Inaugural Dissertation

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Diplommathematikerin Levke Bentzien

aus Bretten

2000

NP-Completeness Notions under Strong Hypotheses

1. Gutachter:	Prof. Dr. Klaus Ambos-Spies
2. Gutachter:	Dr. Frank Stephan

Tag der mündlichen Prüfung: 30. Mai 2000

Von der Fakultät genehmigte Fassung

Contents

Acknowledgments

This thesis was written during the time from October 1995 to March 1999. First of all I would like to thank my advisor Klaus Ambos-Spies for the stimulating and encouraging start and the constant support he provided for the investigations in this thesis, and the many fruitful and interesting discussions.

For helpful and thought-provoking discussions, I would like to thank especially Klaus Ambos-Spies, Richard Beigel, Bernd Borchert, Peter Fejer, Jack Lutz, Gunther Mainhardt, Elvira Mayordomo, Wolfgang Merkle, Andrej Morosov, Andre Nies, Jan Reimann, Frank Stephan, Sebastiaan Terwijn and Jongge Wang.

Also I would like to thank Klaus-Jörn Lange, Peter Kadau, Henning Fernau, Rolf Niedermeier, and Klaus Reinhardt. Their support, kindness, and talents in establishing a good working atmosphere has been very helpful.

I am indebted to Prof. Klaus Gloede, Prof. G. H. Müller and Prof. A. Brandis for their interest and guidance during my studies at the University of Heidelberg.

Chapter 1

Introduction

Surely the most famous open problem in the field of structural complexity is the so-called P = NP problem. In his address at the International Congress of Mathematicians in Paris in 1900, Hilbert posed a list of 23 problems that were meant — and in fact turned out — to be both challenging for the mathematical community and pointing to the future . This was the tenth problem.

> 'Eine diophantische Gleichung mit irgendwelchen Unbekannten und mit ganzen rationalen Zahlenkoeffizienten sei vorgelegt:
>
> *Man soll ein Verfahren angeben, nach welchem sich mittels einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*'
>
> ['Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: *To devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*'] (citation and translation taken from [Buh93])

The way Hilbert stated this problem appears both optimistic and demanding. In fact, too optimistic. It took seven decades till finally Matijasevič [Mat70] proved that the correct solution is a negative one. What does it mean for Hilbert's problem to have a negative solution? Apparently, he based his question on an intuitive notion of "finite number of operations" of a "process", i.e., on an intuitive notion of algorithm, that could enable him to check any suggested *positive* solution. But in order to *prove* that no positive solution exists, this intuitive notion has to be replaced by a formal one.

With the work of Gödel [Göd31], Kleene [Kle36], Post [Pos36], Church [Chu33, Chu36], and Turing [Tur36, Tur37] on such formalisms started the development of a robust notion of *computability*, which can be equivalently characterized by various logical calculi and schemes, or by the machine model of Turing. This model most easily allows for a formal measurement of resources used by the machine in finding solutions, and for a notion of the "relative difficulty" of explicit problems. By formulating a self-referential problem about his machine model, the so called Halting Problem, Turing also marked the *borderline of computability* by showing that this problem lies beyond it. This result gave the ground stone in the solution of Hilbert's Problem, since Matijasevič finally succeeded in finding the right way to relate the problem of solvability of a diophantine equation to the Halting problem such that the status of the latter as proven by Turing entails the solution of Hilbert's problem to the negative.

When the formal notion of computability found its practical counterpart in real-life computers, it soon became obvious that the borderline which is represented by the Halting Problem is at astronomic distance to a realistic, practical borderline of computability. In 1965, Edmonds [Edm65] suggested the polynomial-time bound as theoretical characterization of this practical borderline. Allthough only polynomials of very low degree are of significance in practice, this suggestion soon became accepted among computer scientists.

The class that embodies these, not only computable but, above that, *feasible* problems is denoted by P. A problem belongs to this class if membership of any instance, represented by an input string of certain length in a fixed alphabet Σ, can be decided by a Turing machine that stops and "accepts" or "rejects" the input after a finite number of "steps" (which makes the problem computable), while the total number of steps needed to end this "computation" is bounded by a polynomial in the length of the input (which makes the problem feasible).

Many natural and fundamental tasks in computer science belong to P. For example, sorting of lists, matrix multiplication and inversion, manipulations of graphs, approximating solutions to differential equations and the like. However, for many interesting tasks, for which it would be desirable to have a clever and quick algorithm, it is doubtful whether they belong to P. Typically, these are problems such that any solution seems "hard to find, easy to check". For example, packing problems, the traveling-salesman-problem, many decision problems of graph properties, and, as one of the "standard representatives" of these kind of problems, to decide whether a given sentence of propositional logic has a satisfying truth-value assignment to its variables. No clever way of *finding* solutions to these problems in polynomial time are known (like the way one *finds* the lexicographic ordering of a great number of names), but any *suggested* or *guessed* solution can be written down, checked and verified in polynomial time. The class that embodies these problems is denoted by NP for *nondeterministic polynomial time.* Here nondeterminism is the theoretical model for "guessing" the solution.

The polynomial bound on the length of possible "guesses" ensures that problems in NP allow for the strategy of "checking all the possibilities". But, obviously, this strategy might consume a number of steps that is not bounded by a polynomial in the input length. Typically, there are exponentially many sensible guesses to check, hence the best bound on the total number of steps seems to be at least exponential in the total length of the guess. From this point of view the polynomial many steps needed to check each of these possibilities appears almost negligible.

These considerations show that the class NP is located between the *polynomial* time class P and the *exponential* time class EXP, that $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{EXP}$. But, which one of these inclusions is proper has been one of the pressing open questions in complexity theory for more than three decades now. For many practical reasons it would be both desirable (except in cryptography), and challenging to learn that $\mathrm{P} = \mathrm{NP}$, but this is widely disbelieved.

A further uncertainty about the class NP concerns closure under complementation. Deterministic time classes like P or EXP, as well as deterministic and nondeterministic space classes (see the independent results of Immerman [Imm88] and Szelepcsényi [Sze88]) are closed under complementation. It is not known whether this holds for nondeterministic time classes like NP.

Consider the problems PRIME and COMPOSITE. A natural number is a member of COMPOSITE if it is not a prime number, and PRIME is the complement of this set in $\mathbb{N}$. It is easy to see that COMPOSITE is a problem in NP. (On input n just nondeterministically "guess" two numbers $a, b < n$ and check that $n = a \cdot b$.) So PRIME belongs to the class $co\text{-NP} = \{A : \overline{A} \in \mathrm{NP}\}$. On the other hand, some knowledge of number theory shows that $\text{PRIME} \in \mathrm{NP}$, too. Besides the trivial inclusions

$$\mathrm{P} \subseteq \mathrm{NP} \cap co\text{-NP} \subseteq \mathrm{NP},$$

the exact relation between the classes P, $\mathrm{NP} \cap co\text{-NP}$, and NP are unknown The wide-spread belief tells that these classes all differ.

From a structural point of view the uncertainty about the class NP is especially puzzling. While the internal structure of the class EXP induced by polynomial time reductions is well understood and very rich, these reducibilities do not impose any interesting structure on the class P. So how might be the internal structure of NP? Very poor or very rich? Any answer to these questions would, at least in part, settle the uncertainty about the true localization of NP within the deterministic time hierarchy.

So how to study the structure of NP in a situation where any interesting result would answer the $\mathrm{P} = \mathrm{NP}$ problem? Obviously, any considerations on this topic have to assume that NP is different from P, since otherwise there would be left no interesting question to ask. In addition, this is the most widely accepted assumption about NP, which makes results especially appealing that are based on this assumption only. Unfortunately, very little is known about the structure of NP under this assumption. Hence, for the time being, stronger hypotheses about NP are needed in order to obtain structural results.

This thesis studies the structure inside NP under hypotheses that reflect different ways to say that "NP is not small" and focuses on results that separate not only the different kinds of polynomial time reducibilities but also the completeness notions induced by them on the class NP. Mathematics provides two ways to talk about "small" sets, the measure-theoretic concept of measure zero sets and the topological category concept of meager sets. (See [Oxt80] for a deeper discussion of the relation of these concepts.) Inspired by the work of Lutz [Lut90, Lut92, Lut97], Mayordomo [LutM96, May94a, May94b] and others, where NP is studied under a measure-theoretic "non-smallness assumption", in this thesis a category-based approach as defined by Ambos-Spies in [Amb96] is used. This approach is based on the amalgamation of concepts of Ambos-Spies, Fleischhack and Hewig ([AmbFH88]) and Fenner ([Fen91]). One reason for the choice is that this approach results in more lucid and understandable proves. Moreover, the way how non-smallness assumptions allow for diagonalizations against polynomial time reductions seems to be most naturally reflected in a category-based setting.

1.1 Overview

The remainder of this chapter provides fundamental definitions and introduces a number of structural properties used in the sequel.

Chapter 2 introduces the resource-bounded category and measure concepts necessary for stating the non-smallness hypotheses for NP that are studied in the subsequent chapters. In relation to these concepts, resource-bounded randomness and genericity are defined, and some properties of p-random and p-generic sets are discussed.

In Chapter 3 we give and compare hypotheses about NP of different strength and summarize some known results that can be obtained under these hypotheses.

In Chapter 4 the relative strength of Lutz' measure theoretic non-smallness hypothesis and the category-based hypotheses that are used in this thesis are compared. This is done by defining an ordering on possible hypotheses about NP that is based on relativizations.

Finally, Chapter 5 provides the main contributions of this thesis, namely a detailed study of NP-completeness notions induced by various polynomial-time reductions. These results are divided into two main groups, depending on the strength of the hypothesis that is used. In addition, possible strengthenings of the results are discussed.

1.2 Preliminaries

In this section some fundamental concepts are introduced. We present these basic notions in an informal way. Exact definitions can be found in any textbook on computational complexity like [BalDG88, Pap94, BovC94, WagW86, HopU79].

1.2.1 Turing Machines

Turing machines are a theoretical model designed to cover the notion of computability. A Turing machine can be pictured as a machine equipped with a fixed number of *tapes* and for each tape with a corresponding *head* that moves left and right on this tape and may read and write *symbols* on it. The symbols are taken from some fixed, finite *alphabet* , denoted by Σ.

One of the tapes is declared as *input tape* and has a read-only head. The other tapes are the *working tapes.* The tapes are meant to be unlimited in both directions and divided into *cells.* Each cell can either contain one of the symbols from the alphabet or can be blank. One step of the machine consists of scanning the cells under the different heads, writing symbols in these cells and moving some (or all) of the heads one cell left or right. Which actions are to be carried out depends not only on the content of the scanned cells, but on the current *state* of the machine. This is controlled by a finite list of instructions, called *transitions* that determine, depending on the current state and content of the scanned cells, the action for each head and the state for the next step. There is an *initial state* and, possibly, several accepting states. In the beginning, the machine is in the initial state, all its working tapes contain only blanks, and the head for the input tape is scanning the leftmost non-blank cell of this tape.

A *computation* of a Turing machin M (on input x) is a (possibly infinite) sequence of complete descriptions called *configurations* of the individual consecutive steps performed by M when given input x. So each configuration describes the current content of the cells on the tapes, the position of every head and the current state of M, and two subsequent configurations have to match the corresponding transition.

The machine *stops* if there is no transition that matches the current combination of state and content of scanned cells. A finite string w, also called *word*, over Σ is *accepted* by Turing machine M, if M stops in an accepting state when w is written on its input tape. And the *language accepted by M* is the set of words accepted by M.

M represents a function f, if one of the working tapes of M is declared as *output tape*, and in case M stops on input x the word that is written on this tape then is interpreted as output $y = f(x)$. Note that M need not stop on all inputs, in which case f is said to be *undefined* on these inputs. f is called *partial recursive* function. If M stops on all inputs, the function computed by M is called *recursive* or *computable.*

Besides this machine model of computability, Turing also provided a concept that is of central importance for structural considerations, the *oracle Turing machines.* An oracle Turing machine is a Turing machine that is equipped with an extra *query tape* and can enter three additional states, called `QUERY`, `YES` and `NO`. Associated with M is a set, called *oracle* such that, whenever M enters the state `QUERY`, membership of the word written on the query tape (called the *query*) determines the next state of M. If the query is a member of the oracle, the next state is `YES`, otherwise it is `NO`. These states are also called *answers* of the oracle.

So a computation of an oracle Turing machine M depends on the oracle and instead of a single computation we can also consider the *computation tree* of M on input x, where each node represents a query. In general, the queries made by M will depend on the oracle answers to previous queries. In this case M is said to make *adaptive queries* or simply is called *adaptive* oracle Turing machine. Otherwise M is called *non-adaptive*, i.e., M always makes the same queries, independent of the oracle. In this case the actions of M can be pictured as first producing a list of queries, getting the list of corresponding oracle answers, and then proceeding the computation without further entering a query state.

If the list of transitions that controls the Turing machine M does not represent a function, but a relation, M is called *nondeterministic Turing machine.* In this case, for each input x there exists the *computation tree for M on input x*, where each path of this tree represents a possible computation, according to the list of transitions. Here x is *accepted* by M if there is at least one path in the computation tree that represents an accepting computation.

A Turing machine is completely determined by its list of transitions, which essentially is a finite piece of text, that can be coded into the natural numbers. If we fix such a coding, then any natural number either is a code for a specific Turing machine, or it codes garbage. If we interpret this garbage as the Turing machine with empty list of transitions, we can use the natural numbers as list of *all* Turing machines. Moreover, we can define a *universal* Turing machine, that given a word x and a number n decodes n and then simulates the coded Turing machine on input x.

This machine model of computability immediately yields a formal measurement of resources consumed by a computation. This feature, in special, made it so popular among computer scientists. The units of time are just the single steps that consist of applying (one of) the appropriate transitions from the list.

Thus the running time of a Turing machine M is the number of steps M may use on an input of length n.

The class of languages accepted by a (non)deterministic Turing machine with running time $r \in O(t(n))$ is denoted by DTIME(t(n)) (NTIME($t(n)$)).

We will consider the deterministic time classes

$$\begin{aligned}
\mathrm{P} &= \mathrm{DTIME}(\mathit{poly}) = \bigcup_{k\geq 1}\mathrm{DTIME}\left(n^k\right),\\
\mathrm{E} &= \mathrm{DTIME}\left(2^{lin}\right) = \bigcup_{k\geq 1}\mathrm{DTIME}\left(2^{kn}\right),\\
\mathrm{EXP} &= \mathrm{DTIME}\left(2^{poly}\right) = \bigcup_{k\geq 1}\mathrm{DTIME}\left(2^{n^k}\right), \text{and}\\
\mathrm{EE} &= \mathrm{DTIME}\left(2^{O(2^n)}\right) = \bigcup_{k\geq 1}\mathrm{DTIME}\left(2^{2^{n+k}}\right);
\end{aligned}$$

and their nondeterministic counterparts

$$\begin{aligned}
\mathrm{NP} &= \mathrm{NTIME}(\mathit{poly}) = \bigcup_{k\geq 1}\mathrm{NTIME}\left(n^k\right),\\
\mathrm{NE} &= \mathrm{NTIME}\left(2^{lin}\right) = \bigcup_{k\geq 1}\mathrm{NTIME}\left(2^{kn}\right),\\
\mathrm{NEXP} &= \mathrm{NTIME}\left(2^{poly}\right) = \bigcup_{k\geq 1}\mathrm{NTIME}\left(2^{n^k}\right), \text{and}\\
\mathrm{NEE} &= \mathrm{NTIME}\left(2^{O(2^n)}\right) = \bigcup_{k\geq 1}\mathrm{NTIME}\left(2^{2^{n+k}}\right).
\end{aligned}$$

In our notation we will not distinguish between this classes of languages and the corresponding classes of functions computed by Turing machines within the given time bound.

1.2.2 Polynomial Time Reducibilities

Oracle Turing machines can be used to make precise relations like "problem B is at most as hard to compute as problem A", thereby imposing a structure on the class of computable problems. Suppose there is an oracle Turing machine M such that the language accepted by M if given oracle A is the set B ($B = L(M, A)$). Then the problem of deciding membership in B is "reduced" to the problem of deciding membership in A via M, since the combination of M with a procedure for deciding A yields a procedure for deciding B, where the queries of M to the oracle A are replaced by computations for A. In this case M is called *reduction from B to A*. The reductions used in this thesis are the following:

Definition 1.2.1. Let A, B be sets. We write

1. $A \leq_T^{\mathrm{p}} B$ if $A = L(M, B)$ for some polynomial time-bounded oracle Turing machine M. (A *P-T*-reduces to B via M.)
2. $A \leq_{tt}^{\mathrm{p}} B$ if $A \leq_T^{\mathrm{p}} B$ via a non-adaptive oracle Turing machine M. (A *P-tt*-reduces to B via M.)
3. $A \leq_{btt}^{\mathrm{p}} B$ if $A \leq_{tt}^{\mathrm{p}} B$ via some M that makes a constant number of queries, which is called the norm of M. (A *P-btt*-reduces to B via M.)

4. $A \leq^{\mathrm{P}}_{btt(k)} B$ if $A \leq^{\mathrm{P}}_{btt} B$ via some M of norm k. (A *P-btt(k)*-reduces to B via M.)

5. $A \leq^{\mathrm{P}}_{bT(k)} B$ if $A \leq^{\mathrm{P}}_{T} B$ via some M that makes at most k queries for some constant $k \geq 1$. (A *P-bT(k)*-reduces to B via M.)

6. $A \leq^{\mathrm{P}}_{m} B$ if there is a function g in P such that for all inputs x $A(x) = 1 \iff B(g(x)) = 1$. ($A$ *P-m*-reduces to B via M.)

7. $A \leq^{\mathrm{P}}_{1} B$ if $A \leq^{\mathrm{P}}_{m} B$ via some one-to-one function g. (A *P-1*-reduces to B via M.)

Definition 1.2.2. Let $r \in \{m, btt(k), btt, tt, T\}$ and let $\mathbf{C}$ be a complexity class. A set A is *hard* for $\mathbf{C}$ under *P-r*-reductions ($\mathbf{C}$-r-hard for short) if $B \leq^{\mathrm{P}}_{r} A$ for all $B \in \mathbf{C}$. And A is *complete* for $\mathbf{C}$ under *P-r*-reductions ($\mathbf{C}$-r-complete for short) if $A \in \mathbf{C}$ and $B \leq^{\mathrm{P}}_{r} A$ for all $B \in \mathbf{C}$.

P-T-completeness was introduced by Cook [Coo71] and *P-m*-completeness by Karp [Kar72] and Levin [Lev73]. Ladner, Lynch and Selman compared the strength of the polynomial time reducibilities in [LadLS75]. They showed that

$$P\text{-}1 \prec P\text{-}m \prec P\text{-}btt(1) \prec P\text{-}btt \prec P\text{-}tt \prec P\text{-}T$$

where $r \prec r'$ indicates that r is strictly stronger than r'. I.e., $A \leq^{\mathrm{P}}_{r} B$ implies $A \leq^{\mathrm{P}}_{r'}$ for all sets A and B, while there are sets A and B such that A is *P*-r'-reducible to B, but is not *P-r*-reducible to B. A great deal of attention was paid to the completeness notions induced by these reducibilities for complexity classes like NP, E, EXP, NE, or NEXP. For E, EXP, NE, and NEXP the relations among the completeness notions are completely determined. Namely Ko and Moore [KoM81] and Watanabe [Wat87] showed that *P-btt(k)*-, *P-btt(k+1)*-, *P-btt*-, *P-tt*-, and *P-T*-completeness can be separated for E and EXP, whereas by results of Berman [Ber76] and Homer, Kurtz and Royer [HomKR93] *P-1*-, *P-m*- and *P-btt(1)*-completeness coincide. Buhrman [Buh93] proved these results for other complexity classes containing E like e.g., NE and NEXP (see Buhrman and Torenvliet [BuhT94] for a survey on these results). Very little is known about NP-complete sets under these reduciblities, unless P = NP. Chapter 3 provides a summary of results about NP under various hypotheses.

1.2.3 Notation

For the larger part the notation is standard. $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{Q}^+$, $\mathbb{R}$, and $\mathbb{R}^+$ are the sets of natural numbers, (nonnegative) rational numbers and (nonnegative) real numbers, resp. $\exists^\infty x$ and $\forall^\infty x$ denote "there are infinitely many x" and "for all but finitely many x", respectively.

The binary alphabet $\{0, 1\}$ is denoted by Σ and $\Sigma^* = \{0, 1\}^*$ denotes the set of finite binary strings, also called words. The subsets of Σ^* are called languages, or problems, or simply sets. Sets of languages are called classes. Capital letters A, B, C, ... denote sets, boldface capital letters $\mathbf{A}$,$\mathbf{B}$, $\mathbf{C}$, ... denote classes.

Lower case letters $..., x, y, z$ from the end of the alphabet are used to denote strings, while the other letters denote numbers, with the exception of d, f, g, h and s, t which also denote functions.

The capital letters M and N are reserved for Turing machines. $M(x) = 1$, resp. $M(x) = 0$ indicates that, given input x, the Machine M halts in an accepting (resp. rejecting) state. In addition, $M^A(x) = 1$, $(M^A(x) = 0)$ indicates that M is an oracle Turing machine and is equipped with the oracle set A. Thus $L(M, A) = \{x : M^A(x) = 1\}$.

For a string x, $x(m)$ denotes the $(m+1)$th bit in x, i.e., $x = x(0)...x(n-1)$, where $n = |x|$ is the length of x. (Sometimes the notation $x[m]$ is used as well, for the sake of better readability.) λ is the empty string. vw is the concatenation of the strings v and w and w^n is the n-fold iteration of w. $x \sqsubseteq y$ $(x \sqsubset y)$ denotes that the string x is a (proper) prefix of y.

We identify strings with numbers by letting n be the $(n + 1)$st string under the canonical length-lexicographic ordering on strings. Occasionly, this string is denoted by s_n. Hence for two strings x and y, $x < y$ if $|x| < |y|$ or $|x| = |y|$ and x is lexicographical less than y. And if y is the n-th successor of x in this ordering, y is denoted by $x + n$.

Sets are identified with their characteristic function, i.e., $x \in A$ iff $A(x) = 1$. The characteristic sequence of a set A is the infinite sequence $A(s_0)A(s_1)\ldots$, which is also identified with the set A. Initial segments of this sequence of length n are interchangably denoted by $A \restriction n$ or $A \restriction s_n$. We write $x \sqsubset A$ and say that A *extends* x if x is a finite initial segment of the characteristic sequence of A. If $\{y_n : n \geq 0\}$ is a sequence of strings such that $y_n \sqsubset y_{n+1}$ for all $n \geq 0$ the unique set extending all strings y_n is denoted by $\lim_{n\geq 0} y_n$.

So strings are used in two different meanings: as elements of sets and as finite initial segments of sets. In an attempt to avoid confusion, usually the notation $X \restriction x$ is used for strings intended to denote initial segments. Then $X \restriction x$ denotes a string of length x and, for $y < x$, $X(y)$ or $(X \restriction x)(y)$ will denote the $(y+1)$th bit of $X \restriction x$. If used as an oracle, $X \restriction x$ is interpreted as $(X \restriction x)0^\infty$, i.e., as the finite set $\{z < x : X(z) = 1\}$.

$|A|$ is the cardinality of the set A. For a number n, $A^{=n}$ $(A^{\leq n})$ is the set of strings in A of length (at most) n. (In the case of $A = \Sigma^*$ the $*$ is omitted then.) For sets A and B, $A \oplus B = \{0x : x \in A\} \cup \{1x : x \in B\}$ is the join (effective disjoint union) of A and B, $\overline{A} = \Sigma^* \setminus A$ is the complement of A and $A \Delta B = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference of A and B.

For a class $\mathbf{A}$, $\mathbf{A}^c$ is the complement of $\mathbf{A}$ and *co*-$\mathbf{A} = \{A : \overline{A} \in \mathbf{A}\}$ is the dual class of $\mathbf{A}$.

For a function f, domain and range of f are denoted by $dom(f)$ and $rng(f)$. For a partial recursive function f, $f(x)\downarrow$ ($f(x)\downarrow = y$) indicates that f halts on input x (and outputs y). If f is total and length-increasing and A is a set, the set $A_f = \{x : f(x) \in A\}$ is called extraction of A. Two extractions A_{f_1} and A_{f_2} are independent, if $rng(f_1) \cap rng(f_2) = \emptyset$.

Lower case Greek letters denote finite partial functions from Σ^* to $\{0,1\}$. We say β *extends* α ($\alpha \subseteq \beta$) if the graph of α is contained in the graph of β and we say β *extends* α *along* γ (denoted by $\beta = \alpha \sqcup \gamma$) if $\beta(x) = \alpha(x)$ for $x \in dom(\alpha)$ and $\beta(x) = \gamma(x)$ for all $x \in dom(\gamma) \setminus dom(\alpha)$. Similarly, a set X extends α ($\alpha \subset X$) if α coincides with the characteristic function of X on $dom(\alpha)$. If used as an oracle, a finite function α is interpreted as the finite set $\{x \in dom(\alpha) : \alpha(x) = 1\}$. So, for a query $x \notin dom(\alpha)$, the oracle α returns the answer 0.

1.2.4 Structural Properties

In this section we summarize structural properties of sets that are used in the subsequent chapters. In the following let A be a set and $\mathbf{C}$ a complexity class.

A first characteristic of a set is the distribution of its members. The *census function* of set A gives, for given n, the total number of members of A of length at most n and is denoted by

$$c_A(n) = |\{x \in A : |x| \leq n\}| = |A^{\leq n}|.$$

A is called *tally* if $A \subseteq \{0\}^*$. If, like in this case, the census function of A is bounded by a polynomial, then A is called *sparse*. On the other hand, A is of *exponential density*, if there is a constant $\epsilon > 0$ such that $c_A(n) > 2^{n^\epsilon}$ for allmost all n. And A has *exponential gaps* if there are infinitely many n such that $A \cap \Sigma^{=n} = \emptyset$.

The following properties are related to the complexity class $\mathbf{C}$.

Definition 1.2.3. Let A be a set. A is $\mathbf{C}$-*immune* if A is infinite and every subset of A is either finite or not a member of $\mathbf{C}$. A is $\mathbf{C}$-*bi-immune* if A and $\overline{A}$ are $\mathbf{C}$-immune.

Proposition 1.2.4. *Let A be an infinte and coinfinite set. $A \notin \mathbf{C}$ if A is $\mathbf{C}$-immune. A is $\mathbf{C}$-bi-immune iff $A \cap C \neq \emptyset$ and $\overline{A} \cap C \neq \emptyset$ for all infinite $C \in \mathbf{C}$.*

Definition 1.2.5. Let A be a set and f a function in $\mathbf{C}$. f *agrees with* A if $f(x) = f(y)$ implies that $A(x) = A(y)$. Let the *collision set of* f be

$$Coll_f = \{x : \exists y < x : f(x) = f(y)\}$$

A is $\mathbf{C}$-*incompressible* if every function in $\mathbf{C}$ that agrees with A has a finite collision set.

