $\mathrm{wsol} \cdot \mathcal{R}$ und $\mathrm{ssol} \cdot \mathcal{R}$ enthalten alle Sprachen, die durch eine nichtdeterministische Turingmaschine akzeptiert werden können, die eine schwache bzw. starke Lösungsrelation aus $\mathcal{R}$ besitzen. Der Unterschied zwischen wsol und ssol besteht im Umgang mit Wörtern, die nicht zu der betrachteten Sprache gehören. Bei wsol sind die Lösungsrelationen nicht definiert, bei ssol müssen die Lösungsrelationen durch entsprechende Werte anzeigen, wenn das übergebene Wort nicht zu der Sprache gehört.
Es ergeben sich dabei unter anderem die folgenden Resultate:

$$\begin{array}{lcllcl}
\mathrm{wsol} \cdot \mathrm{FP} & = & \mathrm{P} & \mathrm{ssol} \cdot \mathrm{FP} & = & \mathrm{P} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{P} & = & \mathrm{UP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{P} & = & \mathrm{UP} \cap \mathrm{coUP} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{UP} & = & \mathrm{UP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{UP} & = & \mathrm{UP} \cap \mathrm{coUP} \\
\mathrm{wsol} \cdot \mathrm{fun} \cdot \mathrm{NP} & = & \mathrm{NP} & \mathrm{ssol} \cdot \mathrm{fun} \cdot \mathrm{NP} & = & \mathrm{NP} \cap \mathrm{coNP}
\end{array}$$

Selbständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, den 3.2.2004

André Große

Tabellarischer Lebenslauf

Persönliche Daten

Name	André Große
Geburt	16.2.1975 in Jena

Schulausbildung

1981 – 1989	Polytechnische Oberschule „Otto Grotewohl“ in Jena
1989 – 1993	Spezialschule mathematisch-naturwissenschaftlich-technischer Richtung „Carl Zeiss“ in Jena Abschluß: Abitur

Studium

10/1994 – 3/2000	Studium der Mathematik an der Friedrich-Schiller-Universität Jena Abschluß: Diplom-Mathematiker
4/2000 – 3/2002	Landesgraduiertenstipendium des Landes Thüringen an der Friedrich-Schiller-Universität Jena
seit 4/2002	Wissenschaftlicher Mitarbeiter am Institut für Informatik der Friedrich-Schiller-Universität Jena