Lemma 1.2.6. *([BalS85]) Let* $\mathbf{C}$ *be a class closed under complement. Then every* $\mathbf{C}$*-incompressible set is* $\mathbf{C}$*-bi-immune.*

Proof. Let A be a $\mathbf{C}$-incompressible set and $C \in \mathbf{C}$. Suppose that $A \cap C = \emptyset$. Since $\mathbf{C}$ is closed under complement, the function $f : \Sigma^* \to \Sigma$ defined by $f(x) = x$ if $x \notin C$ and $f(x) = a_0$ if $x \in C$ for some fixed $a_0 \notin A$ is in $\mathbf{C}$ and agrees with A. Hence the collision set of f is finite. But this implies that C must be finite as well. Similarly, C is finite if $\overline{A} \cap C = \emptyset$. □

Definition 1.2.7. A is *autoreducible* if there is a polynomial-time bounded oracle Turing machine M such that, for every input x, $A(x) = M^A(x)$ and M never queries its input. And A is *selfreducible* if there is a polynomial-time bounded oracle Turing machine M such that, for every input x, $A(x) = M^A(x)$ and all the queries made by M are of length shorter than the length of the input.

Chapter 2

Resource-Bounded Category and Measure

2.1 Classical Category and Measure

This section recapitulates two classical classification schemes for subclasses of the Cantor space 2^ω. Being quantitative classification schemes, these allow for a distinction between "small" and "large" classes of languages. In either scheme, a class appears "small" if it may be considered "negligible" in the related setting. Besides the usual definitions, both schemes will be given a game-theoretic characterization. I.e., given a class $\mathbf{A}$, for each classification scheme there will be a game associated with $\mathbf{A}$ such that $\mathbf{A}$ is "small" if and only if there is a winning strategy for one particular player. Since a strategy is a function determining the next move of the player, resource-bounded versions of these classification schemes later are obtained by emposing appropriate bounds on the corresponding strategies.

2.1.1 Baire Category and Banach-Mazur-Games

The first classification scheme, Baire Category, is given in terms of the standard topology on the Cantor space.

Definition 2.1.1. For any string x, the class $\mathbf{B}_x = \{A : x \sqsubset A\}$ is *basic open.* A class is *open* if it is the union of basic open classes.

From the viewpoint of topology, the nowhere dense classes are negligible. So here the "small" classes, called *of first category* or *meager*, are defined as follows.

Definition 2.1.2. A class $\mathbf{A}$ is *dense* if $\mathbf{A}$ intersects all basic open classes; and $\mathbf{A}$ is *nowhere dense* if $\mathbf{A}$ is contained in the complement of an open and dense class.

$\mathbf{A}$ is *meager*, if $\mathbf{A}$ is the countable union of nowhere dense classes; and $\mathbf{A}$ is *comeager*, if the complement of $\mathbf{A}$ is meager.

Remark 2.1.3. A class $\mathbf{A}$ is nowhere dense iff for every string x there is a string y such that $\mathbf{B}_{xy} \cap \mathbf{A} = \emptyset$.

Proposition 2.1.4. *Let $\mathbf{A}$, $\mathbf{A}'$, and $\mathbf{A}_e$ $(e \geq 0)$ be classes. Let A be a set.*

(i) If $\mathbf{A}$ is meager and $\mathbf{A}' \subseteq \mathbf{A}$ then $\mathbf{A}'$ is meager; and if $\mathbf{A}$ is comeager and $\mathbf{A} \subseteq \mathbf{A}'$ then $\mathbf{A}'$ is comeager.

(ii) If the classes $\mathbf{A}_e$ $(e \geq 0)$ are meager, then $\bigcup_{e\geq 0} \mathbf{A}_e$ is meager; and if the classes $\mathbf{A}_e$ $(e \geq 0)$ are comeager, then $\bigcap_{e\geq 0} \mathbf{A}_e$ is comeager

(iii) $\{A\}$ is meager (in fact, nowhere dense).

Proof. (i) and (ii) are immediate by definition. And (iii) follows from the above remark about nowhere dense classes. □

Note that the closure under countable unions 2.1.4 (ii) holds for meager classes, but in general not for nowhere dense classes. 2.1.4 (ii) and 2.1.4 (iii) imply that any countable class is meager.

The consistency of this quantification scheme is shown in Baire's Theorem, which will be given here in the following form.

Theorem 2.1.5. *No basic open class is meager.*

Proof. Fix a string x and assume that $\mathbf{B}_x = \bigcup_{e\geq 0} \mathbf{A}_e$ is the countable union of nowhere dense classes $\mathbf{A}_e$. For any given basic open set $\mathbf{B}_y$ we inductively define a sequence y_e $(e \geq 0)$. Let $y_0 = y$ and let y_{e+1} be the least string extending y_e such that $\mathbf{A}_e \cap \mathbf{B}_{y_{e+1}} = \emptyset$. This sequence exists since all classes $\mathbf{A}_e$ are nowhere dense. Then the limit set $Y = \lim_{e\geq 0} y_e$ extends y and is not a member of any of the classes $\mathbf{A}_e$, hence Y is a member of $\mathbf{B}_y \setminus \mathbf{B}_x$. Since y was chosen arbitrarily, this implies that the complement of $\mathbf{B}_x$ is dense, a contradiction. (Recall that $\mathbf{C}$ is dense if $\mathbf{C}$ intersects all basic open classes.) □

Corollary 2.1.6. *The Cantor Space 2^ω is not meager. No class can be both meager and comeager.*

Proof. This follows from Theorem 2.1.5 by Proposition 2.1.4. □

An alternative characterization of meager classes can be given by Banach-Mazur-Games. Associated with a given class $\mathbf{A}$, consider the following game for two players (A) and (B). Player (A) starts by choosing a string y_1. Then (B) chooses a string $y_2 \sqsupset y_1$, followed by the choice $y_3 \sqsupset y_2$ of (A), and so on. This way a unique set Y, that extends all strings y_n $(n \geq 0)$, is defined. Player (A) wins if $Y \in \mathbf{A}$, and player (B) otherwise. When Mazur introduced these games around 1928, he conjectured that player (B) has a winning strategy if and only if the class $\mathbf{A}$ is meager. This later was proven by Banach.

Definition 2.1.7. Let $\mathbf{A}$ be a class. A *winning strategy* for player (B) in the Banach-Mazur-Game associated with $\mathbf{A}$ is a family of functions $f_n : \Sigma^* \to \Sigma^*$ $(n \geq 1)$ such that, firstly, f_n is defined on all sequences $\langle y_1, \ldots y_{2n-1} \rangle$ satisfying

$$y_1 \sqsubset \ldots \sqsubset y_{2n-1} \quad \text{and} \tag{2.1}$$

$$y_{2i} = f_i(\langle y_1, \ldots y_{2i-1} \rangle) \quad \text{for all } 1 \leq i < n. \tag{2.2}$$

In this case, $f_n(\langle y_1, \ldots y_{2n-1} \rangle)$ extends y_{2n-1}. And secondly, for each sequence $(y_n)_{n\geq 1}$ satisfying (2.1) and (2.2), $\lim_{n \geq 1} y_n \notin \mathbf{A}$.

Theorem 2.1.8. *Let $\mathbf{A}$ be a class. In the associated Banach-Mazur-Game, player (B) has a winning strategy iff $\mathbf{A}$ is meager.*

Proof. For one direction, consider a class $\mathbf{A}$ and let $(f_n)_{n\geq 1}$ be a winning strategy for player (B). For every $n \geq 1$ and every sequence of natural numbers $i_1, \ldots, i_n \geq 1$ inductively define strings $x_{i_1 \ldots i_n}$ and $z_{i_1 \ldots i_n}$ by

$n = 1$: Let $x_1 = 0$, $z_1 = f_1(0)$. For every $i > 1$ let x_i be the least string w such that $\mathbf{B}_w \cap \bigcup_{1 \leq j < i} \mathbf{B}_{z_j} = \emptyset$, and let $z_i = f_1(x_i)$.

$n > 1$: Suppose that $x = x_{i_1 \ldots i_{n-1}}$ and $z = z_{i_1 \ldots i_{n-1}}$ are allready fixed. Abbreviate $i_1 \ldots i_{n-1}$ by s. Let $x_{s1} = z0$ and $z_{s1} = f_n(\langle x_{i_1}, z_{i_1}, \ldots, z, x_{s1} \rangle)$. For $i > 1$ let x_{si} be the least string $w \sqsupset z$ such that $\mathbf{B}_w \cap \bigcup_{1 \leq j < i} \mathbf{B}_{z_{sj}} = \emptyset$. And again, $z_{si} = f_n(\langle x_{i_1}, z_{i_1}, \ldots, z, x_{si} \rangle)$.

This way two families of strings are defined such that for every $n \geq 1$ and every finite sequence $i_1, \ldots, i_n$ of natural numbers it holds that

(i) $z_{i_1 \ldots i_n} = f_n(\langle x_{i_1}, z_{i_1}, \ldots, z_{i_1 \ldots i_{n-1}}, x_{i_1 \ldots i_n} \rangle)$;

(ii) $x_{i_1 \ldots i_n} \sqsupset z_{i_1 \ldots i_{n-1}}$;

(iii) the set $\{z_{i_1 \ldots i_n} : i_1, \ldots, i_n \geq 1\}$ is prefix-free;

(iv) the class $\mathbf{G}_n = \bigcup_{i_1, \ldots, i_n \geq 1} \mathbf{B}_{z_{i_1 \ldots i_n}}$ is open and dense.

By (i) and (ii), for any infinite sequence $(i_n)_{n\geq 1}$ of positive natural numbers $x_{i_1}, z_{i_1}, x_{i_1 i_2}, z_{i_1 i_2}, \ldots$ represents a possible "run" in the Banach-Mazur-Game associated with $\mathbf{A}$ whereby player (B) applies strategy $(f_n)_{n\geq 1}$. By hypothesis about this strategy, the set

$$\lim_{n \to \infty} x_{i_1 \ldots i_n} = \lim_{n \to \infty} z_{i_1 \ldots i_n} \notin \mathbf{A}. \tag{2.3}$$

Now let $\mathbf{D} = \bigcap_{n \geq 1} \mathbf{G}_n$ (see (iv)). By (iii), for every $X \in \mathbf{D}$, there is a unique sequence $\hat{\imath}_1, \hat{\imath}_2, \ldots$ such that $X \sqsupset x_{\hat{\imath}_1 \ldots \hat{\imath}_n}$ for every $n \geq 1$. Hence $X \notin \mathbf{A}$ by (2.3). But this implies that

$$\mathbf{A} \subseteq \mathbf{D}^{\mathbf{c}} = \bigcup_{n \geq 1} \mathbf{G}_n^{\mathbf{c}}$$

whence $\mathbf{A}$ is meager by (iv).

The other direction is similar to the proof of Theorem 2.1.5 and quite straightforward. (If $\mathbf{A}$ is meager, then player (B) can choose y_{2i} such that $\mathbf{B}_{y_{2i}} \cap \mathbf{C}_i = \emptyset$, if $\mathbf{A}$ is the union of the nowhere dense classes $(\mathbf{C}_i)_{i \geq 1}$.) □

2.1.2 Lebesgue Measure on the Cantor Space

The classical Lebesgue measure μ on the Cantor space is induced by the equiprobable measure ν on $\{0,1\}$ which assigns to both 0 and 1 the probability 1/2. Thus μ is the completed product measure induced by ν (see [Fel86] or [Oxt80] for details). For example, for the basic open class $\mathbf{B}_x = \{A : x \sqsubset A\}$, $\mu(\mathbf{B}_x) = 2^{-|x|}$. A model for this measure is the (independent) tossing of a fair coin. So $\mu(\mathbf{B}_x) = 2^{-|x|}$ is the probability that a randomly chosen sequence of coin tosses begins with the events $x(0), \ldots, x(|x|-1)$. The "small" classes with respect to measure are the measure-0 classes.

Definition 2.1.9. An infinite sequence $\mathcal{B} = \{\mathbf{B}_{x_n} : n \geq 0\}$ of basic open sets is an *ε-cover* of a class $\mathbf{C}$ if

$$\mathbf{C} \subseteq \bigcup_{n \geq 0} \mathbf{B}_{x_n} \qquad \text{and} \qquad \sum_{n \geq 0} \mu(\mathbf{B}_{x_n}) \leq \varepsilon.$$

Example 2.1.10. For any set A, the sequence $\mathcal{B}_A = \{\mathbf{B}_{A \restriction n} : n > k\}$ is a 2^{-k}-cover of $\{A\}$.

Definition 2.1.11. A class $\mathbf{C}$ has *measure* 0 ($\mu(\mathbf{C}) = 0$) if, for all $n \geq 0$, there is a 2^{-n}-cover of $\mathbf{C}$. A class $\mathbf{C}$ has *measure* 1 ($\mu(\mathbf{C}) = 1$) if the complement $\mathbf{C}^{\mathbf{c}}$ of $\mathbf{C}$ has measure 0.

We write $\mu(\mathbf{C}) \neq i$ ($i = 0, 1$) to indicate that $\mathbf{C}$ does not have measure i. Note that, for any cover $\mathcal{B} = \{\mathbf{B}_{x_n} : n \geq 0\}$ of the power class of $\{0,1\}^*$, $\sum_{n \geq 0} \mu(\mathbf{B}_{x_n}) \geq 1$. Moreover, μ is consistent in the sense that for no class $\mathbf{C}$, $\mu(\mathbf{C}) = 0$ and $\mu(\mathbf{C}) = 1$.

Lemma 2.1.12. *Let A be a set, $\mathbf{C}$ and $\mathbf{D}$ be classes. Then*

(i) $\mu(\{A\}) = 0$.

(ii) If $\mathbf{C} \subseteq \mathbf{D}$ and $\mu(\mathbf{D}) = 0$ then $\mu(\mathbf{C}) = 0$.

(iii) Let $\mathbf{C} = \bigcup_{n \geq 0} \mathbf{C}_n$ be the union of the measure-0 classes $\mathbf{C}_n$, $n \geq 0$. Then $\mu(\mathbf{C}) = 0$.

Proof. (i) follows from Example 2.1.10 while (ii) is immediate by definition. For a proof of (iii), fix $m \geq 0$. To show that $\mathbf{C}$ possesses a 2^{-m}-cover, by assumption choose $2^{-(m+k+1)}$-covers $\mathcal{B}_k = \{\mathbf{B}_{x_{k,n}} : n \geq 0\}$ of $\mathbf{C}_k$. Then, for $\hat{x}_{\langle k,n \rangle} = x_{k,n}$, $\mathcal{B} = \{\mathbf{B}_{\hat{x}_e} : e \geq 0\}$ is a 2^{-m}-cover of $\mathbf{C}$. □

Corollary 2.1.13. *For any countable class* $\mathbf{C}$, $\mu(\mathbf{C}) = 0$. □

Ville ([Vil39]) gave an equivalent characterization of the measure-0 classes based on betting strategies in a fair betting game. Consider a gambler which bets on the consecutive bits of a hidden sequence in 2^ω. Depending on the string w of bits revealed in the previous rounds, she distributes her capital among the two possible outcomes 0 and 1. Then the next bit is revealed and the capital bet on the correct guess is doubled, while the other is lost. The gambler wins in this game if her capital is unbounded as the game proceeds.

Definition 2.1.14.

(a) A *martingale* is a function $d : \Sigma^* \to \mathbb{R}^+$ such that $d(\lambda) > 0$ and, for every $w \in \Sigma^*$,

$$d(x0) + d(x1) \leq 2 \cdot d(x). \tag{2.4}$$

$d(\lambda)$ is the *norm* of d. And d is *normed* if $d(\lambda) = 1$.

(b) A martingale d *succeeds* on a set A if

$$\limsup_{n \geq 0} d(A \restriction n) = \infty.$$

$S^\infty[d]$ denotes the class of sets on which the martingale d succeeds. A martingale d *succeeds* on a class $\mathbf{C}$ if $\mathbf{C} \subseteq S^\infty[d]$.

We can interpret the martingale d as the function that records the current capital $d(x)$ of the gambler after the consecutive outcomes $x(0), \ldots, x(|x|-1)$. Then d is related to the "strategy" of the gambler, since $d(xi)/(2 \cdot d(x))$ is the fraction of the current capital she bets on the outcome $i \leq 1$, after she has seen the string x of previous outcomes.

Martingales and their success sets are related to measure zero sets, analog to the relation between Banach-Mazur-Games and meager classes.

Theorem 2.1.15. *[Vil39] The class* $\mathbf{C}$ *has measure zero iff there is a martingale that succeeds on every set in* $\mathbf{C}$.

2.2 Resource-Bounded Genericity

This section presents the central definitions that will be used to obtain the separation results in Chapter 5. Instead of the measure theoretic approach of Lutz, there a category-based approach is applied to hypothesize non-smallness of NP. Since category can be seen as direct formalization of diagonalizations, it's preference is quite natural for the investigation of polynomial time reductions.

The basic notion of this approach are *time bounded extension functions.* Extension functions are designed to reflect finite extension methods to enforce a certain property. However, a direct imposing of time bounds to classical extension functions, as was done by Lutz in [Lut90], yields a type of category that is too weak for our purposes (see [Amb96] for details). The extension functions considered here are (possibly) partial, and do not fix the extension completely, but only on selected and (possibly) wide-spread bits. The definition below was developed by Ambos-Spies in [Amb96] combining features of the category concepts of Ambos-Spies, Fleischhack and Huwig [AmbFH88] and Fenner [Fen91].

Definition 2.2.1. ([Amb96], [AmbFH88], [Fen91]) Let $t(n) \geq n$ be a recursive function.

1. A partial function f: $\{0,1\}^* \to (\{0,1\}^* \times \{0,1\})^*$ such that $f \in \mathrm{DTIME}(t(n))$ is a $t(n)$-*extension function* if, whenever f is defined on input $X \restriction n = X(0) \dots X(n-1)$,
$$f(X \restriction n) = (y_0, i_0), \dots, (y_m, i_m) \tag{2.5}$$
for some $m \geq 0$, some strings y_j with $n \leq y_0 < \cdots < y_m$, and some $i_j \in \{0,1\}$ $(0 \leq j \leq m)$.

2. A *simple* $t(n)$-*extension function* f is a $t(n)$-extension function f such that, whenever $f(X \restriction x)\downarrow$, then $f(X \restriction x) = (x, i)$ for some $i \leq 1$, in which case we also write $f(X \restriction x) = i$.

3. A $t(n)$-extension function f is *bounded* if there is a constant number $k \geq 1$ such that, whenever (2.5) applies, i.e., f is defined, then $m < k$. If k is known, f is also called k-*bounded.*

4. f is a *waiting, bounded* $t(n)$-*extension function* if the domain of f is a subset of
$$Init_0 := \{X \restriction 0^n \,:\, X \subseteq \Sigma^*, n \geq 1\},$$
and there is a constant $l \geq 1$ such that, whenever $f(\alpha)$ is defined for $\alpha \in Init_0$ of length $2^n - 1$, then
$$f(\alpha) = (y_{\alpha,1}, i_{\alpha,1}), \dots, (y_{\alpha,l_\alpha}, i_{\alpha,l_\alpha}) \tag{2.6}$$
where $l_\alpha \leq l$, $0^n \leq y_{\alpha,1} < \cdots < y_{\alpha,l_\alpha}$, $pos(\alpha) = (y_{\alpha,1}, \dots, y_{\alpha,l_\alpha})$ is computable in $t(2^n - 1)$ steps and $i_{\alpha,j}$ is computable in $t(2^{|y_{\alpha,j}|} - 1)$ steps.

If neither a time bound for f nor any particular properties are specified, the general term *extension function* will be used for f. The next definition is an example for such a general setting.

Definition 2.2.2. Let f be an extension function and A a set.

1. A *meets* the extension function f *at* n if $f(A \upharpoonright n)$ is defined, say (2.5) holds for $X = A$, and $A(y_j) = i_j$ for $j \leq m$; and A *meets* f if A meets f at some n.
2. The extension function f is *dense along* A if $f(A \upharpoonright n) \downarrow$ for infinitely many n.

This definition plays the key role in the understanding of generic sets and their relation to finite extension methods. Every time f is defined on an initial segment $\alpha = A \upharpoonright n$ of the characteristic sequence of A, $f(\alpha)$, as given in (2.5) specifies a set of finite extensions of α, $B = \{\beta \sqsupset \alpha : \beta(y_j) = i_j \text{ for all } 0 \leq j \leq m\}$. Now A meets f at n iff there is a $\beta \in B$ such that $\beta \sqsubset A$. This way finite extension strategies are modelled into extension functions. The advantage of this approach lies, on the one hand, in locality (A has to meet the extension function only once), and coordination of requirements on the other (A can meet conflicting extension functions by choosing appropriate "moments" for each). This is reflected in the restriction on dense extension functions, since density of f along A ensures that there are infinitely many "chances" for A to meet f. In the following, we will use two different genericity notions derived from the above definitions.

Definition 2.2.3. Let $t(n) \geq n$ be a recursive function.

1. A set G is *general* $t(n)$*-generic* if G meets every $t(n)$-extension function f which is dense along G; and G is *general p-generic* if G is general n^k-generic for all $k \geq 1$.
2. A set G is $t(n)$*-generic* if G meets every simple $t(n)$-extension function f which is dense along G; and G is *p-generic* if G is n^k-generic for all $k \geq 1$.
3. A class $\mathbf{M}$ is *general p-meager* if, for some $k \geq 1$, $\mathbf{M}$ does not contain any general n^k-generic set.
4. A class $\mathbf{M}$ is *p-meager* if, for some $k \geq 1$, $\mathbf{M}$ does not contain any n^k-generic set.

The difference between $t(n)$-generic and general $t(n)$-generic sets lies in the reach of the corresponding extension functions. While a simple $t(n)$-extension function only specifies the next bit, in general the only limit on the length and the number of strings $y_0, \ldots, y_m$ for which bits $i_0, \ldots, i_m$ are specified by an extension function is the time bound $t(n)$. Presenting the input to a $t(n)$-extension function in the form $X \upharpoonright y$ for some $y \in \Sigma^*$ implies that

$$2^{|y|} - 1 \leq |X \upharpoonright y| < 2^{|y|+1}. \tag{2.7}$$

Hence, measured in the length of the string y, an n^k-extension function f can use $O(2^{k \cdot |y|})$ many steps in order to compute $f(X \upharpoonright y)$. While this bound does not suffice to specify bits for *all* strings of some length polynomial in $|y|$, it allows for specifications for polynomial many such strings and for a profound access to the input. The next Theorem is an example for this kind of use of the input.

Theorem 2.2.4. *([Amb96]) Let* $\mathbf{C}$ *be a complexity class that is closed under P-m-reductions. Then the following are equivalent:*

(1) $\mathbf{C}$ *is not p-meager.*

(2) There is an n^2*-generic set in* $\mathbf{C}$.

(3) There is an n^k*-generic set in* $\mathbf{C}$ *for every* $k \geq 1$.

(4) There is a p-generic set in $\mathbf{C}$.

(5) There is a $2^{\log^k n}$*-generic set in* $\mathbf{C}$ *for every* $k \geq 1$.

Proof. By definition, (5) implies (4), (3) and (2); and (1) is equivalent to (3). So assume that there is an n^2-generic set A in $\mathbf{C}$. For each $k \geq 1$ let

$$\begin{aligned} A_k &= \{x : 0^{k \cdot |x|} 1x \in A\}, \\ B_k &= \{x : 0^{|x|^{k+1}} 1x \in A\}, \end{aligned}$$

and, for some nondecreasing function with unbounded range $f : \mathbb{N} \to \mathbb{N}$ computable in polynomial time with respect to the unary representation, let

$$A_f = \{x : 0^{f(|x|) \cdot |x|} 1x \in A\}.$$

These sets are P-m-reducible to A, hence members of $\mathbf{C}$. In [Amb96] it is shown that for each $k \geq 1$, A_k is n^k-generic, A_f is p-generic, and B_k is $2^{\log^k n}$-generic. The idea is to assume a dense extension function that is not met by the related set and which thus can be transformed into an n^2-extension function that is not met by A. $\square$

In contrast to this result on the $t(n)$-genericity of sets P-m-reducible to an n^2-generic set, the analog does not hold for sets to which an n^2-generic set can be reduced.

Theorem 2.2.5. *([JueL95, AmbNT96]) If A is an n^k-generic set for some $k \geq 2$ and $A \leq^{\mathrm{p}}_{m} C$, $A \leq^{\mathrm{p}}_{btt(c)} B$ for some sets $B, C \in \mathrm{DTIME}(2^{dn})$, then B is not $n^{(d+1)(c+1)}$-generic and C is not n^{d+1}-generic.*

Corollary 2.2.6. *If A is a p-generic set, then A is not P-btt-hard for* E.

Proof. By the following Corollary 2.2.8, there is an n^2-generic set in E. Hence the Theorem implies that an *P-btt*-hard set for E is not p-generic . □

Besides these relative results on the complexity of $t(n)$-generic sets, Ambos-Spies, Neis and Terwijn have shown the following strong existence theorem for $t(n)$-generic sets.

Theorem 2.2.7. *([AmbNT96]) Let $t(n)$, $t'(n)$ and $f(n)$ be nondecreasing functions on $\mathbb{N}$ such that $t(n)$ and $t'(n)$ are time-constructible, $t(n), t'(n) \geq n$, $f(n)$ is polynomial time computable with respect to the unary representation of numbers, and the range of f is unbounded. Let B be a set in* $\mathrm{DTIME}(t'(n))$*. Then there is a $t(n)$-generic set A such that*

$$A \in \mathrm{DTIME}(2^{n+1}(t'(n) + n^2 t(2^{n+1}) \log t(2^{n+1})))$$

and for any $n \geq 0$

$$\|(A \triangle B) \cap \Sigma^{=n}\| \leq f(n).$$

Sketch of proof. Fix an effective enumeration $\{f_e : e \geq 0\}$ of simple $t(n)$-extension such that the set

$$F = \{0^e 1xi : f_e(x)\downarrow = i\}$$

is computable in time $O(e \cdot t(2^{n+1}) \log t(2^{n+1}) + e)$ for $n = |x|$. Note that A will be $t(n)$-generic if the following requirements are met:

$$R_e : f_e \text{ dense along } A \Rightarrow A \text{ meets } f_e \tag{2.8}$$

This will be ensured by constructing the set A in stages. At each stage s, membership of the string z_s in A is decided, whence at the end of stage s $\alpha_s = A \restriction z_{s+1}$ is fixed. At the same time, a set Sat_s is fixed which records the requirements met at the end of stage s. At each stage s the minimal index $e < f(|z_s|)$ is chosen such that $e \notin Sat_{s-1}$ and $f_e(\alpha_{s-1})\downarrow$. If such an index exists and $f_e(\alpha_{s-1}) = i$, let $\alpha_s = \alpha_{s-1}i$ and $Sat_s = Sat_{s-1} \cup \{e\}$. Otherwise, let $\alpha_s = \alpha_{s-1}B(z_s)$ and $Sat_s = Sat_{s-1}$.

An easy induction shows that each requirement R_e is met, either by A meeting f_e or by f_e not being dense along the resulting set. Hence A is $t(n)$-generic. Since $A(z_s)$ differs from $B(z_s)$ only if some requirement R_e is chosen to be met at stage s, and since there are at most $f(n)$ many such stages for which $|z_s| = n$, $\|(A \triangle B) \cap \Sigma^{=n}\| \leq f(n)$. Finally, given $A \upharpoonright z_s$ and Sat_{s-1}, $A(z_s)$ can be computed in time

$$f(|z_s|) \cdot f(|z_s|) \cdot t(|A \upharpoonright z_s|) \log t(|A \upharpoonright z_s|) + t'(|z_s|)$$

by assumption about F. W.l.o.g. we may assume that $f(n) \leq n$, whence the claimed time bound for A follows from (2.7). □

Corollary 2.2.8. *There is a sparse n^k-generic set in* DTIME$(2^{(k+2)n})$ *and a sparse p-generic set in* DTIME(2^{n^2})

Proof. In Theorem 2.2.7 let $B = \emptyset$, $f(n) = t'(n) = n$, and $t(n) = n^k$, $t(n) = n^{\log\log n}$ resp. □

In the sequel, central interest will be on (general) p-generic sets. The following summary of properties of (general) p-generic sets illustrates the diagonalization power of these sets.

Proposition 2.2.9. *Let f be a $t(n)$-extension function, and A a (general) $t(n)$-generic set. Then A meets f not only once, but infinitely often.*

Proof. Suppose that A meets f only finitely often. Then there is a natural number m such that the function

$$f_m(x) = \begin{cases} f(x) & \text{if } f(x)\downarrow \text{ and } |x| > m \\ \uparrow & \text{otherwise} \end{cases}$$

is not met by A. Since f_m is a $t(n)$-extension function, this implies the desired contradiction. □

Proposition 2.2.10. *Every general n^2-generic set has exponential gaps.*

Proof. Consider the function

$$f(X \upharpoonright x) = \begin{cases} (0^n, 0), (0^n + 1, 0), \ldots (1^n, 0) & \text{if } x = 0^n \\ \downarrow \quad \text{otherwise.} \end{cases}$$

Then f is an n^2-extension function and by Proposition 2.2.9 A meets f at infinitely many x. So $A \cap \Sigma^{=n} = \emptyset$ for infinitely many n. □

For p-generic sets the following Lemma will be extremely useful, since it relaxes the limitation on simple extension functions.

Lemma 2.2.11. *([AmbB97]) If A is p-generic and f is a bounded or a waiting bounded n^k-extension function for some $k \geq 1$, then A meets f.*

Proof. We will only give the proof for waiting bounded n^k-extension functions. So assume that there is a constant $l \geq 1$ such that (2.6) holds whenever $f(\alpha)$ is defined. W.l.o.g. we may assume that $l_\alpha = l$ for all strings α on which f is defined. We split f into l simple n^{k+1}-extension functions $f_1, \ldots, f_l$ as follows: Given k with $1 \leq k \leq l$ and a string $X \restriction y$ let $f_k(X \restriction y) = i_{\alpha,k}$, where α is the shortest initial segment $X \restriction 0^n$ of $X \restriction y$ such that, for $f(\alpha) = (y_{\alpha,1}, i_{\alpha,1}), \ldots, (y_{\alpha,l}, i_{\alpha,l})$, $y = y_{\alpha,k}$ and $(X \restriction y)(y_{\alpha,j}) = i_{\alpha,j}$ for $1 \leq j < k$. If no such α exists, $f_k(X \restriction y)$ is undefined. Then, as one can easily check, f_1 is dense along all sets, f_{k+1} is dense along all sets which meet f_k infinitely often $(1 \leq k < l)$, and a set which meets f_l meets the extension function f, too. Since A meets any simple n^{k+1}-extension function which is dense along G not just once but infinitely often by Proposition 2.2.9 the above implies that any p-generic set G will meet f. □

Next we summarize a number of known properties of (general) p-generic sets that are related to a complexity class **C**. Some of the proofs are included, and may serve as an introduction to the typical lines of arguments used in later chapters.

Theorem 2.2.12. *([Amb96, AmbNT96]) If A is an n^k-generic set then A is* $\mathrm{DTIME}(2^{k \cdot n})$*-bi-immune.*

Proof. Suppose an n^k-generic set A that is not $\mathrm{DTIME}(2^{k \cdot n})$-bi-immune. Then there is an infinite language L in $\mathrm{DTIME}(2^{k \cdot n})$ such that either $L \subseteq A$ or $L \subseteq \overline{A}$. Define the simple extension functions f and g by

$$f(X \restriction n) = \begin{cases} 0 & \text{if } L(n) = 1 \\ \uparrow & \text{otherwise} \end{cases}$$

$$g(X \restriction n) = \begin{cases} 1 & \text{if } L(n) = 1 \\ \uparrow & \text{otherwise} \end{cases}$$

It is easy to see that the assumptions about L imply that these functions are dense n^k-extension functions, whence A meets both of them. So suppose that A meets f at n and g at m. This implies that n is a witness for $L \not\subseteq A$, and m is a witness for $L \not\subseteq \overline{A}$, a contradiction. □

Corollary 2.2.13. *There is no n^k-generic set in* $\mathrm{DTIME}(2^{k \cdot n})$ *and there is no p-generic set in* E.

Proof. The former follows since no **C**-bi-immune set is a member of **C**, while this implies the latter by definition of p-generic sets and E. □

A similar argument can be established for $\mathrm{DTIME}(2^{k \cdot n})$-incompressibility, p-selfreducibility and p-autoreducibility.

Theorem 2.2.14. *If A is an n^k-generic set then A is* $\mathrm{DTIME}(2^{(k-1)\cdot n})$-*incompressible.*

Theorem 2.2.15. *If A is p-selfreducible, then A is not p-generic.*

Proof. Assume that A is p-selfreducible, say via a polynomial-time bounded oracle Turing machine M, and the running time of M is bounded by n^k. Let $f : \Sigma^* \to \Sigma$ be the simple n^k-extension function defined by

$$f(X \restriction n) = 1 - M^{X \restriction n}(n).$$

Since f can simulate the query steps of M on input n by scanning the input $X \restriction n$, f is a simple $n^{\max\{2,k\}}$-extension function, and f is dense along all sets. Hence A meets f at some n. But this implies that

$$A(n) = f(A \restriction n) = 1 - M^{A \restriction n}(n) = 1 - M^A(n),$$

a contradiction to the assumption that A is p-selfreducible via M. □

Theorem 2.2.16. *If A is p-autoreducible, then A is not general p-generic.*

Proof. The ideas for this proof are analog to the ideas for Theorem 2.2.15. Assume that A is general p-generic and p-autoreducible via M. Recall that, given as oracle, the string $X \restriction n$ is interpreted as the set $(X \restriction n)0^\infty$. So given a set X, Let $i_n^X = M^{X \restriction n}(n)$ and $COND_n^X = \{(q,0) : q \in Q(x, M, X \restriction n) \,\&\, q > n\} \cup \{(n, 1 - i_n^X)\}$. Let f be the function that, given $X \restriction n$ as input, outputs the members of $COND_n^X$ in the appropriate order. Then f is computable in polynomial time. Hence A meets f at some n. This implies that $A(n) = 1 - i_n^{A \restriction n}$ and $M^A(n) = M^{A \restriction n}(n)$, whence

$$M^A(n) = A(n) = 1 - i_n^{A \restriction n} = 1 - M^{A \restriction n}(n) = 1 - M^A(n).$$

This is contradictory, whence no set can be both general p-generic and p-autoreducible. □

Finally, the following observation provides one of the basic tools in applications of generic sets to considerations about NP-r-completeness notions.

Theorem 2.2.17. *If A is a (general) p-generic set, f_1 and f_2 are independent p-extracting functions, then A_{f_1} and A_{f_2} are incompatible with respect to P-btt- (P-tt-)reductions.*

Proof. Here we will only treat one of two assertions, since the proof of the other is similar. So suppose that A is a general p-generic set, and $A_{f_1} \leq_{tt}^{\mathrm{P}} A_{f_2}$ via the P-tt-reduction M. Let

$$LONG(M,n) = \{f_2(q) : q \in Q(M, 0^n) \,\&\, |f_2(q)| \geq n\}$$

be the set of strings that determine those oracle answers in the computation of $M^{A_{f_2}}(0^n)$ which are not determined by $A \restriction 0^n$. The extension function that is designed to refute the assumed reduction between A_{f_1} and A_{f_2} is defined as follows.

For any set X let X_n be the f_2-extraction of $X^{<n}$, and $i_n^X = M^{X_n}(0^n)$. Finally, let

$$\begin{aligned} COND_n^X &= LONG(M,n) \times \{0\} \\ &\cup \{(f_1(0^n), 1 - i_n^X)\} \end{aligned}$$

Again, given $X \restriction 0^n$ as input, the extension function f corresponding to $COND_n^X$ is polynomial time bounded. (Note that f_1 and f_2 are disjoint, so f is well defined.) Hence A meets f at some n. This implies that

$$\begin{aligned} A_{f_2}(q) &= (A \restriction 0^n)_{f_2}(q) = A_n(q) \text{ for all } q \in Q(M, 0^n), \text{ whence} \\ M^{A_{f_2}}(0^n) &= M^{A_n}(0^n) = i_n^A, \text{ whereas} \\ A_{f_1}(0^n) &= 1 - i_n^A, \text{and} \\ A_{f_1}(0^n) &= M^{A_{f_2}}(0^n), \end{aligned}$$

a contradiction. Therefore there is no P-tt-reduction from A_{f_1} to A_{f_2}. □

We close this section with the proof of an existence theorem for general p-generic sets that was shown in [Amb96]. In contrast to Theorem 2.2.7, it lacks the bound on disagreement with the given set B.

Theorem 2.2.18. *([Amb96]) Let $t(n)$ be a nondecreasing, time-constructible function on $\mathbb{N}$ such that $t(n) \geq n$. Then there is a general $t(n)$-generic set A such that*

$$A \in \mathrm{DTIME}(2^{2n+1} n^2 t(2^{n+1}) \log t(2^{n+1})).$$

Sketch of proof. Fix an enumeration $\{f_e : e \geq 0\}$ of $t(n)$-extension functions such that, by means of a standard universal transducer, the function

$$F(0^e 1x) = \begin{cases} 1 f_e(x) & \text{if } f_e(x)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

can be computed in time $O(e \cdot t(2^{n+1}) \log t(2^{n+1}) + e)$. Note that A will be general $t(n)$-generic if the following requirements are met:

$$R_e : f_e \text{ dense along } A \Rightarrow A \text{ meets } f_e \tag{2.9}$$

This will be ensured by constructing the set A in stages. At each stage s, membership of the string z_s in A is decided, whence at the end of stage s $\alpha_s = A \restriction z_{s+1}$ is fixed. At the same time, a set Sat_s is fixed which records the requirements met at the end of stage s. Note that a requirement R_e should not enter the set Sat_s at a stage where $f_e(\alpha_{s-1})$ is defined, like in the proof of Theorem 2.2.7, but at a stage where any set extending α_s meets f_e. Therefore the bitwise construction of A may cause that the process of meeting requirement R_e (by taking the appropriate actions at several stages) is injured by taking action for a requirement of higher priority. For $i \leq 1$ and an extension function f, call a string $X \upharpoonright y$ *i-critical for f on x*, if $f(X \upharpoonright x)$ is defined, say $f(X \upharpoonright x) = (y_0, i_0), \ldots, (y_m, i_m)$, there is an index $0 \leq j \leq m$ such that $(y_j, i_j) = (y, i)$, and for all $0 \leq k < j$ it holds that $X(y_k) = i_k$.

Then the construction is as follows. Initially, let $\alpha_{-1} = Sat_{-1} = \emptyset$.

Stage s: Requirement R_e *requires attention* if $e < |z_s|$, $e \notin Sat_{s-1}$, and there is a $t \leq s$ and an $i \leq 1$ such that α_{s-1} is i-critical for f_e on z_t. If no requirement requires attention, let $\alpha_s = \alpha_{s-1}0$ and $Sat_s = Sat_{s-1}$. Otherwise, choose e minimal such that R_e requires attention. Choose the minimal $t \leq s$ and $i \leq 1$ (in this order) such that α_{s-1} is i-critical for f_e on z_t. Let $\alpha_s = \alpha_{s-1}i$. If $f(\alpha_{s-1} \upharpoonright z_t) = (y_0, i_0), \ldots, (y_m, i_m)$ and $z_s = y_m$ let $Sat_s = Sat_{s-1} \cup \{e\}$, otherwise, let $Sat_s = Sat_{s-1}$.
end of stage s

A straightforward induction shows that every requirement requires attention at most finitely often and is met. For details on this argument see [Amb96] or the proof of Theorem 4.1.1 in Chapter 4. So it remains to show that $A \in \mathrm{DTIME}(2^{n+1}t'(n))$ for $t'(n) = n^2 t(2^{n+1}) \log t(2^{n+1})$. As in the proof of Theorem 2.2.7 we have to show that, given α_{s-1} and Sat_{s-1} the actions at stage s can be done in time $t'(|z_s|)$. Let $|z_s| = n$. In the worst case, for each $e < n$ and each $t \leq s$, $F(0^e 1 \upharpoonright t)$ has to be computed and the output compared with α_{s-1}. This takes time $O(n \cdot n \cdot t(2^{n+1}) \log t(2^{n+1}) + 2^{n+1}))$. □

Corollary 2.2.19. *For $c \geq 1$ it holds that*

(i) there is a general n^c-generic set in $\mathrm{DTIME}(2^{c(n+2)})$

(ii) there is a general p-generic set in $\mathrm{DTIME}(2^{n^2})$

(iii) there is a general $2^{(\log^c n)}$-generic set in $\mathrm{DTIME}(2^{n^c})$

Proof. Apply Theorem 2.2.18 to the functions $t(n) = n^c$, $t(n) = n^{\log n}$, and $t(n) = n^{(\log n)^{c-1}}$. □

2.3 Resource-Bounded Randomness

Based on the martingale characterization of measure zero classes of Ville, Schnorr ([Sch71a, Sch71b]) proposed that a set X should be called *recursively random* if no recursive martingale succeeds on X. By additionally emposing resource bounds on the martingales, Schnorr introduced resource-bounded randomness concepts. Later Lutz ([Lut92]) extended these concepts by considering notions of resource bounded measure. Here we focus on these concepts for deterministic time complexity classes. We will admit only rational valued martingales, hence time bounds for martingales can be defined in the usual way.

Definition 2.3.1. Let $t(n)$ be a non-decreasing, recursive function on $\mathbb{N}$. A *$t(n)$-martingale* is a martingale such that $d \in \mathrm{DTIME}(t(n))$, i.e.

$$2 \cdot d(w) \geq d(w0) + d(w1). \tag{2.10}$$

These martingales form the basic notion both for resoure-bounded measure and randomness. Here we follow the definition of $t(n)$-martingales of Schnorr [Sch71b] and Ambos-Spies, Terwijn and Zheng [AmbTZ97]. Compared to the different setting in [AmbMW96], where a $t(n)$-martingale is induced by "betting ratios" computable in time $O(t(n))$, or the notion of Lutz that is based on $\mathrm{DTIME}(t(n))$ approximations of real valued martingales, these definitions result in different notions of $t(n)$-randomness, but yield equivalent definitions of p-randomness.

Definition 2.3.2. Let $t(n)$ be a nondecreasing recursive function. A class $\mathbf{A}$ has $t(n)$-measure zero ($\mu_t(\mathbf{A}) = 0$) if there is a $t(n)$-martingale d that succeeds on every $A \in \mathbf{A}$. A class $\mathbf{A}$ has $t(n)$-measure one ($\mu_t(\mathbf{A}) = 1$) if $\overline{\mathbf{A}}$ has $t(n)$-measure zero. The set A is called *$t(n)$-random* if no $t(n)$-martingale d succeeds on A.

Definition 2.3.3. A class $\mathbf{A}$ has p-measure zero ($\mu_p(\mathbf{A}) = 0$) if there is an n^k-martingale d that succeeds on every $A \in \mathbf{A}$ for some $k \geq 1$. A class $\mathbf{A}$ has p-measure one ($\mu_p(\mathbf{A}) = 1$) if $\overline{\mathbf{A}}$ has p-measure zero. The set A is called *p-random* if A is n^k-random for all $k \geq 1$.

Before we focus on the notion of p-randomness, we give some basic properties of $t(n)$-random sets.

Lemma 2.3.4. *Let $t(n)$, $t'(n)$ be nondecreasing recursive functions such that $t(n) \leq t'(n)$ for almost all n. Then it holds that*

(i) any $t'(n)$-random set is $t(n)$-random, and

(ii) if A is $t(n)$-random, then $\overline{A}$ is $t(n)$-random, too.

Proof. (i) is immediate by definition. For a proof of (ii) suppose a $t(n)$-martingale d that succeeds on $\overline{A}$. Then the function $\overline{d}$ defined by $\overline{d}(w) = d(\overline{w})$ succeeds on A, where $\overline{w}(m) = 1 - w(m)$ for $0 \leq m < |w|$. □

Lemma 2.3.5. *Let A be a set. Then the following statements are equivalent.*

(i) A is p-random.

(ii) $\{A\}$ does not have p-measure zero, that is, $\mu_p(\{A\}) \neq 0$

(iii) If $\mathbf{C}$ *has p-measure one, then* $A \in \mathbf{C}$.

Allthough μ_p is consistent with the Lebesque measure μ, this lemma shows that measurable classes w.r.t. μ need not be measurable with respect to μ_p, e.g., the singletons of p-random sets. Another important difference is closure under countable unions. However, if $\mathrm{DTIME}(t'(n))$ contains a universal function for $\mathrm{DTIME}(t(n))$, the countable union of $t(n)$-measure zero classes has $t'(n)$-measure zero. This observation also yields a general existence theorem for $t(n)$-random sets.

Definition 2.3.6. ([AmbTZ97]) Let $t(n)$ be a time-constructible, nondecreasing function. Fix an effective enumeration of $\mathrm{DTIME}(t(n))$. The function $d : \Sigma^* \to \mathbb{Q}^+$ is called *universal* for $t(n)$-martingales (corresponding to the enumeration $\{f_i : i \geq 1\}$) if d is defined by

$$d(w) = \sum_{i=0}^{2|w|} 2^{-i} D(i,w) + 4 \cdot 2^{-|w|} \tag{2.11}$$

and $D(i,w) = f_i(w)$ if f satisfies (2.10) for all x such that $x1$ is length-lexicographic less than w; and $D(i,w) = 1$ otherwise.

Lemma 2.3.7. *Let D and d as above. Then*

(i) d is a martingale

(ii) $S^\infty[d] \supseteq \bigcup_{d' \in \mathrm{DTIME}(t(n))} S^\infty[d']$

Proof. Note that the function d_i defined by $d_i(w) = D(i,w)$ is a martingale. W.l.o.g. assume that these martingales are normed. Thus $D(i,w) \leq 2^{|w|}$ for all i and

$$\begin{aligned}
d(w0) + d(w1) = & \\
& \sum_{i=0}^{2|w|+2} 2^{-i} D(i,w0) + \sum_{i=0}^{2|w|+2} 2^{-i} D(i,w1) + 2 \cdot 4 \cdot 2^{-|w|-1} \\
\leq\ & 2 \cdot \sum_{i=0}^{2|w|+2} 2^{-i} D(i,w) + 2 \cdot 4 \cdot 2^{2-|w|}(1 - 1/2) \\
=\ & 2 \cdot d(w) + 2^{-2|w|-1} D(2|w|+1, w) + \\
& \quad 2^{-2|w|-2} D(2|w|+2, w) - 4 \cdot 2^{-|w|} \\
\leq\ & 2 \cdot d(w) + 2^{-|w|}(1/2 + 1/4 - 4) \leq 2 \cdot d(w)
\end{aligned}$$

and d is a supermartingale, too. For a proof of (ii), note that for $d' \in \mathrm{DTIME}(t(n))$ and i such that $d'(w) = D(i,w)$

$$\limsup_{n\to\infty} d(X \restriction n) \geq 2^{-i} \cdot \limsup_{n\to\infty} d'(X \restriction n).$$

So if d' succeeds on X then d succeeds on X as well. □

Thus a bound on the running time of the martingale d yields the following.

Corollary 2.3.8. *There is a $t(n)$-random set in* $\mathrm{DTIME}(2^{n+1} \cdot t'(2^{n+1}))$ *for $t'(n) \geq n^3 t(n)(\log t(n))^4$ ([AmbTZ97]).*

Corollary 2.3.9. *For $k \geq 1$ there is an n^k-random set in* E *and an $n^{(\log n)^k}$-random set in* EXP*, whence a p-random set in* EXP.

We close this section by a short comparison of p-randomness with (general) p-genericity.

Theorem 2.3.10. *([AmbNT96, AmbTZ97]) Every $n \cdot t(n)$-random set is $t(n)$-generic.*

Proof. Assume an $n \cdot t(n)$-random set A and a simple $t(n)$-extension function that is dense along A. Define a martingale d by $d(\lambda) = 1$ and for all $w \in \Sigma^*$, $i \in \Sigma$ let

$$d(wi) = \begin{cases} 2 \cdot d(w) & \text{if } f(w)\downarrow = 1 - i \\ 0 & \text{if } f(w)\downarrow = i \\ d(w) & \text{otherwise} \end{cases}$$

Then A has to meet f, since otherwise d will be nondecreasing along A and will be doubled infinitely often, hence will succeed on A. Since d is an $n \cdot t(n)$-martingale, this yields a contradiction. □

Corollary 2.3.11. *Every p-random set is p-generic .*

This correspondence between randomness and genericity shows that the genericity concept that was obtained by restrictions on simple extension functions is weak enough to be compatible with resource bounded measure. Therefore we obtain the results in the last section for p-generic sets also for p-random sets.

Corollary 2.3.12. *Let A be a p-random set in* EXP. *Then A is* P*-bi-immune,* P*-incompressible, not p-selfreducible and not P-(btt)-complete for* EXP.

But, of course, there are many characteristics in which random sets differ from generic sets. These are consequences from the differences between unsuccessful martingales and infinitely often met extension functions. While for a generic set the ratio of strings at which an extension function is met to the "chances" to do so is unpredictible, a unsuccessful martingale has to "lose" in the majority of chances. Intuitively, a random set shows a more "regular" irregularity than generic sets, and this regularity can be expressed by statistical laws. (In [Wan96] this is discussed in detail.) Here we point out only a few examples of these characteristics of $t(n)$-random sets that are of importance in this thesis.

Theorem 2.3.13. *Let A be n^2-random and P a set in* P. *Then $A^{=n} \neq P^{=n}$ for almost all $n \geq 1$.*

Proof. Consider the following normed martingale.

$$d((X \restriction x)i) = \begin{cases} d(X \restriction x) & \text{if} \quad \exists y < x \text{ such that} \\ & \quad\ |y| = |x| \,\&\, P(y) \neq X(y) \\ \frac{3}{2}d(X \restriction x) & \text{if} \quad P(x) = i \text{ and } X(y) = P(y) \\ & \quad\ \text{for all } y < x \text{ of length } |x| \\ \frac{1}{2}d(X \restriction x) & \text{if} \quad P(x) = 1 - i \text{ and } X(y) = P(y) \\ & \quad\ \text{for all } y < x \text{ of length } |x| \end{cases}$$

Since P is computable in polynomial time, d is an n^2-martingale. Now consider the sequence $(a_n)_{n\in\mathbb{N}}$ defined by $a_n = d(A \restriction 0^{n+1})$. If $A^{=n} \neq P^{=n}$, then $a_n \geq 1/2 \cdot a_{n-1}$. If $A^{=n} = P^{=n}$, then $a_n = (\frac{3}{2})^{2^n} a_{n-1}$. Taken together and using $2^n \geq 2n$, this implies that

$$a_n \geq \left(\frac{1}{2}\right)^n \cdot \left(\frac{3}{2}\right)^{2^n} \geq \left(\frac{1}{2}\right)^n \cdot \left(\frac{3}{2}\right)^{2n} \left(\frac{9}{8}\right)^n$$

for each n such that $A^{=n} = P^{=n}$. Since otherwise d would succeed on A, there are only finitely many such n. □

In contrast to Theorem 2.3.10 and Corollary 2.3.11 this establishes the following incompatability.

Corollary 2.3.14. *([Amb96]) No set can be both p-random and general p-generic .*

Proof. Since general p-generic sets have exponential gaps, this is an immediate consequence of the previous theorem with the empty set in place of P. □

The difference between p-random and p-generic sets becomes even more obvious when the density of a set is considered.

Lemma 2.3.15. *Every p-random set A is dense. I.e., there is an $\varepsilon > 0$ such that $|A^{\leq n}| > 2^{n^\varepsilon}$ for almost all n.*

Proof. Assume a set A that is not dense. Then for infinitely many n it holds that $|A^{\leq n}| \leq 2^{\sqrt{n}}$. Define the n-martingale d by $d(\lambda) = 1$, $d(w0) = 3/2 \cdot d(w)$, and $d(w1) = 1/2 \cdot d(w)$. Since $|(\Sigma^*)^{\leq n}| = 2^{n+1} - 1$, it follows that

$$\limsup_{n\to\infty} d(A \restriction z_n) \geq \lim_{n\to\infty} \left((1/2)^{2^{\sqrt{n}}} \cdot (3/2)^{2^{n+1}-1-2^{\sqrt{n}}}\right) = \infty.$$

Hence d succeeds on A, so A is not p-random. □

Finally, we want to establish the analog to Theorem 2.2.4.

Lemma 2.3.16. *Let* $\mathbf{C}$ *be a complexity class that is closed under P-m-reducibility. Then the following are equivalent:*

(i) There is an n^2*-random set in* $\mathbf{C}$.

(ii) There is an n^k*-random set in* $\mathbf{C}$ *for every* $k \geq 1$.

(iii) There is a p*-random set in* $\mathbf{C}$.

(iv) There is an $2^{\log^k n}$*-random set in* $\mathbf{C}$ *for every* $k \geq 1$.

The proof follows the same argument as in the case of Theorem 2.2.4.

Chapter 3

Non-Smallness Hypotheses about NP

3.1 Introduction

By the time hierarchy Theorem the levels of the exponential time hierarchy are proper, whence $P \subset E \subset EXP$. The relation between determinism and non-determinism implies that NP is localized somewhere in between, i.e.,

$$P \subseteq NP \subseteq EXP,$$

and there are relativizations realizing the two extremes (see [BakGS75]). Both extremes are widely disbelieved because of their contra-intuitive consequences (see Chapter 1). The internal structure of P as well as of EXP is well understood and the two classes differ substantially in structure (see below). Hence the two extreme possibilities to localize NP are contrary not only with respect to the size of NP, but also with respect to the internal structure of NP.

On the other hand, considering structural properties about NP will have implications on the localization of NP. For instance, the statement that NP contains a set witnessing that P-T- and P-tt-reductions differ (resp. coincide), implies that $NP \neq P$ ($NP \neq EXP$ resp.). Hence a study of the internal structure of NP depends on a previously chosen hypothesis about NP.

That NP contains a non-negligible part of EXP and therefore exceeds P substantially, often can be expressed by stating that NP contains a set with some appropriate structural property ψ not shared by the sets in P. Since NP is contained in EXP, it suffices to consider properties shared by some set in EXP. Obviously, the weakest (that is, necessary) such statement would be that NP contains a set not in P, while the strongest (admissible) such statement would be that NP contains an EXP-complete set (which implies that NP equals EXP).

To obtain a meaningful non-smallness hypothesis about NP we consider the relativized version of such a statement and require that the conditions mentioned above relativize.

Definition 3.1.1. A *non-smallness hypothesis about* NP is a sentence about sets φ of the form $\varphi = \exists D\, \psi(D)$ in which complexity classes like P, NP, E, EXP, FP, etc. may appear as constants. It is called *possible*, if it satisfies the conditions

(i) consistency:

there is a set A such that $\mathrm{NP}^A \models \varphi^A$,

(ii) nontriviality:

for all sets A, $\mathrm{P}^A \not\models \varphi^A$, and

(iii) admissability:

for all sets A, $\mathrm{EXP}^A \models \varphi^A$.

And it is called *necessary*, if $\mathrm{NP}^A \models \varphi^A$ for all sets A for which $\mathrm{P}^A \neq \mathrm{NP}^A$.

Here φ^A denotes the relativization of φ to A, the sentence that is obtained by replacing the Turing machines involved in the statement φ by oracle Turing machines equipped with oracle A.

Conditions (ii) and (iii) ensure that the property ψ will be incompatible with membership in P but will occur within EXP, regardless of the chosen relativization. Condition (i) states that for sets in NP the property ψ is conceivable. Since there are oracles for which NP = EXP, this condition is already implied by condition (iii).

Possible non-smallness hypotheses about NP are compared in strength according to an ordering that is also based on relativizations. Consider two hypotheses ψ and φ stating the existence of sets with property ψ' and φ' resp. If the existence of a set in NP with property ψ' implies the existence of a set in NP with property φ' and this fact relativizes, we call ψ stronger than φ.

Definition 3.1.2. Let φ, ψ be possible non-smallness hypotheses about NP. ψ is *stronger than* φ ($\varphi \preceq \psi$) if for all sets A,

$$\mathrm{NP}^A \models \psi^A \quad \Rightarrow \quad \mathrm{NP}^A \models \varphi^A.$$

ψ is *strictly stronger than* φ ($\varphi \prec \psi$) if $\varphi \preceq \psi$ and $\psi \not\preceq \varphi$.

Proposition 3.1.3. *The ordering $\preceq$ on the possible non-smallness hypotheses about* NP *has a minimal and a maximal element.*

Proof. Consider the sentences

$$\begin{aligned}\varphi_{min} &= \exists D(D \notin \mathrm{P}) \\ \varphi_{max} &= \exists D(D\ P\text{-}m\text{-complete for EXP}).\end{aligned}$$

Both are possible non-smallness hypotheses about NP. Let ψ be a possible non-smallness hypothesis about NP and let A be any set. If $\mathrm{NP}^A \models \psi^A$ then the second condition in Definition 3.1.1 implies that $\mathrm{NP}^A \neq \mathrm{P}^A$, whence $NP^A \models \varphi_{min}^A$. On the other hand, if $NP^A \models \varphi_{max}$, then $\mathrm{NP}^A = \mathrm{EXP}^A$. Hence the third condition implies that $\mathrm{NP}^A \models \psi^A$. So

$$\varphi_{min} \preceq \psi \preceq \varphi_{max}.$$

□

3.2 Weak and strong hypotheses about NP

Since φ_{min} is the most plausible hypothesis about NP, structural properties obtained under it are of special interest. But only very little is known about the internal structure of NP under this minimal hypothesis (see below for details). We distinguish two kinds of possible non-smallness hypotheses, weak hypotheses and strong hypotheses, without giving a strict line of difference. Roughly speaking, a hypothesis is called *strong* if, in some sense, it states that NP contains a non-negligible subset of E, and that NP is not a subset of E. And it is called *weak* otherwise. We start with discussing some weak hypotheses about NP.

Nontriviality Hypothesis:

(N) $\exists D \in \mathrm{NP}\,(D \notin \mathrm{P})$

Sparseness Hypothesis:

(S) $\exists D \in \mathrm{NP}\,(D \notin \mathrm{P}\ \&\ D$ is sparse$)$

Tally Set Hypothesis:

(T) $\exists D \in \mathrm{NP}(D \notin \mathrm{P}\ \&\ D$ is tally$)$

Exptally Set Hypothesis:

(TT) $\exists D \in \mathrm{NP}(D \notin \mathrm{P}\ \&\ D \subseteq \{0^{2^n} : n \geq 0\}$

Bi-immunity Hypothesis:

(BI) $\exists D \in \mathrm{NP}(D\ p\text{-bi-immune})$

Incompressibility Hypothesis:

(INC) $\exists D \in \mathrm{NP}(D\ p\text{-incompressible})$

Proposition 3.2.1. $(N) \preceq (S) \preceq (T) \preceq (TT) \preceq (BI) \preceq (INC)$

Proof. The only nontrivial relation is $(TT) \preceq (BI)$. Note that, if D is a p-bi-immune set in NP, then $D' = \{0^{2^n} : n \geq 0 \,\&\, 0^{2^n} \in D\}$ is an exptally set in NP $\setminus$ P. Since this fact relativizes, $(TT) \preceq (BI)$. □

Moreover, all these relations are strict in the sense given above. Next we summarize known consequences of these hypotheses.

Theorem 3.2.2. *1. Assume (N). Then there is a set in* NP $\setminus$ P *which is not* NP*-P-m-complete ([Lad75]). In fact, there are infinitely many P-m-degrees in* NP*.*

The NP*-m-complete sets are neither sparse ([Mah82]) nor p-selective ([Sel79]).*

2. Assume (T). Then E $\neq$ NE*, there is a set in* NP $\setminus$ P *which is p-selective, and there is a set in* NP $\setminus$ P *which is not P-d-selfreducible ([Sel79]).*

3. Assume (TT). Then EE $\neq$ NEE *([Sel79]), and there is a search problem in* NP *that is not reducible to the corresponding decision problem ([BelG94]).*

In contrast to the rich and diverse structure inside EXP, it should be pointed out that nothing is said here about polynomial time reduciblities with unbounded number of queries to the oracle, or about the completeness notions induced by polynomial time reduciblities. Such kinds of consequences seem to require stronger hypotheses as the following.

Genericity Hypothesis:

(G) $\quad \exists D \in \text{NP}\ (D\ p\text{-generic})$

Randomness Hypothesis:

(R) $\quad \exists D \in \text{NP}\ (D\ p\text{-random})$

General Genericity Hypothesis:

(GG) $\quad \exists D \in \text{NP}\ (D \text{ general } p\text{-generic})$

Completeness Hypothesis :

(C) $\quad \exists D \in \text{NP}\ (D\ p\text{-}m\text{-complete for EXP})$

Since the corresponding results in Sections 2.2 and 2.3 relativize,

$$(G) \preceq (GG) \qquad \text{and} \qquad (G) \preceq (R).$$

By Lemma 2.3.16, (R) is equivalent to the non-smallness hypothesis introduced by Lutz in [Lut92], stating that NP does not have measure zero in E ($\mu(\text{NP}|\text{E}) \neq 0$). There are several interesting consequences of this hypothesis, among the most interesting are the following.

Proposition 3.2.3. *([LutZ97]) Assume (R). Then*

(1) there is a set in NP *that is P-btt(3)-complete, but not P-m-complete for* NP *([LutM96]), in fact, not* NP*-btt*(2)*-complete ([May94b]).*

(2) For $\epsilon > 0$*, no sparse set can be P-tt(*$n^{1-\epsilon}$*) -complete or P-T(*$n^{0.5-\epsilon}$*)-complete for* NP.

This thesis concentrates on the hypotheses (G) and (GG), since for p-generic sets their relation to diagonalizations aganinst polynomial time reductions is more immediate than for p-random sets. Moreover, it turns out that results in the flavour of 3.2.3 (1) in many cases also are obtained under (G), while proofs under this hypothesis can be given in a more direct, lucid way. On the other hand, results in the flavour of 3.2.3 (2) are related to the Borel-Cantelli-Lemma, which enables diagonalizations against polynomial time reductions for which the number of queries is not encreasing too fast with the input length. It is in not clear, how such results could be within the reach of category-based arguments.

Chapter 4

Relativizations Comparing Strong Hypotheses about NP

In this chapter the strong hypotheses about NP, (G), (R), (GG) and (C) are studied under the ordering $\prec$ introduced in Definition 3.1.2. As was pointed out in Chapter 2, the genericity concept used to define p-genericity is weak enough to become compatible with p-randomness. Hence it is obvious that $(G) \preceq (R) \preceq (C)$ and $(G) \preceq (GG) \preceq (C)$. In fact, it holds that $(G) \prec (R) \prec (C)$ and $(G) \prec (GG) \prec (C)$. To show this, it suffices to construct an oracle A such that, relative to A, NP contains a general p-generic set but no p-random set; and an oracle relative to which NP contains an n^2-random set but no general p-generic set. Both constructions are quite involved, so each is discussed in a separate section.

4.1 Genericity versus Randomness

The following theorem is a generalization of the Lemma stating $(G) \prec (R)$ given in [AmbB97], and the proof given below is based on ideas used there.

Theorem 4.1.1. *There is an oracle A such that, relative to A,* NP *contains a general p-generic set but no p-random set.*

Proof. We will construct a set A with the required properties in stages. I.e., we will effectively enumerate a sequence of finite characteristic functions $(\alpha_s)_{s\geq 0}$ which has the characteristic function of A as its limit.

In order to guarantee that NP^A does not contain any p^A-random set we will ensure that every set in NP^A agrees with some P^A-set on all strings of length n for infinitely many numbers n. This will be achieved by letting the oracle A look like the canonical NP^A-complete set K^A on sufficiently large intervals infinitely often. Let N_e be the e-th nondeterministic oracle Turing machine with respect to some standard numbering and let

$$K^A = \{\langle 0^e, x, 0^n\rangle \ : \ N_e^A \text{ accepts } x \text{ in } \leq n \text{ steps}\},$$

where, for technical convenience, we assume that $|\langle x, y, z\rangle|$ is odd for all strings x, y, z. Then the construction of A will ensure that

$$\exists^\infty n\, \forall x\, (n \leq |x| \leq 2^n \;\Rightarrow\; A(x) = K^A(x)) \tag{4.1}$$

That this suffices to eliminate random sets in NP^A is shown as follows.

Claim 1: Assume that (4.1) holds. Then NP^A does not contain any p^A-random set.

Proof. Given $B \in \mathrm{NP}^A$, there is an index e and a polynomial p such that $x \in B$ iff $\langle 0^e, x, 0^{p(|x|)}\rangle \in K^A$. Since $|\langle 0^e, x, 0^{p(|x|)}\rangle|$ is polynomial bounded in $|x|$, it follows from (4.1) that there are infinitely many numbers n such that, for the P^A-set $\hat{B} = \{x : \langle 0^e, x, 0^{p(|x|)}\rangle \in A\}$, $B^{=n} = \hat{B}^{=n}$. By the relativized version of Theorem 2.3.13, this implies that B is not p^A-random. □

In order to achieve the second goal of the construction we will ensure that the set

$$G^A = \{x : \exists y\, (|y| = |x|^2 \;\&\; xy \in A)\}$$

will be general p^A-generic. Note that the strings xy that decide about membership of x in G^A are of even length. Fix a recursive enumeration $(f_e)_{e\geq 0}$ of the oracle dependent p-extension functions such that w.l.o.g. f_e is an n^e-extension function. Then to make G^A general p-generic relative to A it suffices to meet the requirements

$$R_e : f_e^A \text{ dense along } G^A \;\Rightarrow\; G^A \text{ meets } f_e^A$$

for $e \geq 0$. These requirements will be met in the following way. If the hypothesis of R_e holds then at some stage s of the construction we will choose the finite extension α_s of the previously specified finite part α_{s-1} of A in such a way that α_s will force that f_e^A is defined on input $G^A\restriction s$, say

$$f_e^A(G^A\restriction s)\downarrow= (x_0, i_0), \ldots , (x_m, i_m). \tag{4.2}$$

Hence, to meet requirement R_e we have to ensure that $G^A(x_j) = i_j$ for all $0 \leq j \leq m$. This intention is documented by adding the pairs (e, x_j, i_j) $(0 \leq j \leq m)$ to the list of announcements *Announce*$_s$, used during the appropriate stages to choose extensions $\alpha_{s_o}, \ldots , \alpha_{s_m}$ such that α_{s_j} forces $G^A(x_j) = i_j$ for $0 \leq j \leq m$. This process will be aborted only if a requirement $R_{e'}$ of higher priority provides a contradicting announcement $(e', x_j, 1-i_j)$ for some $0 \leq j \leq m$. Hereby requirement $R_{e'}$ is of higher priority than requirement R_e, if $e' < e$. Since there are only finitely many requirements of higher priority, we will argue that this will happen only finitely often, whence at some stage s, α_s forces G^A to meet f_e^A. Moreover, by hypothesis there are infinitely many stages to start the above process. This will allow us to spread out the actions for meeting the requirements R_e in such a way that, by letting $A(x) = K^A(x)$ in the intermediate phases, condition (4.1) will be satisfied. In the construction below this will be implemented by allowing only requirements R_e to act at stage s for which $e < d(|s|)$ for some slowly growing function d.

If we take action for requirement R_e and choose α_s to force (4.2) and $\alpha_{s_o}, \dots, \alpha_{s_m}$ to force $G^A(x_j) = i_j$ for $j \leq m$, we have to make sure that this action will not simultaneously force $f^A_{e'}(G^A \restriction s')\downarrow = (y_0, j_0), \dots, (y_l, j_l)$ and $G^A(y_k) = 1 - j_k$ for some $k \leq l$, $e' < e$ and s'. Otherwise, for some requirement $R_{e'}$, the actions of the requirements R_e with $e > e'$ could force $f^A_{e'}$ to be dense along G^A in such a way that we will not be able to ensure that G^A meets $f^A_{e'}$. So the combined actions of the lower priority requirements could cause the failure of $R_{e'}$. This problem is overcome, firstly, by imposing strict bounds on the possible extensions of α_{s-1} to choose from, and, secondly, by choosing α_s carefully enough to maintain the reason for favouring R_e over requirements of higher priority to take action at stage s.

Before stating the formal construction we need the following notation.

For a string x call the set $code(x) = \{xy : |y| = |x|^2\}$ the *coding region of* x, let $m(x)$ be the least element of $code(x)$, and call a finite function α *x-honest* if, for all $y \geq x$ and all $z \in dom(\alpha) \cap code(y)$, $\alpha(z) = 0$. Then $x \in G^A$ iff $A \cap code(x) \neq \emptyset$, and, for any x-honest α with $code(x) \not\subseteq dom(\alpha)$ we can find extensions β_i forcing $G^X(x) = i$ for all extensions X of β_i by letting β_0 be the extension of α along $\{(z, 0) : z \in code(x)\}$ and by letting β_1 be the extension of α along $\{(z, 1)\}$ for some string $z \in code(x) - dom(\alpha)$. In the construction of A below, the part α_{s-1} of A specified by the end of stage $s-1$ will be chosen to extend $A \restriction m(s)$ whence $G^A \restriction s$ will be determined by the end of stage $s-1$, namely $G^A \restriction s = G^{\alpha_{s-1}} \restriction s$.

To describe the dependence of $f^X_e(G^X \restriction x)$ on the oracle X, let $\varphi(X, e, x)$ be the use function of this computation, i.e.,

$$\varphi(X, e, x) = \\ \{(z, X(z)) \ : \ z \text{ is queried in the computation of } f^X_e(G^X \restriction x).\}$$

Then, for any oracles X and Y such that $X \restriction m(x) = Y \restriction m(x)$ (whence $G^X \restriction x = G^Y \restriction x$) and $\varphi(X, e, x) = \varphi(Y, e, x)$, $f^X_e(G^X \restriction x) = f^Y_e(G^Y \restriction x)$. Moreover, since f_e is an n^e-extension function, $|dom(\varphi(X, e, x))| \leq |G^X \restriction x|^e$, which is less than $2^{e \cdot (|x|+1)}$ by (2.7). And, for any $z \in dom(\varphi(X, e, x))$, $|z| \leq 2^{e \cdot (|x|+1)}$. This implies that for any oracle X the set

$$Q^X_s = \{z : \exists x \leq s \, \exists e < d(|s|) \, (z \in dom(\varphi(X, e, x)))\}$$

has at most

$$q_s = 2^{|s|+1} \cdot d(|s|) \cdot 2^{d(|s|) \cdot (|s|+1)}$$

elements. Note that $q_s < 2^{|s|^2}$ for almost all numbers s if $d(|s|) \leq \log(|s|)$ whereas $|code(s)| = 2^{|s|^2}$. This will ensure that in the construction we can choose α_s to force the computations $f_e^A(G^A \upharpoonright x)$ to converge (for $e < d(|s|)$ and $x \leq s$) without exhausting $code(s)$ completely. We assume that the function d limiting the number of requirements which are considered at some stage is chosen so that $q_s < 2^{|s|^2}$ holds for all s and such that, for all $e < d(|s|)$, $2^{e\cdot(|s|+1)} < 2^{2^{|s|}}$.

Moreover, in order to guarantee (4.1), we choose d to be nondecreasing and to satisfy $d(n) \leq \log(n)$ and

$$\begin{array}{rcl} n_j &=& \mu l(d(l) > j) \implies n_{j+1} > g(n_j) \text{ for all } j \geq 0 \\ && \text{where } g(m) = 2^{2^{\cdot^{\cdot^{2^m}}}} \}8m(m+1)\text{-times.} \end{array} \tag{4.3}$$

For convenience, we define a function $t : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ inductively by

$$t(0,m) = m \qquad \text{and} \qquad t(n+1,m) = 2^{t(n,m)}, \tag{4.4}$$

whence $g(m) = t(8m^2 + 8m, m)$.

Stage s ($s \geq 0$) of the construction of A consists of three parts where the first two parts contain the actions for meeting the requirements R_e while in the third part condition (4.1) will be ensured. In stage s we do not only define the initial part α_s of A but also sets Sat_s and $Announce_s$ where Sat_s contains the indices of the requirements which have been satisfied by the end of stage s, while a tuple $(e, x', i) \in Announce_s$ indicates that there is a chance to meet requirement R_e if $\alpha_{x'}$ forces $G^A(x') = i$. The initial values of these parameters are $\alpha_{-1} = Sat_{-1} = Announce_{-1} = \emptyset$.

Stage s of the construction of A:

Step 1: Requirement R_e is *active* if $e < d(|s|)$, $e \notin Sat_{s-1}$, and for all tuples $(e', x, i) \in Announce_{s-1}$ it holds that $e' \neq e$. R_e *requires attention via* β if R_e is active, $f_e^\beta(G^\beta \upharpoonright s) \downarrow$ and β is an s-honest extension of α_{s-1}. Requirement R_e *requires attention* if R_e requires attention via some β.

If some requirement requires attention then fix the least e and β (in this order) such that R_e requires attention via β. Here the finite function β is considered less than β', if $|dom(\beta)| < |dom(\beta')|$, or if the domains are of the same cardinality and the position of β is prior to that of β' with respect to some effective numbering of finite characteristic functions. Let γ be the extension of β along the use function $\varphi(\beta, e, s)$. Having thus fixed

$$f_e^\gamma(G^\gamma \upharpoonright s) = f_e^\beta(G^\beta \upharpoonright s) = (x_0, i_0), \dots, (x_m, i_m),$$

let

$$Announce'_s = Announce_{s-1} \cup \{(e, x_j, i_j) : 0 \leq j \leq m\},$$

and say that R_e *receives attention* (*via* β). Otherwise, let $\gamma = \alpha_{s-1}$ and $Announce'_s = Announce_{s-1}$.

Step 2: In case that $x \neq s$ for all $(e,x,i) \in Announce'_s$, let $\delta = \gamma$, $Announce_s = Announce'_s$, $Sat_s = Sat_{s-1}$ and proceed to step 3. Otherwise, select the tuple $(e,s,i) \in Announce'_s$ with the minimal first component e. Let γ' be the extension of γ along the union of the use functions $\varphi(\gamma,e',x)$ for all numbers $e' \leq e$ and strings $x \leq s$, let z be the least string in $code(s) \setminus dom(\gamma')$, and let

$$\begin{aligned}
\delta &= (\gamma' \cup (z,i)) \sqcup (code(s) \times \{0\}) \\
Announce_s &= Announce'_s \setminus \{(e',x',i') : \\
&\qquad (e',s,1-i) \in Announce'_s\} \\
Sat_s &= Sat_{s-1} \cup \{e : (e,s,i) \in Announce_s \,\& \\
&\qquad \forall (e,x',i') \in Announce_s\, [x' \leq s]\}
\end{aligned}$$

Step 3: For the extension δ of α_{s-1} defined in Step 2 let

$$\alpha_s = \delta \sqcup (K^{\alpha_s} \restriction m(s+1)).$$

end of stage s

This completes the construction. Note that $K^{\alpha}(x)$ only depends on α for the strings in $dom(\alpha)$ which are less than x. So in Step 3, α_s and $K^{\alpha_s} \restriction m(s+1)$ can be inductively defined by fixing $\alpha_s(y)$ for the strings $y < m(s+1)$ with $y \notin dom(\delta)$ in order.

The correctness of the construction is established by the following claims. Let $Sat = \bigcup_{s \geq 0} Sat_s$.

Claim 2: For all $s \geq 0$, α_s is well defined, α_s extends α_{s-1}, $dom(\alpha_s)$ contains all strings less than $m(s+1)$, and, for $z \in dom(\alpha_s)$ with $z \not< m(s+1)$, $|z| < 2^{2^{|s|}}$. Moreover, for any $x > s$, $z \in dom(\alpha_s) \cap code(x)$ implies that $\alpha_s(z) = 0$ and $z \in dom(\varphi(\alpha_s, e, s'))$ for some e such that $e < d(|s|)$ and $s' \leq s$.

Proof: The proof, which is by induction on s, is based on the following observations. Obviously, $dom(\alpha_s)$ contains all strings less than $m(s+1)$ by step 3 of the construction. So consider γ, γ', and δ as constructed in the first two steps. If no requirement receives attention at stage s, then $\gamma = \alpha_{s-1}$. If R_e receives attention via β at stage s, then γ is an s-honest extension of α_{s-1}, and, by minimality of β, $\gamma \setminus \alpha_{s-1} \subseteq \varphi(\beta,e,s)$. Hence, if $\delta = \gamma$, then the claim follows immediately from the induction hypothesis and the equation $\varphi(\beta,e,s) = \varphi(\alpha_s,e,s)$. On the other hand, if $\delta \neq \gamma$, then an analog argument shows that γ' is an s-honest extension of α_{s-1} and $dom(\gamma') \cap code(s) \subseteq Q_s^{\gamma'}$. Hence the string $z \in code(s) \setminus dom(\gamma')$ required for the definition of δ exists. Moreover, for all $z \geq m(s+1)$, $z \in dom(\alpha_s) \setminus$

$dom(\alpha_{s-1})$ implies that $z \in Q_s^{\gamma'} = Q_s^{\alpha_s}$. Hence, the claim follows by the induction hypothesis and the above remark on the length of the strings in $dom(\varphi(X, e, s))$. □

Claim 3: If $e \in Sat_s$ then G^A meets f_e^A at some $x \le s$.

Proof: Assume that $e \in Sat_s$ and let s' be the stage where e entered into Sat, i.e. $e \in Sat_{s'} \setminus Sat_{s'-1}$. By definition of $Sat_{s'}$ this implies that $(e, s', i) \in Announce_{s'}$ for a unique $i \in \Sigma$. Choose the maximal $t \le s'$ such that R_e received attention at stage t via some β, $f_e^\beta(G^{\alpha_{t-1}} \restriction t) = (x_0, i_0), \dots, (x_m, i_m)$ and $x_m = s'$. Then $(e, x_j, i_j) \in Announce_t$ for all $0 \le j \le m$. Since R_e can receive attention only if R_e is active, for all stages t' between t and s', these are the only tuples in $Announce_{t'}$ with e as first component. Now assume that G^A does not meet f_e^A at t. Then there is a minimal j such that $G^{\alpha_{x_j}}(x_j) = 1 - i_j$. But this implies that $j < m$ and by minimality of j there was an announcement $(e, x_j, i_j) \in Announce'_{x_j}$. Hence another announcement $(e', x_j, 1 - i_j)$ of higher priority was fulfilled during the stage x_j. By maximality of t this implies that no tuple with e as first component can be a member of $Announce_{s'}$, contrary to the assumption about s'. This completes the proof of Claim 3, since $f_e^\beta(G^{\alpha_{t-1}} \restriction t) = f_e^A(G^A \restriction t)$ and $G^A(x_j) = G^{\alpha_{x_j}}(x_j) = i_j$ for all $0 \le j \le m$. □

Claim 4: Whenever a requirement R_e receives attention during stage s, there is a requirement $R_{e'}$ and a stage $s' \ge s$ such that $e' \le e$, $e' \in Sat_{s'} \setminus Sat_{s'-1}$, and $|s'| \le t(2(e+1), |s|)$.

Proof: Consider the case of requirement R_e receiving attention via β during stage s, i.e.,

$$f_e^A(G^A \restriction s) = f_e^\beta(G^{\alpha_{s-1}} \restriction s) = (x_0, i_0), \dots, (x_m, i_m). \tag{4.5}$$

Let $t_{e,s}$ denote the minimal x_j such that $G^A(x_j) \ne i_j$ if such an x_j exists and denote x_m otherwise. We want to show, by induction on e, that then there are $e' \le e$ and a stage $s' \ge t_{e,s}$ such that $|s'| \le t(2(e+1), |s|)$ and $e' \in Sat_{s'} \setminus Sat_{s'-1}$.

If R_0 receives attention at stage s, then $t_{0,s} = x_m$ and the claim holds for $e' = 0$ and $s' = t_{0,s} = x_m$, since for all $j \le m$, the tuple $(0, x_j, i_j)$ is chosen during step 2 of stage x_j. So assume that the claim holds for $e' < e$ and R_e receives attention at stage s via β and (4.5) holds. Now either e enters into Sat at stage x_m and the claim holds for $e' = e$ and $s' = x_m = t_{e,s}$. Or at the stage $\hat{s} = t_{e,s} = x_j$, there is an $\hat{e} < e$ such that $(\hat{e}, \hat{s}, 1 - i_j) \in Announce'_{\hat{s}}$, and this tuple is chosen during step 2. Therefore $R_{\hat{e}}$ received attention at some stage $t \le \hat{s}$ and by the induction hypothesis we know that there are $e' \le \hat{e} < e$ and a stage $s' \ge t_{\hat{e},t}$ such that $|s'| \le t(2(\hat{e}+1), |t|)$ and $e' \in Sat_{s'} \setminus Sat_{s'-1}$. Since at stage $\hat{s}$, the tuple $(\hat{e}, \hat{s}, 1 - i_j)$ is a member of $Announce'_{\hat{s}}$, it follows that $t_{\hat{e},t} \ge t_{e,s}$ and therefore $s' \ge t_{\hat{e},t} \ge t_{e,s} \ge s$. Moreover, since $|t| \le |t_{e,s}| \le 2^{2^{|s|}}$ and $\hat{e}+1 \le e$, it follows that $|s'| \le t(2(\hat{e}+1), 2^{2^{|s|}}) \le t(2(e+1), |s|)$ by definition of the function t (see (4.4)). □

Claim 5: Every requirement R_e receives attention at most finitely often.

Proof: Assume that R_e receives attention at infinitely many stages. Then we may choose a sequence $s_0, s_1, s_2, \ldots$ of stages such that for all $i \ge 0$, R_e receives attention at stage s_i and $|s_{i+1}| > t(2(e+1), |s_i|)$. Then Claim 4 implies that $Sat_{s_{i+1}} \setminus Sat_{s_i} \ne \emptyset$ and $\{0, \ldots, e\} \subseteq Sat_{s_{e+1}}$. Therefore R_e does not require attention at any stage $s > s_{e+1}$, contrary to the assumption. □

Claim 6: Every R_e is met

Proof: Assume that f_e^A is dense along G^A, but $e \notin Sat$. By Claim 5 there is a stage s_e such that $e < d(|s_e|)$ and for all $e' \le e$, $R_{e'}$ does not receive attention at any stage $s > s_e$. Then we may choose s such that

$$f_e^A(G^A \restriction s) \downarrow = (x_0, i_0), \ldots, (x_m, i_m) \qquad \text{and} \qquad |s| > 2^{2^{|s_e|}}.$$

By the choice of s, the requirement R_e is active at stage s. In case that R_e requires attention at stage s, some $R_{e'}$ receives attention at this stage. But the minimality of e' implies that $e' \le e$, a contradiction to the choice of $s > s_e$. Therefore R_e does not require attention at stage s. Choose $\hat{s}$ minimal such that $|\hat{s}| = 2^{2^{|s|}}$ and let $\beta = \alpha_{\hat{s}}$. Since $\beta \supseteq \varphi(A, e, s)$ but R_e does not require attention via β this implies that β is not an s-honest extension of α_{s-1} .

Then there is a minimal $s' \ge s$ such that $\beta(z) = 1$ for some $z \in \mathit{code}(s')$. (Note that Claim 2 implies that $s' \le \hat{s}$.) Then $\alpha_{s'-1}$ is s'-honest by Claim 2 while $\alpha_{s'} \subseteq \beta$ is not. Hence some tuple $(e', s', 1) \in \mathit{Announce}'_{s'}$ was selected during step 2 of stage s'. If $e' < e$, the assumption about s_e implies that $|s'| \le 2^{2^{|s_e|}} < |s|$, a contradiction to the choice of $s' \ge s$. Hence $e' \ge e$, and the choice of γ' during step 2 of stage s' implies that $\beta' = \alpha_{s'-1} \sqcup \varphi(\gamma', e, s)$ is an s'-honest extension of $\alpha_{s'-1}$, that $\beta' \subseteq \alpha_{s'}$, and that $f_e^{\beta'}(G^{\alpha_{s-1}} \restriction s) = f_e^{\alpha_{s'}}(G^{\alpha_{s-1}} \restriction s) = f_e^A(G^A \restriction s)$. Moreover, by minimality of s', β' is an s-honest extension of α_{s-1}, too. Since $f_e^{\beta'}(G^{\alpha_{s-1}} \restriction s) = f_e^A(G^A \restriction s)$ is defined, R_e requires attention via β' at stage s, a contradiction.

This completes the proof, since we have shown that for all $e \ge 0$ either f_e^A is not dense along G^A, or $e \in Sat$ and R_e is met by Claim 3. □

Claim 7: Condition (4.1) is satisfied.

Proof: Let $l(n) = \min\{m : d(m) = n\}$. We will show that for all sufficiently large n there is a number m with $l(n) \le m < 2^m \le l(n+1)$ such that $A(x) = K^A(x)$ for all strings x with $m \le |x| \le 2^m$.

Note that $l(n) \ge n$ whence, by (4.3), $l(n+1) \ge t(8n^2 + 8n, l(n))$. Hence, for l_i $(i = 0, \ldots, 4n)$ defined by $l_0 = l(n)$ and $l_{i+1} = t(2n+2, l_i)$, $l(n) = l_0 < l_1 < \cdots < l_{4n} \le l(n+1)$. On the other hand, only the n requirements

$R_0, \ldots, R_{n-1}$ can receive attention at a stage s for which $|s| < l(n+1)$. But if R_e receives attention at stage s and $|s| \leq l_k$, then, by Claim 4, some requirement $R_{e'}$ ($e' \leq e$) enters into *Sat* at some stage s' for which $|s| \leq |s'| \leq l_{k+1}$ and $R_{e'}$ does not receive attention at any stage greater than s'. So we can pick $k \leq 3n$ such that no requirement receives attention at any stage s with $l_k \leq |s| < l_{k+2}$. It follows by construction that $A(x) = K^A(x)$ for all strings $x < m(0^{l_{k+2}})$ with $x \notin \mathit{dom}(\alpha_{s_0})$ where $s_0 = 1^{l_k - 1}$. Since $|x| < l_{k+1}$ for $x \in \mathit{dom}(\alpha_{s_0})$ by Claim 2, this implies that $A(x) = K^A(x)$ for all strings x with $l_{k+1} \leq |x| < l_{k+2}$. □

This completes the proof of Theorem 4.1.1. □

Corollary 4.1.2. *For the strong hypotheses (G), (R), (GG) and (C) it holds that*

$$(G) \prec (R), \qquad (GG) \prec (C), \quad \textit{and} \quad (R) \not\preceq (GG).$$

Proof. This is a direct consequence of Theorem 4.1.1 together with the relativized versions of Theorems 2.3.11 and 2.3.9. □

4.2 Randomness versus Genericity

This section deals with the analog of Theorem 4.1.1 with p-randomness in place of general p-genericity and vice versa. As in the previous proof, there will be a structural property of sets such that, relative to any oracle, no general p-generic set has this property, but relative to the constructed oracle, every set in NP will have this property.

Definition 4.2.1. A set A is *locally n-2^{cn}-verbose* if there is an infinite set $B \in \mathrm{P}$ and a partial function $f : \Sigma^* \to \Sigma^*$ in $\mathrm{DTIME}(2^{cn})$ with domain B such that, for all $x \in B$, $|f(x)| = |x|$ and

$$f(x) \neq A(x)A(x+1)\ldots A(x+|x|-1). \tag{4.6}$$

I.e., f maps x to a string of length $|x|$ that differs in some bit from the section of the characteristic sequence of A of the same length that starts with the bit $A(x)$.

Lemma 4.2.2. *If A is a general p-generic set, then A is not locally n-2^{cn}-verbose for any $c \geq 1$.*

Proof. Fix $c \geq 1$ and suppose that A is locally n-2^{cn}-verbose via some funtion f. Define the n^c-extension function $\hat{f}$ by

$$\hat{f}(X \restriction x) = (x, f(x)[0]), (x+1, f(x)[1]), \ldots, (x+|x|-1, f(x)[|x|-1])$$

if $f(x)\downarrow$; and $\hat{f}(X \restriction x)$ undefined otherwise. Then $\hat{f}$ is an n^c-extension function since $f \in \mathrm{DTIME}(2^{cn})$; and $\hat{f}$ is dense along A since the domain of f is infinite. So A meets $\hat{f}$ at some x. But then

$$A(x+i) = f(x)[i]$$

for all $i < |x|$, a contradiction to (4.6). □

Since Lemma 4.2.2 relativizes, the property of locally n-2^{cn}-verboseness may be used to construct an oracle relative to which there is no general p-generic set in NP. For this it suffices to provide the relevant information coded into the oracle. Of course, a betting strategy with equal computational power can get this information from the oracle, too, so we have to argue that this knowledge does not signifficantly effect the chances to win.

Theorem 4.2.3. *There is an oracle A such that, relative to A,* NP *contains an n^2-random set but no general p-generic set.*

Proof. As in the proof of Theorem 4.1.1, we will effectivley enumerate a sequence of finite characteristic functions $(\alpha_s)_{s\geq 0}$ which has the characteristic function of A as its limit. The construction will guarantee that the set

$$G^A = \{x \,:\, \exists y\,(|y| = |x|^2 \,\&\, 1xy \in A)\}$$

will be n^2-random. This will be achieved by fixing A in a way such that d^A does not succeed on G^A where d is the relativized version of the martingale universal for the class of n^2-martingales given in Definition 2.3.6. Since $S^\infty[d^A]$ contains the success sets of all martingales computable in time $O(n^2)$ relative to A, this implies that G^A is n^2-random relative to A.

At the same time, for each $s = \langle i, j\rangle$, we will code a string

$$\omega_s \neq N_i^A(0^{\tau(s)})N_i^A(0^{\tau(s)}+1)\ldots N_i^A(0^{\tau(s)}+\tau(s)-1) \tag{4.7}$$

of length $\tau(s)$ into A. Hereby N_i is the i-th language in NP with respect to some effective numbering, and τ denotes the iterated power of 2, starting with $\tau(0) = 2^4$ and, for $s \geq 0$, $\tau(s+1) = 2^{\tau(s)}$. This coding will ensure that the function

$$f^A(x) = \begin{cases} \omega_s & \text{if } x = 0^{\tau(s)} \\ \text{undefined} & \text{otherwise} \end{cases}$$

will be computable in time $O(2^{cn})$ for some constant c. Thus (4.7) will imply that, relative to A, every set in NP will be n-2^{cn}-verbose, and therefore no set in NP can be general p-generic relative to A.

This coding will affect the behaviour of d^A on inputs $G^A \restriction x$ for $\tau(s) \leq |x| < \tau(s+1)$, but at each stage s we will choose α_s and the string to code ω_s such that for all such x

$$d^A(G^A \restriction x) = d^{\alpha_s}(G^{\alpha_s} \restriction x) \leq c_s \cdot d^A(G^A \restriction 0^{\tau(s)}) \tag{4.8}$$

$$\text{and } \limsup_{n\to\infty} \prod_{s\le n} c_s < \infty. \tag{4.9}$$

Hence d^A will not succeed on G^A, since (4.8) and (4.9) imply that

$$\limsup_{n\to\infty} d^A(G^A \restriction n) \le d^A(G^A \restriction 0^{\tau(0)}) \cdot \limsup_{n\to\infty} \prod_{s\le n} c_s < \infty.$$

Before sketching the idea of the construction, we introduce some notation.

For a string x call the set $G\text{-}code(x) = \{1xy : |y| = |x|^2\}$ the *coding region of* $G(x)$ and let $m(x)$ be the least element of $G\text{-}code(x)$. For $e \ge 0$ and strings x and y of the same length call the set $F\text{-}code(e, x, y) = \{0^{|x|+1}xyz : |z| = |x|^{e+2}\}$ the *F-coding region of e, x and y.*

To describe the dependence of $d^X(w)$ on the oracle X let $\varphi(X, d, w)$ be the use function of this computation, i.e.,

$$\begin{aligned}\varphi(X, d, w) \quad = \quad & \{(z, X(z)) : z \text{ is queried in the computation} \\ & \text{of } d^X(w).\}\end{aligned}$$

The small segments of the characteristic sequence of N_e^X we are interested in also depend on the oracle X, and we will refer to these segments by

$$\omega(X, e, x) = N_e^X(x)N_e^X(x+1)\ldots N_e^X(x+|x|-1).$$

Finally, given a finite partial function β, and a string z, the *z-variant of β*, denoted by $var(\beta, z)$ is the finite partial function

$$var(\beta, z) = \{(z, 1)\} \sqcup \beta.$$

At each stage s we want to fix α_s such that α_s extends $A \restriction m(0^{\tau(s+1)})$ and forces (4.7)–(4.9). Roughly speaking, this is achieved by first computing each and every sensible choice of α_s, then observing the consequence of each choice , and finally, selecting the "best" extension α_s in order to ensure (4.7) – (4.9).

In the first step of each stage $s = \langle e, j\rangle$ we select candidates for α_s, i.e. for each y of length $\tau(s)$ and each p of length $2^{\tau(s+1)} - 1$ that extends $G^{\alpha_{s-1}} \restriction 0^{\tau(s)}$ we fix the way how to choose α_s if we wanted to code $\omega_s = y$ and $G^{\alpha_s} \restriction 0^{\tau(s+1)} = p$. In the second step we compute $\omega(\beta, e, 0^{\tau(s)})$ for each such candidate β and choose ω_s such that $\omega(\beta, e, 0^{\tau(s)}) = \omega_s$ for as few β as possible. Then we replace each β considered in the first step by a corresponding $\gamma = var(\beta, z_s)$ for an a priori fixed string z_s coding ω_s. Finally, in the third step we examine the behavior of d^γ on the prefix strings of $G^\gamma \restriction 0^{\tau(s+1)}$ for all γ fixed during step 2 and fix α_s in correspondence to this behaviour, thereby ensuring (4.7) – (4.9).

Initially, let α_{-1} be the function that maps strings of length less than $\tau(0)$ to 0 and is undefined otherwise. Then the construction is as follows.

Stage s of the construction of A:

Let $x = 0^{\tau(s)}$, $g = G^{\alpha_{s-1}} \upharpoonright x$ and e such that $s = \langle e, j\rangle$ for some $j \geq 0$. For any string z let

$$E_z = \{p : |p| = 2^{\tau(s+1)} - 1 \,\&\, z \sqsubset p\}.$$

Note that E_g contains the strings $G^X \upharpoonright 0^{\tau(s+1)}$ for all $X \supseteq \alpha_{s-1}$.

Step 1: (Fixing extensions of α_{s-1})

For each y of length $|x|$ let z_y denote the least element of $F\text{-}code(e, x, y) \setminus dom(\alpha_{s-1})$ and let $W = \{z_y : |y| = |x|\}$.
For each string p of length $< 2^{\tau(s+1)}$ that has g as prefix we define an extension $\beta(p)$ of α_{s-1} inductively by

$$\beta(g) = \alpha_{s-1} \sqcup \bigcup_{\substack{z\in W\\ j\in\{0,1\}}} \Big(\varphi(var(\alpha_{s-1}, z), d, gj) \setminus (W\times\{0,1\})\Big)$$

$$\beta'(pi) = \beta(p) \cup \{(y, i)\}$$

where y is the least element of $G\text{-}code(s_{|pi|-1}) \setminus dom(\beta(p))$; and

$$\beta(pi) = \beta'(pi) \sqcup \bigcup_{\substack{z\in W\\ j\in\{0,1\}}} \Big(\varphi(var(\beta'(pi), z), d, pij) \setminus (W\times\{0,1\})\Big)$$

for $i \in \{0, 1\}$. Recall that s_n denotes the $(n + 1)$-st string in the length-lexicographic ordering. Hence, for each string p considered above, the construction ensures that the first $|p|$ bits of the characteristic sequence of $G^{\beta(p)}$ coincide with p. This is established by the choice of $\beta'(p)$, and the corresponding string y is chosen such that the adding of the pair (y, i) does not influence the behaviour of $d^{\beta'(p)}(pj)$ for either $j \in \{0, 1\}$. Moreover, $N_e^{\beta(p)}(y) = N_e^{var(\beta(p),z)}(y)$ for all $z \in W$ and $y \in \Sigma^{=|x|}$, since on inputs of length $|x|$ N_e does not query any string in W.

Step 2: (Coding ω_s)

For each string y of length $|x|$ let

$$freq(y) = |\{p \in E_g : \omega(\beta(p), e, x) = y\}|$$

be the number of (maximal) extensions defined in Step 1 that force the segment of N_e we are interested in to equal y. Then we may choose ω_s to be the least of the strings y with minimal $freq(y)$, i.e.

$$freq(\omega_s) \leq freq(y) \quad \text{for all } y \text{ of length } |x|. \tag{4.10}$$

Note that (4.10) implies that $freq(\omega_s) \leq \frac{|E_g|}{2^{|x|}}$.
Having chosen ω_s, we select the unique string $z_{\omega_s} \in W \cap F\textit{-code}(e, x, \omega_s)$ and fix for each string p considered in Step 1 a new extension $\gamma(p)$ of α_{s-1} by

$$\gamma(p) = var(\beta(p), z_{\omega_s}). \tag{4.11}$$

Step 3: (Choosing α_s)
We want to fix $p_0 \in E_g$ such that $\alpha_s \supseteq \gamma(p_0)$ will ensure (4.7)–(4.9). Obviously, we have to choose p_0 such that $\omega(\gamma(p_0), e, x) \neq \omega_s$. So call $p \in E_g$ "good" if $\omega(\gamma(p), e, x) \neq \omega_s$.
Initially, let $p_0 \restriction 0^{\tau(s)} = g$. For each n such that $2^{\tau(s)} - 1 \leq n < 2^{\tau(s+1)} - 1$ we will fix $p_0[n]$ by the following procedure, assuming that $p_0 \restriction n$ is already fixed. For $i = 0, 1$, compute

$$\epsilon_{i,n} = \frac{2 \cdot |\{p \in E_{(p_0 \restriction n)i} \,:\, p \text{ is "good"}\}|}{|\{p \in E_{p_0 \restriction n} \,:\, p \text{ is "good"}\}|} \tag{4.12}$$

Choose $i \in \{0, 1\}$ such that $\epsilon_{i,n} \neq 0$ and

$$d^{\gamma(p_0 \restriction n)}((p_0 \restriction n)i) \leq \epsilon_{i,n} \cdot d^{\gamma(p_0 \restriction n)}(p_0 \restriction n) \tag{4.13}$$

and fix $p_0[n] = i$. If both values for i have this property then let $p_0[n] = 0$. Using the resulting string $p_0 \in E_g$ we define

$$\alpha_s = \gamma(p_0) \sqcup \{(z, 0) \,:\, z < m(0^{\tau(s+1)})\}. \tag{4.14}$$

end of stage s

That A has the required properties is shown in the following claims.

Claim 1: For all $s \geq 0$, α_s is well defined, α_s extends α_{s-1}, and $dom(\alpha_s)$ contains all strings $z < m(0^{\tau(s+1)})$. Moreover, $dom(\alpha_s)$ contains at most $2^{\tau(s+1)} \cdot 2 \cdot 2^{\tau(s)} \cdot 2^{7 \cdot \tau(s+1)}$ strings $z \geq m(0^{\tau(s+1)})$ and $\alpha_s(z) = 0$ for all those z.

Proof. The proof follows by induction on s from the following observations.

(i) *W is well defined.*

By hypothesis, $dom(\alpha_{s-1})$ covers at most $2^{\tau(s)} \cdot 2 \cdot 2^{\tau(s-1)} \cdot 2^{7 \cdot \tau(s)}$ of the $2^{\tau(s)^{e+2}}$ elements of $F\textit{-code}(0^{\tau(s)}, y, e)$ (for $|y| = |x|$ and $s = \langle e, j \rangle$ for some j). The definition of τ ensures that

$$2^{\tau(s)^{e+2}} \geq 2^{\tau(s)^2} > 2^{9\tau(s)}, \tag{4.15}$$

whence for every y of length $|x|$ the corresponding string z_y can be selected.

(ii) *For each p considered in Step 1, $\beta(p)$ is well defined and extends α_{s-1}, $dom(\beta(p))$ contains at most $|p| \cdot 2 \cdot 2^{\tau(s)} \cdot 2^{7\cdot(|p|+1)}$ members $z \geq m(s_{|p|})$ and $\beta(p)(z) = 0$ for all those z.*

By induction on $|p|$, this follows easily from the construction and (i). Just note that the hypothesis about $dom(\beta(p))$ ensures that $\beta'(pi)$ is well defined, and $\beta(pi) \setminus \beta'(pi)$ consists of at most $2 \cdot 2^{\tau(s)} \cdot 2^{7\cdot|pi|}$ pairs $(z, 0)$.

(iii) *p_0 is well defined*

Assume n minimal such that $p_0[n]$ cannot be fixed according to the procedure given in Step 3. Then $p_0 \restriction n$ can be fixed according to this procedure and therefore the denominator in (4.12) is non-zero. Since $\epsilon_{0,n} + \epsilon_{1,n} = 2$, the martingale property of d ensures that at least one of $\epsilon_{0,n}$, $\epsilon_{1,n}$ is non-zero and fulfills (4.13).

Since $\gamma(p_0)$ differs from $\beta(p_0)$ only on the string z_{ω_s}, the analog of (ii) holds for $\gamma(p_0)$, too. By the definition given in (4.14), this proofs the claim for α_s. □

Claim 2: For all x such that $|x| < \tau(s+1)$, $G^A(x) = G^{\alpha_s}(x)$ and $d^A(G^A \restriction x) = d^{\alpha_s}(G^{\alpha_s} \restriction x)$. For all tuples $\langle e, j \rangle$ there is a unique string y of length $\tau(\langle e, j \rangle)$ such that $F(e, 0^{\tau(\langle e,j \rangle)}, y) \cap A \neq \emptyset$.

Proof. Fix some arbitrary $x < 0^{\tau(s+1)}$. By Claim 1, A extends α_s and G-$code(x) \subset dom(\alpha_s)$. This implies that $G^A(x) = G^{\alpha_s}(x)$ and $G^A \restriction x = G^{\alpha_s} \restriction x$. For $s' \leq s$ such that $\tau(s') \leq |x| < \tau(s'+1)$ let p_0' be the string of length $2^{\tau(s'+1)} - 1$ constructed in step 3 of stage s'. Then $G^A \restriction x = p_0' \restriction x$. Since $\alpha_{s'}$ extends $\delta = \beta'(p_0' \restriction (x-1))$ and $\alpha_{s'} \supset \varphi(var(\delta, z_{\omega_s}), d, p_0' \restriction x)$

$$d^{\gamma(p_0' \restriction (x-1))}(p_0' \restriction x) = d^{\alpha_s'}(G^{\alpha_s'} \restriction x) = d^{\alpha_s}(G^{\alpha_s} \restriction x) = d^A(G^A \restriction x). \quad (4.16)$$

Finally, note that $A(z) = 1$ implies that either z is an element of G-$code(x)$ for some x, or z was chosen in step 2 of some stage $s = \langle e, j \rangle$, which implies that $z \in F$-$code(e, 0^{\tau(s)}, y)$ for some y of length $\tau(s)$. Since at stage s exactly one such z is chosen and all the different coding regions used in the construction of A are disjoint, the corresponding string y is unique. □

Claim 3: N_e is n-2^{cn}-verbose for some constant c.

Proof. Consider the following algorithm for a function $f_e^A : \Sigma^* \to \Sigma^*$. On input x, check that $x = 0^{\tau(\langle e,j \rangle)}$ for some $j \geq 0$. Find y of length $|x|$, such that there exists $z \in A \cap F$-$code(x, y, e)$ and z is one of the first $2^{9|x|} \geq 2^{\tau(\langle e,j \rangle)} \cdot 2 \cdot 2^{\tau(\langle e,j \rangle - 1)} \cdot 2^{7\cdot\tau(\langle e,j \rangle)}$ members of F-$code(x, y, e)$. If such a y exists, then output y, otherwise let $f_e^A(x)$ be undefined. Since the elements of F-$code(x, y, e)$ are of length $3|x| + 1 + |x|^{e+2}$ and f_e^A has to make at most $2^{|x|} \cdot 2^{9\cdot|x|}$ queries, f_e^A is computable in time $O(2^{10|x|})$, and $f_e^A(0^{\tau(\langle e,j \rangle)}) \downarrow= \omega_{\langle e,j \rangle}$ as defined during stage $s = \langle e, j \rangle$ by claim 2. It remains to show that the string p_0 constructed during this stage is "good". So assume that p_0 is not "good". Then there exists a minimal prefix z of p_0 such that $p_0 \restriction 0^{\tau(\langle e,j\rangle)} \sqsubset z \sqsubseteq p_0$ and no element of E_z is "good". Fix n such that $z = (p_0 \restriction n)i$ for some $i \in \{0,1\}$. Then $\epsilon_{i,n} = 0$ and $\epsilon_{1-i,n} \neq 0$ by minimality of z. Hence $p_0[n] = 1 - i$, a contradiction to $z \sqsubseteq p_0$. □

Claim 4: d^A does not succeed on G^A.

Proof. We want to show (4.8) for $c_s = \frac{\tau(s+1)}{\tau(s+1)-1}$. Note that (4.16) and (4.13) imply

$$d^A(G^A \restriction (n+1)) = d^{\gamma(p_0 \restriction n)}(p_0 \restriction (n+1)) \leq \epsilon_{i,n} \cdot d^{\gamma(p_0 \restriction n)}(p_0 \restriction n)$$

for $i = p_0[n]$. Hence

$$d^A(G^A \restriction (n+1)) \leq \prod_{k=2^{\tau(s)}-1}^{n} \epsilon_{p_0[k],k} \cdot d^A(G^A \restriction 0^{\tau(s)}).$$

Again, let $g = G^A \restriction 0^{\tau(s)}$. Then the number of "good" strings in E_g is at least $|E_g| - freq(\omega_s) \geq |E_g| - |E_g|/(2^{\tau(s)}) = |E_g|(1 - 1/\tau(s+1))$. Hence

$$\begin{aligned}
\prod_{k=2^{\tau(s)}-1}^{n} \epsilon_{p_0[k],k} &= 2^{n+2-2^{\tau(s)}} \cdot \frac{|\{p \in E_{p_0 \restriction (n+1)} : p \text{ is "good"}\}|}{|\{p \in E_g : p \text{ is "good"}\}|} \\
&\leq 2^{n+2-2^{\tau(s)}} \cdot \frac{|\{p \in E_{p_0 \restriction (n+1)} : p \text{ is "good"}\}|}{|E_g|(1 - 1/\tau(s+1))} \\
&\leq 2^{n+2-2^{\tau(s)}} \cdot \frac{|E_{p_0 \restriction (n+1)}|}{|E_g|(1 - 1/\tau(s+1))} \\
&= \frac{2^{n+2-2^{\tau(s)}+2^{\tau(s+1)}-n-2}}{|E_g|(1 - 1/\tau(s+1))} \\
&= \frac{|E_g|}{|E_g|(1 - 1/\tau(s+1))} \\
&= \frac{\tau(s+1)}{\tau(s+1)-1} = 1 + \frac{1}{\tau(s+1)-1}
\end{aligned}$$

Since $\sum_{s=1}^{\infty} 1/(\tau(s)-1) < \infty$, $\prod_{s=1}^{\infty}(1 + 1/(\tau(s)-1)) < \infty$, as well, whence

$$\limsup_{n\to\infty} d^A(G^A \restriction n) \leq d^A(G^A \restriction 0^{\tau(0)}) \cdot \prod_{s=1}^{\infty}(1 + \frac{1}{(\tau(s)-1)}) < \infty.$$

□

This completes the proof of Theorem 4.2.3. □

Corollary 4.2.4. *For the strong hypotheses (G), (R), (GG) and (C) it holds that*

$$(G) \prec (GG), \qquad (R) \prec (C), \qquad \text{and} \quad (GG) \not\preceq (R).$$

Proof. Again, this is a direct consequence of Theorem 4.2.3 toghether with the relativized versions of Theorems 2.3.11 and 2.2.19. □

Chapter 5

Separating NP-Completeness Notions under Strong Hypotheses

5.1 Introduction

In this chapter, a great variety of NP-completeness notions are studied under the hypotheses (G) and (GG). It was shown in [AmbFH87] that under the hypothesis (G), there are sets in NP witnessing differences between most of the polynomial time reducibilities, and these sets can often be defined in a straightforward way.

Proposition 5.1.1. *([AmbFH87]) Assume (G). Let S be a p-generic set in* NP. *Then*

(i) $S \not\leq^{\mathrm{p}}_{1-li} S$,

(ii) $S \oplus S \not\leq^{\mathrm{p}}_{1} S$,

(iii) for all $k \geq 2$, $S_k \not\leq^{\mathrm{p}}_{btt(k-1)} S$ *for* $S_k = \{x : \{x+1, \ldots, x+k\} \cap S \neq \emptyset\}$,

(iv) $S_\infty \not\leq^{\mathrm{p}}_{btt} S$ *for* $S_\infty = \{\langle x, i\rangle : i \geq 2 \,\&\, x \in S_i\}$.

Sketch of proof. Here only the idea for a proof of (ii) is given. Suppose that $S \oplus S \leq^{\mathrm{p}}_{1} S$ via the one-to-one function g. Then consider the following computation. On input $X \restriction 0^n$ evaluate $q_0 = g(00^n)$ and $q_1 = g(10^n)$. Since g is one-to-one, at least one of these strings differs from 0^n. Denote this differing string by q. If $|q| < n$ then output $(0^n, 1 - X(q))$. Otherwise, output $(0^n, 1), (q, 0)$. This computation pertains to a bounded n^2-extension function f which is dense along every set. By assumption about g, $S(0^n) = S(g(00^n)) = S(g(10^n)) = S(q)$. However, if S meets f at 0^n for some $n \geq 0$ then $S(0^n) = 1 - S(q)$. Thus S does not meet f, a contradiction to p-genericity of S. $\square$

This brief sketch illustrates some of the features of the subsequent proofs. First a reduction M between two sets A_1 and A_2 that depend on a p-generic set A is assumed. Then an infinite sequence of strings x_n $(n \geq 0)$ is fixed. (In many cases $x_n = 0^n$.) Finally, a thorough analysis of the queries made by M on input x_n leads to the definition of an extension function that is defined on every string of the form $X \restriction x_n$. However, if the p-generic set meets this extension function at some x_n, this x_n will witness the fact that M can not be a reduction between A_1 and A_2, a contradiction.

The success of such a construction depends on the relation between the p-generic set A and the sets A_1 and A_2. If the defined extension function f obeys the appropriate bounds, then A will meet f at some x_n. Hence $f(X \restriction x_n)$ should be defined such that A meeting f at x_n implies $A_1(x_n) \neq M^{A_2}(x_n)$. To ensure this, the sets A_i $(i = 1, 2)$ must depend on the p-gneric set A in a specific way.

Definition 5.1.2. Let A and B be sets. B *dominates* A if there is a total function $f : \Sigma^* \to 0, 1$ in P and a polynomial time-bounded oracle Turing machine M such that

$$A(x) = i \qquad \text{whenever} \qquad \langle x, f(x) \rangle \in L(M, B). \tag{5.1}$$

B *completely dominates* A if there are Turing machines M_0 and M_1 such that B dominates A via M_0 and the constant function $f(x) \equiv 0$, and via M_1 and the constant function $f(x) \equiv 1$ as well.

Definition 5.1.3. Let A_1, A_2 and B be sets. A_1 and A_2 are *independently dominated by* B, if there exist f_1, f_2, M_1 and M_2 such that B dominates A_i via M_i and f_i $(i = 1, 2)$, and for every x the two sets of queries made by M_1 on input $\langle x, f_1(x) \rangle$ and by M_2 on input $\langle x, f_2(x) \rangle$ are disjoint.

Obvioulsy, if A is polynomial time reducible to B, then B completely dominates A. In addition, two independent extractions A_{f_0} and A_{f_1} of a set A are independently dominated by A. Other examples important in the following are unions and intersections $A \cup B$ and $A \cap B$ which are both dominated (but not completely dominated) by B. This holds since $(A \cup B)(x) = 1$ whenever $B(x) = 1$, and $(A \cap B)(x) = 0$ whenever $B(x) = 0$.

These considerations together with the observation of Theorem 2.2.17 constitute the basis for the proofs in this chapter. Therefore, they all share a general pattern that will be discussed in advance.

Consider two polynomial time reducibilities r_1 and r_2. Under the hypothesis (G), an NP-r_2-complete set shall be given that is not NP-r_1-complete. For this sake, fix a p-generic set G in NP and an NP-m-complete set $C \in \mathrm{DTIME}(2^n)$. G and C will be used to define the desired set $A \in$ NP. That A has the desired properties will be achieved by selecting sets G_1 and G_2, which are completely and independently dominated by G, and defining A such that

(1) A is *positively* P-*btt*-reducible to $G_1 \oplus C$,

(2) C can be recovered from A by a r_2-reduction,

(3) A is dominated by G_1, and thus dominated by G, and

(4) p-genericity of G and (3) imply that G_2 is *not* r_1-reducible to A.

In case that G_1 and G_2 are members of NP, (1) implies that $A \in$ NP, (2) implies that A is NP-r_2-hard, and (4) implies that A is not NP-r_1-hard. In the proofs of this chapter, (1)−(3) will be quite clear from the definition of A, whereas establishing (4) often requires more careful and envolved arguments. While G_1 and G_1 are *completely* dominated by G, by the dependence on C this does not hold for the constructed set A. In order to show that G_1 is not r_1-reducible to A, thus the dependence of A on the complete set C has to be overcome by exploiting (5.1). To illustrate this idea, and to introduce the notation used in this chapter, first we discuss an example. The construction is taken from [LutM96] where it is discussed under the hypothesis (R), and was adapted for the weaker hypothesis (G) by Ambos-Spies in [Amb94b].

Example 5.1.4. Assume (G). Fix G and C in NP as above. Let

$$\begin{aligned}&G_0 = \{x : 0x \in G\},\\ &G_1 = \{x : 1x \in G\},\\ &A = G_0 \oplus ((G_0 \cap C) \oplus (G_0 \cup C)).\end{aligned}$$

Then A is NP-*btt*(3)-complete, but not NP-m-complete.

Proof. Since G_0 and G_1 are members of NP, $A \in$ NP by standard closure properties of NP. Note that

$$\begin{aligned}x \in C \;&\Leftrightarrow\; x \in G_0 \cap C \quad \text{or} \quad [x \notin G_0 \;\&\; x \in G_0 \cup C]\\ &\Leftrightarrow\; 10x \in A \quad\quad \text{or} \quad [0x \notin A \;\&\; 11x \in A],\end{aligned}$$

whence $C \leq^{\mathrm{p}}_{btt(3)} A$. Therefore (1) and (2) hold for the set A. For property (3), consider any string x. Then $A(0x) = i$ if $G_0(x) = i$ for every $i \in \Sigma$. Since $A(10x) = (G_0 \cap C)(x)$, $A(10x) = 0$ if $G_0(x) = G(0x) = 0$. And since $A(11x) = (G_0 \cup C)(x)$, $A(11x) = 1$ if $G_0(x) = G(0x) = 1$ (see the above remarks about unions and intersections).

Now suppose that $G_1 \leq^{\mathrm{p}}_m A$ via some P-m-reduction M. Let $x_n = 0^n$ for $n \geq 0$. Then

$$G_1(x_n) = M^A(x_n) \qquad \text{for all } n \geq 1. \tag{5.2}$$

Since G dominates A, we can define a bounded p-extension function f such that G meeting f at x_n for some n will imply that

$$G_1(x_n) = 1 - M^A(x_n), \tag{5.3}$$

a contradiction to (5.2). Hence p-genericity of G implies that G_1 is *not* P-m-reducible to A.

The definition of f consists of several steps. Given $n \geq 1$ and the initial segment $X \restriction x_n$ of the characteristic sequence of a set X, in the first step define a function $\alpha_n^X : Q(M, x_n) \to \Sigma$ of *intended oracle answers.* In the example, let

$$\alpha_n^X(q) = \begin{cases} X(0w) & \text{if } q = 0w \text{ and } |q| < n \\ X(0w) \cdot C(w) & \text{if } q = 10w \text{ and } |0w| < n \\ \max\{X(0w), C(w)\} & \text{if } q = 11w \text{ and } |0w| < n \\ 1 & \text{if } q = 11w \text{ and } |0w| \geq n \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

Note that α_n^X only depends on $X \restriction x_n$ and $C^{<|x_n|}$. I.e., if $Y \restriction x_n = X \restriction x_n$ for some set Y, then $\alpha_n^Y = \alpha_n^X$.

Measured in the length of $X \restriction x_n$, the finite function α_n^X is computable in polynomial time since $C \in \mathrm{DTIME}(2^n)$. Also note that $\alpha_n^G(q) = A(q)$, if one of the first three cases in the definition of $\alpha_n^G(q)$ applies. In the second step, for each $q \in Q(M, x_n)$ define a set of conditions $COND_n(q)$ ensuring that $A(q) = \alpha_n^G(q)$, whenever $G(z) = i$ for each tuple (z, i) in $COND_n(q)$. In our case, $COND_n(q) = \emptyset$ if $|q| < n$ or $q = 1q'$ and $|q'| < n$. Otherwise, $\alpha_n^G(q) = 1$ only if $q = 11w$ for some string w. Hence to ensure that $A(q) = \alpha_n^G(q)$, let $COND_n(q) = \{(0w, 1)\}$ if $q = 11w$ and $COND_n(q) = \{(0w, 0)\}$ if $q = 0w$ or $q = 10w$. (See the proof that G dominates A above.) Let

$$i_n^X = M^{\alpha_n^X}(x_n), \tag{5.5}$$

and

$$COND_n^X = \{(1x_n, 1 - i_n^X)\} \cup \bigcup_{q \in Q(M, x_n)} COND_n(q). \tag{5.6}$$

Note that $COND_n^X = \{(y_l, i_l) : l \leq m\}$ for some $m \geq 0$, where $x_n \leq y_0 < y_1 < \cdots < y_m$. So let

$$f(X \restriction x_n) = (y_0, i_0), \ldots, (y_m, i_m),$$

and let f be undefined on strings not of the form $X \restriction x_n$. f is called the *extension function induced by* $COND_n^X$. For this construction it is crucial that the set $COND_n^X$ does not contain conflicting tuples $(z, 0)$ and $(z, 1)$ for some z. In the example this can not occur, since G_0 and G_1 form an independent pair as introduced in Theorem 2.2.17, and the cardinality of $Q(M, x_n)$ is one for every $n \geq 0$.

It remains to show that G meeting f at x_n will imply (5.3), that f is a bounded n^c-extension function for some $c \geq 1$, and that f is dense along

G. In our example, $|COND_n^X| \leq 2$ and, given $X \upharpoonright x_n$ as input, $COND_n^X$ is computable in polynomial time by the above remarks about α_n^X. Hence f is a bounded n^c-extension function for some $c \geq 1$ and is dense along every set. So assume that G meets f at x_n. Then $G(z) = i$ for every $(z, i) \in COND_n^G$. If q is the query made by M on input x_n, either one of the first three cases in the definition of α_n^X applies and $A(q)$ is determined by $G \upharpoonright x_n$ and C (see (5.4)), in which case

$$A(q) = \alpha_n^G(q)$$

or one of the last two cases applies, in which case

$$A(q) = \left\{ \begin{array}{lll} G_0(w) = G(0w) = 0 & \text{if} & q = 0w \\ (G_0 \cap C)(w) = 0 & \text{if} & q = 10w \\ (G_0 \cup C)(w) = 1 & \text{if} & q = 11w \end{array} \right\} = \alpha_n^G(q)$$

by the definition of $COND_n(q)$. But this implies $M^A(x_n) = M^{\alpha_n^G}(x_n) = i_n^G$, whereas the definition of $COND_n^G$ implies that $G_1(x_n) = G(1x_n) = 1 - i_n^G$ since G meets f at x_n. Hence G meeting f at x_n implies (5.3). □

5.2 One Decisive Query

In this section we compare the NP-completeness notions induced by the bounded query reducibilities of fixed norm under the genericity assumption (G). We first consider the case of nonadaptive reductions. Recall that a *polynomial-time bounded-truth-table reduction of norm k (P-btt(k)-reduction* for short) of a set A to a set B is given by polynomial-time functions $h : \Sigma^* \to \Sigma$ (*evaluator*) and $g_1, \ldots, g_k : \Sigma^* \to \Sigma^*$ (*selectors*) such that $A(x) = h(x, B(g_1(x)), \ldots, B(g_k(x)))$ for all x. We write $A \leq_{btt(k)}^{\mathrm{P}} B$ if there is a *P-btt*(k)-reduction from A to B, and a set $B \in \mathrm{NP}$ is NP-*btt*(k)-*complete* if $A \leq_{btt(k)}^{\mathrm{P}} B$ for all $A \in \mathrm{NP}$.

Of central interest in this section will be the following set A_k, which is based on the same idea as the set in Example 5.1.4.

Definition 5.2.1. Let $k \geq 2$ and let G and C be sets. Let $z_1, \ldots, z_{k+1}$ be the first $k+1$ strings of length k, let

$$\widehat{G}_m = \{x : xz_m \in G\} \quad (1 \leq m \leq k)$$

and let

$$\widehat{G} = \bigcup_{m=1}^{k-1} \widehat{G}_m$$

be the union of the first $k-1$ of these sets. Then define A_k as the disjoint union of the $k+1$ sets $\widehat{G}_1, \ldots, \widehat{G}_{k-1}, \widehat{G} \cap C$ and $\widehat{G} \cup C$:

$$A_k = \bigcup_{m=1}^{k-1} \{xz_m : x \in \widehat{G}_m\} \cup \{xz_k : x \in \widehat{G} \cap C\} \cup$$
$$\cup \{xz_{k+1} : x \in \widehat{G} \cup C\}.$$

Note that the sets $\widehat{G}_m$ ($1 \leq m \leq k$) are independently dominated by G. For $k=2$, the set A_2 is essentially the *P-btt(3)*-complete set considered in Example 5.1.4, up to the different coding. As there, it is easy to see that, for every $k \geq 2$, C is *P-btt(k+1)*-reducible to A_k.

Lemma 5.2.2. *If $G \in$ NP and C is P-m-complete for NP then A_k is P-btt(k+1)-complete for NP ($k \geq 2$).*

Proof. Since G and C are members of NP, $A_k \in$ NP by standard closure properties of NP. And $C \leq^{\mathrm{P}}_{btt(k+1)} A_k$ since

$$\begin{array}{lllll} x \in C & \Leftrightarrow & x \in \widehat{G} \cap C & \text{or} & [x \notin \widehat{G} \;\&\; x \in \widehat{G} \cup C] \\ & \Leftrightarrow & x \in \widehat{G} \cap C & \text{or} & [\forall 1 \leq i < k \;(x \notin \widehat{G}_i) \;\&\; x \in \widehat{G} \cup C] \\ & \Leftrightarrow & xz_k \in A_k & \text{or} & [\forall 1 \leq i < k \;(xz_i \notin A_k) \;\&\; xz_{k+1} \in A_k] \end{array}$$

whence A_k is *P-btt(k+1)*-hard for NP. □

If only k non-adaptive queries to A are allowed, reducing C to A appears difficult. Consider strings x and y such that $x \in C \backslash \widehat{G}$ and $y \in \widehat{G} \backslash C$. Which queries to A could help to detect that $x \in C$ and $y \notin C$? Note that at least one of the queries made in the above proof is missing then. By intuition, if the query $q = yz_k$ is missing on input y, the reduction will not be able to interpret the answers to the remaining queries properly. And analogously, the same should hold if the query $q = xz_{k+1}$ is missing on input x. The main Theorem of this section shows that this intuition is not misleading.

Theorem 5.2.3. *Assume (G) and let $k \geq 2$. There is an* NP-*btt*$(k+1)$-*complete set A which is not* NP-*btt*(k)-*complete.*

To see that the sets A_k ($k \geq 2$) (with G a p-generic set) have the desired properties, by Lemma 5.2.2 it remains to show that A_k is not NP-*btt*(k)-complete. This can be proved in analogy to the discussion of Example 5.1.4. So first the dependence of A_k on G is discussed.

Lemma 5.2.4. *For every $k \geq 2$, the set A_k is dominated by the corresponding set G.*

Proof. Corresponding to the parts $\widehat{G}_1, \dots, \widehat{G}_{k-1}$, $\widehat{G} \cap C$ and $\widehat{G} \cup C$ of which A_k is composed, a string $w \in A_k$ is of the form vz_p, $1 \leq p \leq k+1$, where p indicates that membership of w in A_k is determined by the pth component. For $w = vz_p$ we call v the *value* and p the *index* of w. Then the relation between $A_k(w)$ and G depends on the index of w as follows

$$1 \leq p \leq k-1 \Rightarrow A_k(vz_p) = G(vz_p) \tag{5.7}$$

$$\begin{aligned} vz_k \in A_k &\Leftrightarrow v \in C \ \& \ \exists l(1 \leq l \leq k-1 \,\&\, vz_l \in A_k) \\ &\Leftrightarrow v \in C \ \& \ \exists l(1 \leq l \leq k-1 \,\&\, vz_l \in G) \end{aligned} \tag{5.8}$$

$$\begin{aligned} vz_{k+1} \in A_k &\Leftrightarrow v \in C \ \vee \ \exists l(1 \leq l \leq k-1 \,\&\, vz_l \in A_k) \\ &\Leftrightarrow v \in C \ \vee \ \exists l(1 \leq l \leq k-1 \,\&\, vz_l \in G) \end{aligned} \tag{5.9}$$

$$p > k+1 \Rightarrow vz_p \notin A_k \tag{5.10}$$

By (5.7), for w with index $1 \leq p < k$, $A_k(w) = i$ (for $i = 0, 1$) if $G(w) = i$. If the index of w is k then we can only control the second conjunct on the right side of (5.8): if $G(vz_l) = 0$ for all $1 \leq l < k$, then $A(vz_k) = 0$. By the dependence on C, however, here in general it is not possible to force $A_k(vz_k) = 1$. Dually, by (5.9), for w with index $k+1$, $A_k(vz_{k+1}) = 1$, if $G(vz_l) = 1$ for some $1 \leq l < k$, whereas $A_k(vz_{k+1}) = 0$ in general cannot be forced. □

So, for queries w, w' with index k and $k+1$, we try to force $A_k(w) = 0$ and $A_k(w') = 1$, respectively, by the appropriate definition of $COND(w)$ and $COND(w')$.

However, for strings $w = vz_k$ and $w' = vz_{k+1}$ of index k and $k+1$ with the same value v the above strategies for forcing $A_k(w) = 0$ and $A_k(w') = 1$ are not compatible. In order to force these values here, we have to know the value of $C(v)$: If $C(v) = 0$ then $A_k(w) = 0$ by failure of the first conjunct, whence we can force $A_k(w') = 1$ as above. Dually, for $C(v) = 1$, $A_k(w') = 1$ is immediate by the first disjunct in (5.9) whence here it suffices to force $A_k(w) = 0$. However, computing $C(v)$ for a query $vz_k \in \{g_1(x_n), \dots, g_k(x_n)\}$ might need $2^{n^{O(1)}}$ many steps since the length of the query is polynomial in the length of x_n. So for the following it will be important to note that if this situation applies to strings w, w' queried by a *P-btt*-reduction of norm k, and we have to know $C(v)$ in order to decide whether some string vz_l, $1 \leq l \leq k-1$, has to be forced into G (and thereby into A_k!) then we can choose l so that $vz_l \notin QUERY$. Namely, since vz_k and vz_{k+1} are both among the k queries at least one of the $k-1$ strings $vz_1, \dots, vz_{k-1}$ will not be queried.

Now we are ready to prove Theorem 5.2.3. It remains to show that A_k is not NP-$btt(k)$-complete, which will be the most involved part of the proof. Since $\widehat{G}_k \in \mathrm{NP}$ it suffices to show that $\widehat{G}_k \not\leq^{\mathrm{p}}_{btt(k)} A_k$.

Proof of Theorem 5.2.3. Fix a p-generic set $G \in \mathrm{NP}$ and an NP-m-complete set $C \in \mathrm{DTIME}(2^n)$. With these sets, let $\widehat{G}_k$ and A_k as in Definition 5.2.1. For a contradiction assume that $\widehat{G}_k \leq^{\mathrm{p}}_{btt(k)} A_k$ via $(h, g_1, \dots, g_k)$, i.e.,

$$\widehat{G}_k(x) = h(x, A_k(g_1(x)), \dots, A_k(g_k(x))) \tag{5.11}$$

for all strings x.

Note that the set $\widehat{G}_k$ is not used in the definition of A_k and that these sets are independently dominated by G. (See the proof of Lemmma 5.2.4.) In the following we will use p-genericity of G to refute (5.11). We will define a waiting bounded extension function f such that (5.11) will fail for $x_n = 0^n$ if G will meet f at 0^n. So for given n and $X \upharpoonright 0^n$ again a set $COND^X_n$ is defined such that

$$X \upharpoonright 0^n = G \upharpoonright 0^n \quad \& \quad \forall (y, i) \in COND^G_n \quad (G(y) = i) \tag{5.12}$$

will imply

$$\widehat{G}_k(0^n) \neq h(0^n, A_k(g_1(0^n)), \dots, A_k(g_k(0^n))). \tag{5.13}$$

Let $x_n = 0^n$ and let $QUERY_n$ be the set of queries made by the *P-btt(k)*-reduction $[h, g_1, \dots, g_k]$ on input x_n.

For given n and $X \upharpoonright x_n$ we first define the function $\alpha^X_n : QUERY_n \to \{0,1\}$ specifying the intended values for A_k on these queries. Fix $w = vz_p \in QUERY_n$. For the definition of $\alpha^X_n(w)$ we distinguish between short and long strings w. If $|w| < n$ let

$$\alpha^X_n(w) = \begin{cases} X(w) & \text{if } 1 \leq p \leq k-1 \\ C(v) \cdot \max_{1 \leq l < k}\{X(vz_l)\} & \text{if } p = k \\ \max(\{X(vz_l) : 1 \leq l < k\} \cup \{C(v)\}) & \text{if } p = k+1 \end{cases}$$

It is easy to check that in this case $A_k(w) = \alpha^G_n(w)$ by (5.7)–(5.10). For w with $|w| \geq n$ let

$$\alpha^X_n(w) = \begin{cases} 1 & \text{if } p = k+1 \\ 1 & \text{if } [p = k-1 \;\& \\ & \quad \forall l \in \{1, \dots, k-1, k+1\}(vz_l \in QUERY_n)] \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

And for each w not of the form $w = vz_p$ for some $1 \leq p \leq k+1$, let $\alpha^X_n(w) = 0$.

For w with $|w| \geq n$, $A_k(w) = \alpha_n^G(w)$ will be ensured assuming (5.12) by the definition of the set $COND_n^X$ of forcing conditions which we will give next.

For any value v with $|v| + k \geq n$, and such that there is a query $vz_p \in QUERY_n$ for some p with $1 \leq p \leq k+1$, we will define a set $COND_n(v)$ of forcing conditions which will be part of $COND_n^X$. Given such a value v let

$$IND(v) = \{p : \ 1 \leq p \leq k+1 \,\&\, vz_p \in QUERY_n\}$$

be the set of indices of queried strings with this value. Let

$$r = \mu s \geq 1\,(s \notin IND(v) \ \vee \ s = k-1)$$

be the parameter indicating whether there is an index $p < k-1$ such that vz_p is not queried. (Note that $r = k-1$ if such an index does not exist.) Then the definition of $COND_n(v)$ depends on $IND(v)$ and r as follows:

$$k+1 \notin IND(v) \implies COND_n(v) = \{(vz_1, 0), \ldots, (vz_{k-1}, 0)\} \tag{5.15}$$

$$\begin{aligned} &k+1 \in IND(v) \,\&\, k \notin IND(v) \implies \\ &\qquad COND_n(v) = \{(vz_i, j) : \ 1 \leq i \leq k-1 \,\&\, (j = 1 \Leftrightarrow i = r)\} \end{aligned} \tag{5.16}$$

$$\begin{aligned} &k+1 \in IND(v) \,\&\, k \in IND(v) \implies \\ &\quad COND_n(v) = \{(vz_i, j) : 1 \leq i \leq k-1 \,\& \\ &\qquad\qquad \& \, (j = 0 \text{ if } i \neq r) \,\&\, (j = 1 - C(v) \text{ if } i = r)\} \end{aligned} \tag{5.17}$$

Note that in case of (5.17), $r \notin IND(v)$, i.e., $vz_r \notin QUERY_n$, since, by $|IND(v)| \leq |QUERY_n| = k$, $k \in IND(v)$ and $k+1 \in IND(v)$ imply that $s \notin IND(v)$ for some $s \in \{1, \ldots, k-1\}$.

To show that, assuming (5.12), this part of the definition of $COND_n^X$ ensures that $A_k(w) = \alpha_n^G(w)$ for strings $w \in QUERY_n$ with $|w| \geq n$ and value v, fix such a string $w = vz_p$ and distinguish the following cases depending on the index of x.

Case 1: $p < k-1$. Then $\alpha_n^G(w) = 0$ and $(vz_p, 0) \in COND_n(v)$ whence $A_k(w) = 0$ by (5.12) and (5.7).

Case 2: $p = k-1$ and $IND(v) = \{1, \ldots, k-1, k+1\}$. Then $\alpha_n^G(w) = 1$. Moreover, $r = k-1$ whence, by (5.16), $(vz_{k-1}, 1) \in COND_n^X$. So $A_k(w) = 1$ by (5.12) and (5.7).

Case 3: $p = k-1$ and $IND(v) \neq \{1, \ldots, k-1, k+1\}$. Then $\alpha_n^G(w) = 0$. Moreover, either $k+1 \notin IND(v)$ whence case (5.15) applies or $r \neq k-1$. So in any case $(vz_{k-1}, 0) \in COND_n(v)$ whence $A_k(w) = 0$.

Case 4: $p = k$. Then $\alpha_n^G(w) = 0$. If $v \notin C$ then $A_k(w) = 0$ is immediate by (5.8). So assume $v \in C$, i.e., $1 - C(v) = 0$. Since for the definition of $COND_n(v)$ (5.15) or (5.17) applies, it follows that $COND_n(v) = \{(vz_1, 0), \dots, (vz_{k-1}, 0)\}$ whence $G(vz_l) = 0$ for all $l \in \{1, \dots, k-1\}$ by (5.12). Hence $A_k(w) = 0$ by (5.8).

Case 5: $p = k + 1$. Then $\alpha_n^G(w) = 1$. If $v \in C$ then $A_k(w) = 1$ is immediate by (5.9). So assume $v \notin C$, i.e., $1 - C(v) = 1$. Since case (5.16) or (5.17) applies, $(vz_r, 1) \in COND_n(v)$ whence $vz_r \in G$ by (5.12). It follows that $A_k(w) = 1$ by (5.9).

This completes the proof that, assuming (5.12), the above defined sets $COND_n(v)$ force $A_k(w) = \alpha_n^G(w)$ for all queries w with value v. Let

$$i_n^X = h(x, \alpha_n^X(g_1(x_n)), \dots, \alpha_n^X(g_k(x_n))) \tag{5.18}$$

So, by letting $COND_n^X$ be the union of all condition sets $COND_n(v)$ wherby v is the value of some long query in $QUERY_n$ together with the final condition $\{(0^n z_k, 1 - i_n^X)\}$, (5.12) will imply (5.13) since $\widehat{G}_k(0^n) = G(0^n z_k)$.

Since $COND_n^X$ does not contain any conflicting conditions, it only remains to show that the extension function induced by $COND_n^X$ is a waiting bounded p-extension function.

For this sake, first observe that $|COND_n^X| \leq k^2$ since besides the last condition $(0^n z_k, 1 - i_n^X)$ forcing the desired value of the left side of (5.13), for any of the at most k different values v attained by some query in $QUERY_n$, $k-1$ conditions were added to $COND_n^X$. Moreover, the positions $y_0, \dots, y_m$ of the conditions can be computed in $poly(n)$ steps. Finally, given $X \upharpoonright 0^n$, for the computation of the value i of a condition (y, i) $O(2^{|y|})$ steps suffice: If (y, i) is added to $COND_n^X$ via (5.15), (5.16) or (5.17) then obviously $poly(|y|)$ steps suffice unless $(y, i) = (vz_r, 1 - C(v))$ in case (5.17) where the claim follows from $C \in DTIME(2^n)$. This leaves the final condition $(0^n z_k, 1 - i_n^X)$. Here, by (5.18), it suffices to show that $\alpha_n^X(w)$ for $w \in QUERY_n$ can be computed in $O(2^n)$ steps. But for $\alpha_n^X(w)$ defined by (5.2) this follows by $|w| < n$ and $C \in DTIME(2^n)$ while for $\alpha_n^X(w)$ defined by (5.14) obviously $\alpha_n^X(w)$ is computable in $poly(n)$ steps.
This completes the proof of Theorem 5.2.3. □

Next we will prove the analog of Theorem 5.2.3 for the adaptive reducibilities. A *polynomial-time bounded Turing reduction of norm k (P-bT(k)-reduction* for short) is a polynomial-time bounded Turing reduction M in which for any oracle set X and any input x the number of oracle queries is bounded by k. The queries may depend on the oracle, i.e., on the answers of the previous queries, whence the computation or query tree of $M(x)$ where the nodes are labelled with the queries has depth $\leq k-1$ but may have size $2^k - 1$.

Obviously,

$$A \leq^{\mathrm{p}}_{btt(k)} B \Rightarrow A \leq^{\mathrm{p}}_{bT(k)} B \tag{5.19}$$

and, by the above remark on the size of the query tree of a *P-bT(k)*-reduction,

$$A \leq^{\mathrm{p}}_{bT(k)} B \Rightarrow A \leq^{P}_{btt(2^k-1)} B, \tag{5.20}$$

and both implications are optimal with respect to the normes of the reductions (see [Lad75]).

In order to distinguish NP-$bT(k+1)$-completeness from NP-$bT(k)$-completeness assuming (G), we show that the set A_k constructed in the proof of Theorem 5.2.3 is NP-$bT(k)$-complete but not NP-$bT(k-1)$-complete. Together with Theorem 5.2.3 this implies the following stronger result.

Theorem 5.2.5. *Assume (G) and let $k \geq 2$. There is a set which is* NP-*$bT(k)$-complete and* NP-*$btt(k+1)$-complete but neither* NP-*$bT(k-1)$-complete nor* NP-*$btt(k)$-complete.*

Proof. Fix A_k, C and G as in the proof of Theorem 5.2.3. It remains to show that A_k is NP-$bT(k)$-hard but not NP-$bT(k-1)$-hard.

For a proof of the former note that, by definition of A_k,

$$\begin{aligned} x \in C \quad &\Leftrightarrow \quad [x \in \widehat{G} \,\&\, x \in \widehat{G} \cap C] \text{ or } [x \notin \widehat{G} \,\&\, x \in \widehat{G} \cup C] \\ &\Leftrightarrow \quad [A_k \cap \{xz_l : 1 \leq l \leq k-1\} \neq \emptyset \,\&\, xz_k \in A_k] \\ &\qquad \text{or} \\ &\qquad [A_k \cap \{xz_l : 1 \leq l \leq k-1\} = \emptyset \,\&\, xz_{k+1} \in A_k] \end{aligned}$$

whence $C \leq^{\mathrm{p}}_{bT(k)} A_k$. By NP-$m$-completeness of C this implies that A_k is NP-$bT(k)$-hard. The proof that A_k is not NP-$bT(k-1)$-hard is similar to the proof that A_k fails to be NP-$btt(k)$-complete. Hence we will only sketch the proof and we will use the notation introduced previously.

Given a P-$bT(k-1)$-reduction M it suffices to show that

$$\widehat{G}_k(0^n) \neq M^{A_k}(0^n) \tag{5.21}$$

for some n. As in the proof of Theorem 5.2.3, given a number n and an initial segment $X \restriction 0^n$ it suffices to define a set $COND^X_n = \{(y_l, i_l) : l \leq m\}$ of forcing conditions such that $0^n \leq y_0 < y_1 < \cdots < y_m$, the sizes of the sets $COND^X_n$ are unifomly bounded by a constant c, the forcing locations $\{y_0, \ldots, y_m\}$ and the forced values i_l are uniformly computable in $O(2^n)$ and $O(2^{|y_l|})$ steps, respectively, and to ensure that the assumption (5.12) will imply (5.21).

Again, to fix the intended oracle answers, we define a function $\alpha^X_n : \Sigma^* \to \{0,1\}$ and we distinguish between short and long input strings. For

w with $|w| < n$, $\alpha_n^X(w)$ is defined by (5.2) and, for $w = vz_p$ with $|w| \geq n$ we let $\alpha_n^X(w) = 1$ iff the index p of w is $k+1$.

Next, by simulating M on input 0^n define the set $QUERY_n$ of the queries asked by M if the previous queries were answered by $\alpha_n^X(w)$, and let $M^{\alpha_n^X}(0^n)$ denote this computation.

Note that $|QUERY_n| \leq k-1$ since M is $(k-1)$-bounded and, as one can easily show,

$$i_n^X = M^{\alpha_n^X}(0^n) \tag{5.22}$$

can be computed in $O(2^{2n})$ steps by definition of α_n^X.

In order to ensure that, assuming (5.12), $M^{A_k}(0^n) = M^{\alpha_n^G}(0^n)$ and $\widehat{G}_k(0^n) = 1 - M^{\alpha_n^X}(0^n)$ we define the set $COND_n^X$ of forcing conditions as in the proof of Theorem 5.2.3 (using the set $QUERY_n$ defined above and i_n^X as in (5.22)). For the proof of correctness it is crucial to note that now $|QUERY_n| \leq k-1$ whence not only in (5.17) but also in (5.16) $r \notin IND(v)$, i.e., the string forced into A_k there is not a member of the set $QUERY_n$. Hence, assuming (5.12), for $w \in QUERY_n$ with $|w| \geq n$, $A_k(w) = \alpha_n^X(w) = 0$ for strings of index $\leq k-1$ is immediate while, for strings of index k or $k+1$, $A_k(w) = \alpha_n^X(w)$ is shown as in the proof of Theorem 5.2.3. $\square$

Corollary 5.2.6. *Assume (G) and let $k \geq 1$. There is an* NP-*bT*$(k+1)$-*complete set which is not* NP-*bT*(k)-*complete.*

Corollary 5.2.7. *Assume (G) and let $k \geq 2$. There is an* NP-*btt*$(k+1)$-*set which is not* NP-*bT*$(k-1)$-*complete.*

While the separation for the bounded Turing reducibilities above is optimal, the corresponding Theorem 5.2.3 for the truth-table reducibilities leaves a small gap for $k = 1$:

Question 1. Assuming (G), is there an NP-*btt*(2)-complete set which is not NP-*btt*(1)-complete?

Also it remains the question whether the comparison of bounded truth-table and bounded Turing completeness in Corollary 3.4 can be further improved to obtain optimal bounds:

Question 2. Assume (G) and let $k \geq 1$. Is there an NP-*btt*$(k+1)$-complete set which is not NP-*bT*(k)-complete?

Under the stronger hypothesis that there is a p-generic set G in NP $\cap$ *co*-NP we can give affirmative answers to these questions. (See Section 5.6.1 below.)

5.3 Adaptive and Nonadaptive Queries

For the comparison of bounded truth-table and bounded Turing completeness in the other direction we can prove a bound which is optimal by (5.20). The goal of this section is to prove the following theorem.

Theorem 5.3.1. *Assume (G) and let $k \geq 2$. There is an* NP-*bT*(k)-*complete set A which is not* NP-*btt*$(2^k - 2)$-*complete.*

The construction of the set A with the desired properties is inspired by the set A_2 as defined in the previous section. (Note that A_2 is a witness for the case $k = 2$ in Theorem 5.3.1.) For larger k, A has to be defined such that a fixed NP-m-complete set C can be recovered from A by a *P-bT(k)*-reduction M with a query tree of maximal size. On the other hand, A should be dominated by a p-generic set G in such a way that any reduction that skips one of the queries (possibly) made by M encounters strings for which the remaining queries do not provide any useful information about C. Again the lack of direct access to negative information on the generic set G requires a quite involved definition of the set A.

Definition 5.3.2. For any string y let

$$I_i(y) = \{z : zi \sqsubseteq y\} \qquad (i = 0, 1)$$

be the set of proper initial segments z of y such that the i-extension zi is still extended by y.

Note that $I_0(y)$ and $I_1(y)$ are disjoint and the union of these sets consists of all proper initial segments of y, whence $|I_0(y)| + |I_1(y)| = |y|$.

Using this notion we define sets $C_\&$ and $C_\vee$ depending on C and G as follows:

$$C_\& = \{\overline{x}y : |y| = k - 2 \,\&\, [x \in C \,\&\, \overline{x}y \in G \,\&\, \forall z \in I_1(y)(\overline{x}z \in G)]\}$$

$$C_\vee = \{\overline{x}y : |y| = k - 2 \,\&\, [x \in C \,\vee\, \overline{x}y \in G \,\vee\, \exists z \in I_0(y)(\overline{x}z \in G)]\}$$

where $\overline{x}y$ denotes the coded pair (x, y) defined by $\overline{x}y = 1^{|x|+1}0xy$. (For the following note that $\overline{x}y \in \Sigma^* \setminus \{0\}^*$ and, for $|y| = k - 2$, $|\overline{x}y| = 2|x| + k$.)

Proposition 5.3.3. *The sets $C_\&$ and $C_\vee$ are dominated by the corresponding set G.*

Proof. Consider a string $w = \overline{x}y$ with $|y| = k - 2$. In case of the set $C_\&$, control about G only allows of forcing $G_\&(w) = 0$. This can be achieved either by letting $G(\overline{x}y) = 0$ or by selecting some member $z \in I_1(y)$ and letting $G(\overline{x}z) = 0$—provided $I_1(y)$ is nonempty.

The analog holds for $C_\vee$. $C_\vee(w) = 1$ can be forced by letting $G_\vee(w) = 1$ or by letting $G(\overline{x}z) = 1$ for some member z of $I_0(y)$. □

It is crucial to note that a single bit of information about $G(\overline{x}z)$ for some prefix z of y may already determine the value $C_{\&}(\overline{x}y)$ or $C_{\vee}(\overline{x}y)$. On the other hand, $C_{\&}(\overline{x}y) = 1$ implies $x \in C$ and $C_{\vee}(\overline{x}y) = 0$ for some y of length $k-2$ implies $x \notin C$. However, the information that $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$ for some y of length $k-2$ helps to determine $C(x)$ only if

$$\forall i \leq 1 \, \forall z \in I_i(y) \; G(\overline{x}z) = i$$

and the value $G(\overline{x}y)$ is known.

Hence, let G and C be sets in NP. Then the desired set A is defined as the disjoint union of $G^- = G \setminus \{0\}^*$, $C_{\&}$ and $C_{\vee}$:

$$A = \{v\underline{1} : \; v \in G^-\} \; \cup \; \{v\underline{2} : \; v \in C_{\&}\} \; \cup \; \{v\underline{3} : \; v \in C_{\vee}\} \tag{5.23}$$

where $\underline{1}$, $\underline{2}$, $\underline{3}$ are the first three strings of length 2.

Proposition 5.3.4. *The set A defined above is in* NP, *and $C \leq^{\mathrm{p}}_{bT(k)} A$.*

Proof. Note that $C_{\&}$ and $C_{\vee}$ are positively P-btt-reducible to $C \oplus G$. Hence $A \in \mathrm{NP}$ by standard closure properties of NP.

We first point out some basic relations among the parts of A and the sets C and G. Note that, for any string y of length $k-2$,

$$C_{\&}(\overline{x}y) \leq G(\overline{x}y) \leq C_{\vee}(\overline{x}y). \tag{5.24}$$

For any string x let y_x be the unique string of length $k-2$ satisfying

$$\forall z \sqsubset y_x \quad (\overline{x}z \in G \Leftrightarrow z1 \sqsubseteq y_x) \tag{5.25}$$

Then $\overline{x}z \in G$ for all $z \in I_1(y_x)$ and $\overline{x}z \notin G$ for all $z \in I_0(y_x)$. I.e., the last conjunct in the definition of $C_{\&}$ is true while the last disjunct in the definition of $C_{\vee}$ is false, whence

$$\overline{x}y_x \in G \Rightarrow C(x) = C_{\&}(\overline{x}y_x) \tag{5.26}$$

and

$$\overline{x}y_x \notin G \Rightarrow C(x) = C_{\vee}(\overline{x}y_x) \tag{5.27}$$

So given x, two adaptive queries to A suffice to determine $C(x)$ if the string y_x is known. Therefore it remains to show that y_x can be found by $k-2$ adaptive queries to A. Obviously, this can be done by fixing $y_x = y_x(0) \dots y_x(k-3)$ where

$$y_x(l) = G(\overline{x}(y_x \restriction l)) = A(\overline{x}(y_x \restriction l)\underline{1})$$

for $l \leq k-3$. □

To demonstrate that A can not be NP-*btt*(2^k-2)-complete, for given x the behaviour of $C_{\&}$ and $C_{\vee}$ for strings $\overline{x}y \neq \overline{x}y_x$ will be discussed. For this purpose we introduce the following notation.

Given strings u and v, call u *to the left of* v ($u <_L v$), if there are strings w_0, w_1, and w_2 such that $u = w_0 0 w_1$ and $v = w_0 1 w_2$. Then, for strings $y \neq y_x$ of length $k-2$ we have

$$\begin{array}{lcl} y <_L y_x & \Rightarrow & C_{\vee}(\overline{x}y) = 1 \\ y_x <_L y & \Rightarrow & C_{\&}(\overline{x}y) = 0 \end{array} \tag{5.28}$$

This follows from the fact that, by definition of y_x, for the longest common initial segment z of y and y_x, $G(\overline{x}z) = 1$ and $z \in I_0(y)$ if $y <_L y_x$ while $G(\overline{x}z) = 0$ and $z \in I_1(y)$ if $y_x <_L y$.

Proposition 5.3.4 implies that A becomes NP-$bT(k)$-complete, if we choose C NP-m-complete. As in the preceding sections, we are able to prove that A is not NP-*btt*(2^k-2)-complete, provided that G is a p-generic set and $C \in \mathrm{DTIME}(2^n)$.

Proposition 5.3.5. *Let G be a p-generic set in* NP, *C a set in* DTIME(2^n) *and $k \geq 2$. Then G is not P-btt(2^k-2)-reducible to the set A as defined in (5.23).*

Proof. Given a p-*btt*(r)-reduction $(h, g_1, \ldots, g_r)$ with $r \leq 2^k-2$ it suffices to show that

$$G(0^n) \neq h(0^n, A(g_1(0^n)), \ldots, A(g_r(0^n))) \tag{5.29}$$

for some n, i.e., $G \not\leq^{\mathrm{p}}_{btt(r)} A$. As in the preceding sections this can be established by defining, for any given number n and any initial segment $X \restriction 0^n$, a set $COND^X_n = \{(y_l, i_l) : l \leq m\}$ of forcing conditions such that $0^n \leq y_0 < y_1 < \cdots < y_m$ and the following properties hold: The size of $COND^X_n$ is uniformly bounded by a constant, there are uniform procedures for computing the set $\{y_l : l \leq m\}$ in $O(2^n)$ steps and for computing i_l from y_l in $O(2^{|y_l|})$ steps, and, finally, the condition (5.12) will imply that (5.29) holds.

Fix n and $X \restriction 0^n$. For $QUERY_n = \{g_1(0^n), \ldots, g_r(0^n)\}$ we define a function $\alpha^X_n : QUERY_n \to \{0,1\}$ such that, assuming (5.12), $A(w) = \alpha^G_n(w)$ will be forced by the definition of $COND^X_n$ and α^X_n will be computable in $O(2^n)$ steps.

For the definition of α^X_n and $COND^X_n$ we have to distinguish the different types of elements of A. A string w is called *relevant* if $w = \overline{x}y\underline{j}$, $j \in \{1,2,3\}$, $|y| \leq k-2$, and $|y| = k-2$ if $j \in \{2,3\}$. And w is called *simple* if $w = v\underline{1}$, $v \notin \{0\}^*$, and w is not relevant. For a relevant string $w = \overline{x}y\underline{j}$, $1 \leq j \leq 3$, j is called the *index* of w and x is called the *value* of w.

Note that any element of A is either relevant or simple. For a relevant string $w = v\underline{1}$ with index 1 or a simple string, $A(w) = G(v)$ whence, assuming (5.12), we can force $A(w) = i$ ($i \in \{0,1\}$) by the condition (v, i) if $|v| \geq n$, i.e., $|w| \geq n + 2$.

For a relevant string $w = \overline{x}y\underline{2}$ with index 2, $A(w) = C_{\&}(\overline{x}y)$, and manipulating G only suffices to force $A(w) = 0$. Similarly, for relevant strings with index 3, by disjunctivity of the dependence only $A(w) = 1$ can be forced. Moreover, forcing these values will require to fix G on certain strings with value x, whence forcing the values of A for strings with the same value has to be coordinated.

Based on these observations we define α_n^X and the parts of $COND_n^X$ forcing A to agree with α_n^G as follows. For any irrelevant, non-simple string $w \in QUERY_n$, $\alpha_n^X(w) = 0$, and, for any simple $w = v\underline{1} \in QUERY_n$ let $\alpha_n^X(w) = (X \restriction 0^n)(v)$ if $|v| < n$ and let $\alpha_n^X(w) = 0$ if $|v| \geq n$. Moreover, in the latter case add the condition $(v, 0)$ to $COND_n^X$. Note that, assuming (5.12), in any of the above cases this ensures that $A(w) = \alpha_n^X(w)$.

For the remaining cases let x be a string such that there is a relevant query w in $QUERY_n$ with value x. Then $\alpha_n^X(w)$ is simultaneously defined for all such strings w and a part $COND_x$ of $COND_n^X$ is specified which, assuming (5.12), guarantees the correctness of $\alpha_n^X(w)$. The definition depends on the length of x.

Case 1: $2|x| + k < n$. Then, for any relevant string $w = \overline{x}y\underline{j}$ with value x, $|\overline{x}y| < n$ whence for $w \in QUERY_n$ the definition of $\alpha_n^X(w)$ can be based on the initial segment $X \restriction 0^n$ as follows:

$$\alpha_n^X(\overline{x}y\underline{j}) = \begin{cases} X(\overline{x}y) & \text{if } \ j = 1 \\ C_{\&}^X(\overline{x}y) & \text{if } \ j = 2 \\ C_{\vee}^X(\overline{x}y) & \text{if } \ j = 3 \end{cases}$$

where $C_{\&}^X$ and $C_{\vee}^X$ are defined as $C_{\&}$ and $C_{\vee}$, respectively, with $X \restriction 0^n$ in place of G. Moreover, we let $COND_x = \emptyset$.

In this case, $A(w) = \alpha_n^X(w)$ by the first part of (5.12) and $\alpha_n^X(w)$ can be computed in $O(2^n)$ steps by $C \in DTIME(2^n)$.

Case 2: $n \leq 2|x| + k \leq n + k$. Then let $COND_x$ consist of the conditions $(\overline{x}y, 0)$ for all strings y with $|y| \leq k - 2$ and $|\overline{x}y] \geq n$ and let $X \restriction 0^{n+k}$ be the extension of $X \restriction 0^n$ with $X(z) = 0$ for all strings z with $0^n \leq z < 0^{n+k}$. For the relevant strings $w \in QUERY_n$ with value x, $\alpha_n^X(w)$ is defined as in Case 1 with $X \restriction 0^{n+k}$ replacing $X \restriction 0^n$.

Note that, by the second part of (5.12), $COND_x$ forces $G(\overline{x}y) = 0$ for all $\overline{x}y$ of length $\geq n$ which are relevant for the value of $A(w)$ for the above queries w. So computing $\alpha_n^X(w)$ requires information on C only for the string x and $|x| \leq n$ whence $\alpha_n^X(w)$ can be computed in $O(2^n)$ steps.

Case 3: $n + k < 2|x| + k$. Then, for any string $v = \overline{x}z$, $|v| \geq n$ whence $G(v) = i$ ($i \in \{0,1\}$) can be forced by adding the condition (v, i) to $COND_x$.

We will force these values in such a way that for all strings y of length $k-2$, $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$, whence $A(w) = 0$ and $A(w) = 1$ for relevant strings w with value x which have index 2 and 3. Hence, we let $\alpha_n^X(v\underline{2}) = 0$ and $\alpha_n^X(v\underline{2}) = 0) = 1$, respectively, for those strings $v = \overline{x}y$ of length $2|x|+k$ which are queried. For the definition of $COND_x$ and α_n^X for the index-1 strings with value x we distinguish the following cases.

Case 3.1: $\exists z\ (|z| \leq k-2 \,\&\, \overline{x}z\underline{1} \notin QUERY_n)$. Then let z_0 be the least such string and let

$$\begin{aligned} COND_x \;=\; & \{(\overline{x}z, 0)\ :\ |z| \leq k-2 \,\&\, z <_L z_0 1\} \\ & \cup\{(\overline{x}z, 1)\ :\ |z| \leq k-2 \,\&\, z_0 0 <_L z\} \\ & \cup\{(\overline{x}z, i)\ :\ i \leq 1 \,\&\, z \in I_i(z_0)\} \\ & \cup\{(\overline{x}z_0, 1 - C(x))\}. \end{aligned}$$

For any relevant string $\overline{x}z\underline{1} \in QUERY_n$ with value x and index 1, let $\alpha_n^X(\overline{x}z\underline{1})$ be the unique number i with $(\overline{x}z, i) \in COND_x$. By (5.12) this will obviously imply that $\alpha_n^X(\overline{x}z\underline{1}) = A(\overline{x}z\underline{1})$. Moreover, by choice of z_0, the last part of $COND_x$ is not used for determining $\alpha_n^X(\overline{x}z\underline{1}) = i$ whence this value can be computed in $poly(n)$ steps. To show that $COND_x$ forces the intended values for strings with index 2 and 3 too, fix y with $|y| = k-2$. It suffices to show that, assuming (5.12), $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$. First observe that, by the third part of $COND_x$, $z_0 \sqsubseteq y_x$ for the unique string y_x satisfying (5.25), and that the first two parts of $COND_x$ imply that

$$(z <_L z_0 \;\Rightarrow\; G(\overline{x}z) = 0) \,\&\, (z_0 <_L z \;\Rightarrow\; G(\overline{x}z) = 1) \tag{5.30}$$

for all strings z with $|z| \leq k-2$. So if $y <_L z_0$ then, by (5.30) and (5.24), $C_{\&}(\overline{x}y) = 0$. On the other hand, $z_0 \sqsubseteq y_x$ implies that $y_L y_x$, whence $C_{\vee}(\overline{x}y) = 1$ by (5.28). For $z_0 <_L y$, $C_{\&}(\overline{x}y) = 0$ and $C_{\vee}(\overline{x}y) = 1$ are shown similarly. This leaves the case where $z_0 \sqsubseteq y$, for which it is crucial to note that $C(x) \neq G(\overline{x}z_0)$ by the fourth part of $COND_x$. So, if $z_0 = y$ then $y = y_x$ and $C(x) \neq G(\overline{x}y_x)$ whence the claim follows from (5.26), (5.27), and (5.24). Otherwise, $z_0 \sqsubset y$, say $p = |y| - |z_0|$. Here the proof depends on the value of $C(x)$. If $C(x) = 0$ then $C_{\&}(\overline{x}y) = 0$ is immediate by (5.24). Moreover, by $z_0 \sqsubseteq y_x$ and, by $C(x) = 0$ and by the definition of $COND_x$, $G(\overline{x}z_0 1^l) = 1$ for all $l \leq p$ whence $y_x = z_0 1^p$ and $G(\overline{x}y_x) = 1$. So, either $y <_L y_x$, whence $C_{\vee}(\overline{x}y) = 1$ by (5.28), or $y = y_x$, whence $C_{\vee}(\overline{x}y) = 1$ by $G(\overline{x}y_x) = 1$ and (5.24). Finally if $C(x) = 1$ then, by similar arguments, $C_{\vee}(\overline{x}y) = 1$ is immediate and $y_x = z_0 0^p$ and $G(\overline{x}y_x) = 0$. So $y_x <_L y$ or $y_x = y$ whence $C_{\&}(\overline{x}y) = 0$ by (5.28) or (5.24), respectively.

Case 3.2: $\forall z\ (|z| \leq k-2 \;\Rightarrow\; \overline{x}z\underline{1} \in QUERY_n)$. Then $QUERY_n$ contains at least $2^{k-1} - 1$ strings with index 1. Since

$$2^{k-1} - 1 + 2^{k-2} + 2^{k-2} = 2^k - 1$$

whereas $|QUERY_n| \leq r \leq 2^k - 2$, for one of the 2^{k-2} strings y of length $k-2$ $\overline{x}y\underline{2} \notin QUERY_n$ or $\overline{x}y\underline{3} \notin QUERY_n$. For the following fix y with $|y| = k-2$ and j with $1 < j \leq 3$ minimal such that $\overline{x}y\underline{j} \notin QUERY_n$. Let

$$\begin{aligned} COND_x &= \{(\overline{x}z, 0) : |z| \leq k-2 \,\&\, z <_L y\} \\ &\quad \cup \{(\overline{x}z, 1) : |z| \leq k-2 \,\&\, y <_L z\} \\ &\quad \cup \{(\overline{x}z, i) : i \leq 1 \,\&\, z \in I_i(y)\} \\ &\quad \cup \{(\overline{x}y, i) : (i = 1 \Leftrightarrow j = 2) \,\&\, (i = 0 \Leftrightarrow j = 3)\} \end{aligned}$$

and, for any relevant string $\overline{x}z\underline{1} \in QUERY_n$ with value x and index 1 let $\alpha_n^X(\overline{x}z\underline{1})$ be the unique number i with $(\overline{x}z, i) \in COND_x$. Then, obviously, $\alpha_n^X(\overline{x}z\underline{1})$ can be computed in $poly(n)$ steps and, by (5.12), $A(\overline{x}z\underline{1}) = \alpha_n^X(\overline{x}z\underline{1})$. It remains to show that, assuming (5.12), $COND_x$ forces the intended values for relevant queries with value x and index 2 or 3. Note that the third part of $COND_x$ ensures that $y = y_x$ whence, by the first two parts, for any string y' of length $k-2$,

$$(y' <_L y_x \Rightarrow G(\overline{x}y') = 0) \,\&\, (y_x <_L y' \Rightarrow G(\overline{x}y') = 1)$$

holds. As in Case 3.1 this implies $A(\overline{x}y'\underline{2}) = C_\&(\overline{x}y') = 0$ and $A(\overline{x}y'\underline{3}) = C_\vee(\overline{x}y') = 1$ for $y' \neq y$ by (5.26), (5.27), and (5.24). Finally, if $y' = y$, then $\overline{x}y\underline{j} \notin QUERY_n$. So, for $j = 2$, it suffices to show that $A(\overline{x}y\underline{3}) = C_\vee(\overline{x}y) = 1$. In this case the fourth part of $COND_x$ consists of the condition $(\overline{x}y, 1)$ whence $G(\overline{x}y) = 1$. So the claim follows from (5.24). Similarly, if $j = 3$, then the final condition of $COND_x$ is $(\overline{x}y, 0)$ whence $G(\overline{x}y) = 0$ and therefore $A(\overline{x}y\underline{2}) = C_\&(\overline{x}y) = 0$ by (5.24). Now the condition set $COND_n^X$ consists of the parts specified above together with the final condition

$$\left(0^n, 1 - h(\alpha_n^X(g_1(0^n)), \ldots, \alpha_n^X(g_r(0^n)))\right).$$

Then, assuming (5.12), $COND_n^X$ forces A to agree with α_n^X on the query set $QUERY_n$ whence, by the final condition, (5.29) holds. The proof of the required bounds on $COND_n^X$ is straightforward by the above remarks. In particular, $|COND_n^X| \leq 2^{2k-1}$ since for any of the $r \leq 2^k - 2$ queries at most $2^{k-1} - 1$ conditions are added to $COND_n^X$. □

This completes the proof of Theorem 5.3.1. Under the assumption (G), there is a p-generic set G in NP. If C is an NP-m-complete set in DTIME(2^n), then for $k \geq 2$ the set A is NP-$bT(k)$-complete by 5.3.4, and A is not NP-$btt(2^k - 2)$-complete by 5.3.5.

5.4 Bounded and Unbounded Number of Queries

In the preceding sections we gave separations of the NP-completeness notions for the bounded query reducibilities of fixed norm (under the assumption (G)). By exploiting the unifomity (in k) of the proof of Theorem 5.2.3, here we separate NP-*btt*-completeness from NP-*tt*-completeness. This requires the following diagonalization lemma.

Lemma 5.4.1. *Let* $\mathbf{C}_n$, $n \geq 0$, *be uniformly recursively presentable classes which are closed under finite variants, let* $D \subseteq \{0\}^* \times \Sigma^*$ *be a recursive set such that*

$$D_{[n]} = \{x : \langle 0^n, x\rangle \in D\} \notin \mathbf{C}_n,$$

and let $f : \mathbb{N} \to \mathbb{N}$ *be a nondecreasing and unbounded recursive function. Then there exists a set* A *and a function* $g : \mathbb{N} \to \mathbb{N}$ *such that*

$$A \notin \bigcup_{n \geq 0} \mathbf{C}_n \tag{5.31}$$

$$\forall\, n\, (A^{=n} = D^{=n}_{[g(n)]}) \tag{5.32}$$

g is polynomial-time computable with respect to the unary representation of numbers. (5.33)

$$\forall n\, (g(n) \leq f(n)) \tag{5.34}$$

Note that (5.32) and (5.33) imply that $A \leq^{\mathrm{p}}_m D$. So Lemma 5.4.1 can be viewed as an infinitary version of Schöning's diagonalization lemma in [Sch82].

Proof. The proof is by a standard delayed diagonalization argument similar to the one in [Sch82]. Let U be a recursive presentation of the classes $\mathbf{C}_n$, $n \geq 0$, i.e. $U \subseteq \mathbb{N} \times \mathbb{N} \times \Sigma^*$ is a recursive set such that

$$\mathbf{C}_i = \{U_i^{[n]} : n \geq 0\} \quad \text{where} \quad U_i^{[n]} = \{x \in \Sigma^* : \langle i, n, x\rangle \in U\}.$$

We define the function $h : \mathbb{N} \to \mathbb{N}$ by

$$h(n) = \mu m > n(\forall i \leq n\, \forall l \leq n\, \exists x\, (|x| \in [n, m)\ \&\ D_{[i]}(x) \neq U_i^{[l]}(x)))$$

Then, for $i \leq n$, the length interval $[n, h(n))$ will contain witnesses for the fact that the i-th diagonal set $D_{[i]}$ does not occur under the first n sets in the class $\mathbf{C}_i$. Since the sets D and U are recursive and the classes $\mathbf{C}_i$ are closed under finite variants h will be total recursive. Therefore we may choose a time-constructible function $r \geq h$ and define the intervals

$$I^r(n) := \{x \in \Sigma^* : r^n(0) \leq |x| < r^{n+1}(0)\}$$

where $r^0(0) = 0$, $r^{n+1}(0) = r(r^n(0))$.

Now choose a polynomial-time computable enumeration α of $\mathbb{N}$ in which every number occurs infinitely often and such that $\alpha(n) < f(n)$ for all numbers n. (E.g., given a polynomial-time computable and invertible pairing function $< \cdot, \cdot >$: $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we can define α by letting $\alpha(\langle n, m\rangle) = m$

if the relation $m \leq f(\langle n,m\rangle)$ can be shown in quadratic time by finding a number $k < \langle n,m\rangle$ with $m \leq f(k)$ and by letting $\alpha(\langle n,m\rangle) = 0$ otherwise.)

Finally we define the desired set A by

$$A := \bigcup_{n\geq 0} I^r(n) \cap D_{[\alpha(n)]}$$

For a proof of (5.31), assume that the claim fails, say $A \in \mathbf{C}_i$, whence $A = U_i^{[k]}$ for some k. Then, by the choice of α, there is an m such that $r^m(0) \geq max\{i,k\}$ and $\alpha(m) = i$. Since $r^{m+1}(0) \geq h(r^m(0))$, by definition of h there exists a string $x \in I^r(m)$ such that $x \in D_{[i]}\Delta U_i^{[k]}$. Since $A(x) = D_{[i]}(x)$ by definition, it follows that $A(x) \neq U_i^{[k]}(x)$ contrary to the assumption.

For a proof of the conditions (5.32) – (5.34), note that, by definition of A, there is a unique function $g : \mathbb{N} \to \mathbb{N}$ such that (5.32) holds. Moreover, since $r(n) > n$, for any number n there is a unique number $s(n) \leq n$ such that $r^{s(n)}(0) \leq n < r^{s(n)+1}(0)$ and, by time constructibility of r, $s(n)$ can be computed in $poly(n)$ steps. Then $g(n) = \alpha(s(n))$, whence (5.33) holds by polynomial time computability of α. Moreover, by choice of α and by weak monotonicity of f, it follows with $s(n) \leq n$ that

$$g(n) = \alpha(s(n)) \leq f(s(n)) \leq f(n)$$

whence (5.34) is satisfied, too. □

Proposition 5.4.2. *A set A is* NP*-btt-complete if and only if A is* NP*-btt(k)-complete for some $k \geq 1$.*

Proof. For a proof of the nontrivial implication assume that A is NP-*btt*-complete and let C be an NP-*m*-complete set. Then $C \leq_{btt}^P A$ whence, by definition, $C \leq_{btt(k)}^P A$ for some k. Hence, for any $B \in$ NP, $B \leq_m^P C \leq_{btt(k)}^P A$ by NP-*m*-hardness of C. It follows that $B \leq_{btt(k)}^P A$ whence A is NP-*btt*(k)-hard. □

Theorem 5.4.3. *Assume (G). There is an* NP*-tt-complete set A which is not* NP*-btt-complete.*

Proof. The set A will be composed of the $btt(k+1)$- but not $btt(k)$-complete sets of Theorem 5.2.3. Let $\widetilde{C}$ be an NP-*m*-complete set in DTIME(2^n) and let $C = \{0^n 1x : n \geq 0 \,\&\, x \in \widetilde{C}\}$ be a padded version of $\widetilde{C}$. Note that C is NP-*m*-complete and $C \in$ DTIME(2^n), too.

For $k \geq 0$ let B_k be the NP-*btt*$(k+3)$-complete set A_{k+2} constructed in the proof of Theorem 5.2.3 from C as above and some fixed p-generic set G. (See Definition 5.2.1 and Lemma 5.2.2) Then $B_k \in$ NP uniformly in k, whence $D = \{\langle 0^k, x\rangle : k \geq 0 \,\&\, x \in B_k\} \in$ NP too. Moreover, for $\mathbf{C}_k = \{B : B$ NP-*btt*(k)-complete$\}$ (for $k \geq 1$ and $\mathbf{C}_0 = \mathbf{C}_1$), the classes

$\mathbf{C}_k$ are uniformly recursively presentable and closed under finite variants, and $D_{[k]} = B_k \notin \mathbf{C}_k$. So, by Lemma 5.4.1, we may fix A and g satisfying (5.31)–(5.34) for the recursive function $f(n) = \mu m(2m + 3 \geq n)$. By (5.32) and (5.33), $A \leq_m^{\mathrm{p}} D$ whence $A \in \mathrm{NP}$. Moreover, by (5.31), A is not NP-*btt*(k)-complete for all $k \geq 1$, whence by Proposition 5.4.2, A is not NP-*btt*-complete.

Finally, for the proof that A is NP-*tt*-hard, it suffices to show that $\widetilde{C} \leq_{tt}^{P} A$. So fix x and let $|x| = n$. Then, for any numbers $k, m \geq 0$, $\widetilde{C}(x) = C(0^m 1x)$ and, by definition of B_k, $C(0^m 1x)$ can be computed from $(B_k)^{=(m+1+n)+k+2}$ with $k + 3$ non-adaptive queries in polynomial time uniformly in x, k and m. On the other hand, by (5.32), $A^{=l} = B_{g(l)}^{=l}$ for all l. Hence, to recover $\widetilde{C}(x)$ from A we need determine numbers m and k such that $g(|0^m 1x| + k + 2) = k$. This is necessary since the way C is coded into the set B_k depends on k. Thus the length $l = m + 1 + n + k + 2$ of the queries made in order to determine $\widetilde{C}(x) = C(0^m 1x)$ has to match the corresponding parameter k, namely $k = g(l)$.

So let $k = g(2n + 3)$ and $m = n - k$. Then $m \geq 0$ since $g(2n + 3) \leq f(2n + 3) = n$ by definition of f and (5.34). Moreover, $|0^m 1x| = 2n + 3$ and $A^{=2n+3} = B_k^{2n+3}$, whence $\widetilde{C}(x) = C(0^m 1x)$ can be recovered from $A^{=2n+3}$ with $g(2n + 3) + 3$ non-adaptive queries. □

5.5 Unbounded Number of Adaptive and Nonadaptive Queries

In this section we want to compare NP-completeness notions induced by reducibilities for which the number of queries is growing in the input length. The proofs presented here are closely related to Theorem 5.2.3 and 5.3.1. However, in contrast to the previous results, meeting an extension function as defined in Lemma 2.2.11 might not suffice to force such an unbounded reduction into a given direction. Therefore, the appropriate tool to diagonalize against unbounded polynomial-time reductions are *general* p-generic sets. On the other hand, Lemma 2.2.11 does not have a real analog in the case of general p-generic sets, whence we will have to build the desired extension functions with a little more caution. Nevertheless, under the hypothesis (GG), analog results as in the previous sections can be obtained for unbounded polynomial time reductions.

Theorem 5.5.1. *Assume (GG). Then there is an* NP*-T-complete set A which is not* NP*-tt-complete.*

Proof. Fix a general p-generic set G in NP and an NP-m-complete set $C \in DTIME(2^n)$. In analogy to Theorem 5.3.1, we define sets $C_\&$ and $C_\vee$ depending on C and G, and a set A of which we will show that it has the desired properties. So let $C_{\&}$ and $C_{\vee}$ be defined by

$$C_{\&} = \{\overline{x}y : |y| = |x| - 2 \,\&\, [x \in C \,\&\, \overline{x}y \in G \,\&\, \forall z \in I_1(y)(\overline{x}z \in G)]\}$$
$$C_{\vee} = \{\overline{x}y : |y| = |x| - 2 \,\&\, [x \in C \,\vee\, \overline{x}y \in G \,\vee\, \exists z \in I_0(y)(\overline{x}z \in G)]\}$$

where $\overline{x}y$ denotes the coded pair (x, y) defined by $\overline{x}y = 1^{|x|+1}0xy$. Note that, for $|y| = |x| - 2$, $|\overline{x}y| = 3|x|$. Let $\underline{1}$, $\underline{2}$, $\underline{3}$, be the first three strings of length 2. Then the set A is defined as in Theorem 5.3.1:

$$A = \{v\underline{1} : v \in G^-\} \cup \{v\underline{2} : v \in C_{\&}\} \cup \{v\underline{3} : v \in C_{\vee}\}$$

where $G^- = G \setminus \{0\}^*$.

Note that, for y_x the unique string — now of length $|x| - 2$ — satisfying (5.25) and y any string of the same length, the basic relations (5.26), (5.27), (5.28), and (5.24) hold again. Hence NP-T-completeness of A is straightforward by the arguments given in the proof of Theorem 5.3.1. I.e., $A \in$ NP by the closure properties of NP and C is $\leq^p_T$-reducible to A, where, on input x, $|x| - 2$ adaptive queries are needed to produce y_x, and two more queries are needed to compute $C(x)$, since

$$C(x) = \begin{cases} C_{\&}(\overline{x}y_x) & \text{if } \overline{x}y_x \in G \\ C_{\vee}(\overline{x}y_x) & \text{if } \overline{x}y_x \notin G \end{cases}$$

by (5.27) and (5.28). So it remains to show that A is not NP-*tt*-hard. This will be achieved by refuting the assumption that $G \leq^P_{tt} A$. Given a *P*-*tt*-reduction M we will show that

$$G(0^n) \neq M^A(0^n) \tag{5.35}$$

for some n. Here, we will establish this by defining, for any given n and any initial segment $X \upharpoonright 0^n$, a set $COND^X_n = \{(y_l, i_l) : l \leq m\}$ of forcing conditions where $0^n \leq y_0 < y_1 < \cdots < y_m$, such that the extension function induced be the sets $COND^X_n$ is computable in polynomial time, and, finally, that the condition (5.12) will imply that (5.35) holds.

So fix n and $X \upharpoonright 0^n$. Let $QUERY_n$ be the set of queries made by M on input 0^n. As before, we first define a function $\alpha^X_n : QUERY_n \to \{0, 1\}$ of intended oracle answers and choose $COND^X_n$ such that (5.12) will ensure that

$$A(w) = \alpha^X_n(w) \tag{5.36}$$

for all $w \in QUERY_n$. Call w *relevant* if $w = \overline{x}y\underline{j}$, $j \in \{1, 2, 3\}$, $|y| \leq |x| - 2$, and $|y| = |x| - 2$ if $j \in \{2, 3\}$. And call w *simple* if $w = v\underline{1}$, $v \notin \{0\}^*$, and w is not relevant. For a relevant string $w = \overline{x}y\underline{j}$, $1 \leq j \leq 3$, j is the *index* of w and x is the *value* of w. For any simple query $w = v\underline{1}$ of length $\geq n + 2$ we add the condition $(v, 0)$ to $COND$. For any irrelevant or simple w, let

$\alpha_n^X(w)$ be as in the proof of Theorem 5.3.1. For the relevant queries w with value x, $\alpha_n^X(w)$ is defined, depending on the length of x, as follows.

Case 1: $2|x| < n$. In this case, $C(x)$ can be computed in $O(2^n)$ steps. So let $Y = (X \upharpoonright 0^n)0^\infty$ be the minimal set that extends $X \upharpoonright 0^n$ and define

$$\alpha_n^X(\overline{x}y\underline{j}) = \begin{cases} Y(\overline{x}y) & \text{if } \ j = 1 \\ C_\&^Y(\overline{x}y) & \text{if } \ j = 2 \\ C_\vee^Y(\overline{x}y) & \text{if } \ j = 3 \end{cases}$$

where $C_\&^Y$ and $C_\vee^Y$ are defined as $C_\&$ and $C_\vee$, respectively, with Y in place of G. The conditions that force A to agree with α_n^X on these queries are gathered in the set $COND_x = \{(\overline{x}y, 0) \ : \ |y| \leq |x| - 2 \,\&\, |\overline{x}y| \geq n\}$.

Case 2: $2|x| \geq n$. Here we can force $A(v\underline{1}) = G(v)$ for any relevant string $v\underline{1}$ with value x. Moreover, w.l.o.g we may assume that there is a relevant string of index 2 or 3 with value x that is not a member of $QUERY_n$. So fix y_0 of length $|x| - 2$ and j_0 with $1 < j_0 \leq 3$ minimal such that $\overline{x}y_0\underline{j_0} \notin QUERY_n$.

For y of length $|x| - 2$ let

$$\alpha_n^X(\overline{x}z\underline{i}) = \begin{cases} 0 & \text{if } i = 2 \\ 1 & \text{if } i = 3 \end{cases}$$

and for z of length $\leq |x| - 2$ let

$$\alpha_n^X(\overline{x}z\underline{1}) = \begin{cases} 0 & \text{if } z <_L y_0 \\ 1 & \text{if } y_0 <_L z \\ i & \text{if } i \leq 1 \text{ and } z \in I_i(y_0) \\ 3 - j_0 & \text{if } z = y_0. \end{cases}$$

Now, for each x such there is a relevant query w with value x, let

$$\begin{aligned} COND_x = {} & \{(w, \alpha_n^X(w)) \ : \ w \in QUERY_n \,\&\, w = \overline{x}z\underline{1}\} \\ & \cup \{(\overline{x}z, i) \ : \ i \leq 1 \,\&\, z \in I_i(y_0)\} \\ & \cup \{(\overline{x}y_0, i) \ : \ i = 3 - j_0\} \\ & \cup \{(\overline{x}y, 0) \ : \ |y| = |x| - 2 \,\&\, y <_L y_0 \,\&\, \overline{x}y\underline{2} \in QUERY_n\} \\ & \cup \{(\overline{x}y, 1) \ : \ |y| = |x| - 2 \,\&\, y_0 <_L y \,\&\, \overline{x}y\underline{3} \in QUERY_n\}. \end{aligned}$$

Obviously, under the assumption (5.12), the first part of $COND_x$ ensures that $A(w) = \alpha_n^X(w)$ for each relevant query of index 1. By the second part, y_0 is the unique string y_x satisfying (5.25). So consider y with $|y| = |x| - 2$ and assume (5.12). For $y \neq y_0$, the last two parts of $COND_x$ then ensure that $C_\&(\overline{x}y) = 0$ and $C_\vee(\overline{x}y) = 1$ by (5.27), (5.28), and (5.24). And for $y = y_0$, the third part of $COND_x$ forces $C_\&(\overline{x}y) = 0$ in case that $j_0 = 3$ and $C_\vee(\overline{x}y) = 1$ in case that $j_0 = 2$ by (5.24). Hence the given set $COND_x$ forces the intended values $A(w)$ for all $w \in QUERY_n$. So let the condition set $COND_n^X$ consist of the parts specified above together with the final condition

$$\left(0^n, 1 - M^{\alpha_n^X}(0^n)\right).$$

Then, assuming (5.12), $COND_n^X$ forces A to agree with α_n^X on the query set $QUERY_n$ whence, by the final condition, (5.35) holds. It remains to show that $COND_n^X$ meets the required bounds. Given n and $X \restriction 0^n$, the function α_n^X can be computed in $O(2^{cn})$ steps for some constant c, whence the same holds for the final condition $(0^n, 1 - M^{\alpha_n^X}(0^n))$. Moreover, for each x such that there is a relevant query with value x in $QUERY_n$,

$$|COND_x| \leq |QUERY_n| + |y_0| + 1.$$

Hence $COND_n^X$ has at most $poly(n)$ members and each is computable in time $O(2^{cn})$. As in the preceding sections, this implies that the extension function induced by $COND_n^X$ is a p-extension function.
This completes the proof of Theorem 5.5.1 □

A relevant feature of the above construction is that the number of possibly relevant queries to the parts of A involving the set C exceeds the total number of queries a *P-tt*-reduction can make on the same input. The next theorems study this aspect more carefully by fixing explicit bounds on the number of queries.

Definition 5.5.2. A reduction M is called *P-T(q(n))*-reduction, if M is a *P-T*-reduction that, given an input of length n, makes at most $q(n)$ many queries. And M is called *P-tt (q(n))*-reduction, if M is a *P-tt*-reduction that, given an input of length n, makes at most $q(n)$ many queries.

The proof of Theorem 5.5.1 can be easily modified to obtain, for given polynomial query counting function $q(n)$, the following.

Theorem 5.5.3. *Assume (GG). For* $r(n) = \lceil \log(q(n) + 1) \rceil + 1$ *there is a P-T(r(n))-complete set for* NP *wich is not P-tt(q(n))-complete for* NP.

Theorem 5.5.4. *Assume (GG). For* $r(n) = q(n) + 1$ *there is a P-T(r(n))-complete set for* NP *wich is not P-T(q(n))-complete for* NP.

Theorem 5.5.5. *Assume (GG). For* $r(n) = 2q(n) + 1$ *there is a P-tt(r(n))-complete set for* NP *wich is not P-tt(q(n))-complete for* NP.

5.6 Filling the Gaps

This final section deals with constructions that semm difficult to adapt for the assumptions (G) and (GG), but offer solutions to questions not covered by the constructions given so far. In the first part the small gap in section 5.2 between *P-btt(2)*- and *P-btt(1)*-completeness is discussed. And in the second part, we study the relation between positive and non-positive completeness notions.

The constructions below depend on the possibility of negative access to the assumed (general) p-generic set within NP. So they are based on the hypotheses

$(G_\cap)$ There is a p-generic set in NP $\cap$ *co*-NP,

and

$(GG_\cap)$ There is a general p-generic set in NP $\cap$ *co*-NP.

5.6.1 Two Nonadaptive Queries

Theorem 5.6.1. *Assume* $(G_\cap)$. *For* $k \geq 1$, *there is a P-btt(k+1)-complete set* A *for* NP *which is not P-bT(k)-complete for* NP.

Proof. Fix $k \geq 1$ and let G be a p-generic set in NP$\cap$*co*-NP. Fix an NP-m-complete set C in DTIME(2^n) and let $z_1, \ldots, z_{k+1}$ be the first $k+1$ strings of length k.

The extractions $\hat{G}_m$ used in the proof of Theorem 5.2.3 again play a central role here. For $1 \leq m \leq k+1$, let

$$\begin{aligned} \hat{G}_m &= \{x : xz_m \in G\} \\ \hat{G} &= \bigcup_{m=1}^{k} \hat{G}_m, \end{aligned}$$

and let

$$A = \bigcup_{m=1}^{k} \{xz_m : x \in C \,\&\, x \in \hat{G}_m\} \cup \{xz_{k+1} : x \in C \,\&\, x \notin \hat{G}\}$$

Then $x \in C$ iff $\exists 1 \leq m \leq k+1 : xz_m \in A$, whence A is NP-*btt*$(k+1)$-hard. Moreover, $A \in$ NP since the complement of G as well as the complement of $\hat{G}$ are sets in NP. So it suffices to show that $\hat{G}_{k+1} \not\leq^{\mathrm{p}}_{bT(k)} A$.

Now the proof follows the general pattern. We assume that $\hat{G}_{k+1}$ is *P-bT(k)*-reducible to A via M, i.e.,

$$\hat{G}_{k+1}(y) = M^A(y) \tag{5.37}$$

for all strings y. Then, given n and $X \upharpoonright 0^n$, let

$$\alpha_n^X(xz_i) = \begin{cases} C(x) \cdot X(xz_i) & \text{if } |xz_i| < n \,\&\, 1 \leq i \leq k \\ C(x) \cdot \prod_{m=1}^{k}(1 - X(xz_m)) & \text{if } |xz_i| < n \,\&\, i = k+1 \\ 0 & \text{otherwise.} \end{cases}$$

For strings w not of the form $w = xz_i$ for some x and $1 \leq i \leq k+1$, let $\alpha_n^X(w) = 0$. Then for each x we have to define a set of conditions $COND_x$ that forces A to agree with α_n^X on strings $w = xy_i$. So let $QUERY_n = Q(M, \alpha_n^X, 0^n)$ be the set of queries made by M on input 0^n if given oracle α_n^X. Then for every string x of length $\geq n-k$ such that there is a query $xz_i \in QUERY_n$ $(1 \leq i \leq k)$, let

$$r = \mu m\{m \geq 1 \,\&\, xz_m \notin QUERY_n\}$$

and

$$\begin{aligned} COND_x = & \bigcup_{m=1}^{k} \{(xz_m, 0) : xz_m \in QUERY_n\} \\ & \cup \{(xz_r, 1) : r \leq k\}. \end{aligned}$$

Finally, let $COND_n^X$ be the union of the parts $COND_x$ above and the final condition

$$(0^n z_{k+1}, 1 - M^{\alpha_n^X}(0^n)). \tag{5.38}$$

Again, under the assumption (5.12), this final condition forces the failure of (5.37), provided that $COND_n^X$ then forces $M^A(0^n) = M^{\alpha_n^X}(0^n)$. So assume (5.12) and consider $q \in QUERY_n$. Obviously, $A(q) = \alpha_n^G(q)$ for q not of the form $q = xz_i$ or $|q| < n$. So fix $q \in QUERY_n$ such that $|q| \geq n$ and $q = xz_i$ for some $1 \leq i \leq k+1$. Then $\alpha_n^X(q) = 0$. In case that $i \leq k$, $COND_x$ forces $G(xz_i) = 0$, whence $A(q) = (C \cap G)(q) = 0$ as well. For $i = k+1$, the fact that $q = xz_{k+1} \in QUERY_n$ implies that the corresponding $r \leq k$. I.e., $xz_r \notin QUERY_n$ and $COND_x$ forces $G(xz_r) = \hat{G}_r(x) = 1$. But then $\hat{G}(x) = 1$, whence $A(xz_{k+1}) = A(q) = 0$.

Finally note that $|COND_n^X| \leq 2k+1$ and the extension function induced by $COND_n^X$ is a bounded p-extension function. □

5.6.2 Positive Reductions

Theorem 5.6.2. *Assume $(G_\cap)$. Then there is a $\leq_{bT(2)}^P$-complete set A for* NP *which is not $\leq_{posT}^P$-complete for* NP.

Proof. Let G be a p-generic set in NP $\cap$ *co*-NP and let C be some NP-$\leq_m^P$-complete set in $DTIME(2^n)$. Let δ be the iterated power of 2, i.e., $\delta(0) = 1$, $\delta(n+1) = 2^{\delta(n)}$, and let x_n denote the string $0^{\delta(n)}$. Finally, let $I_n = \{x : \delta(n) \leq |x| < \delta(n+1)\}$. For all strings x in I_n the way we code $C(x)$ into the desired set A will depend on the value $G(x_n)$, whence A will be NP-$\leq_{bT(2)}^P$-complete. But, by genericity of G, A will not be $\leq_{posT}^P$-complete. To achieve this, we define sets D, D_n, A, and A_n. For all $i \in \{0,1\}$ and

$x \in \Sigma^*$ let

$$D(ix) = \begin{cases} i & \text{if } x \in I_n \,\&\, G(x_n) = i \\ C(x) & \text{if } x \in I_n \,\&\, G(x_n) \neq i \end{cases} \tag{5.39}$$

$$D_n(ix) = \begin{cases} i & \text{if } x \in I_n \\ D(ix) & \text{otherwise} \end{cases} \tag{5.40}$$

and let

$$A = G_0 \oplus D, \tag{5.41}$$

and

$$A_n = (G_0 \setminus \{0^{\delta(n)-1}\}) \oplus D_n \tag{5.42}$$

for $G_0 = \{x : 0x \in G\}$. Note that D and D_n disagree on a string ix only if $x \in I_n$. Morevoer, $D \subset D_n$ only if $G(x_n) = 0$ and $D_n \subset D$ only if $G(x_n) = 1$

Claim 1: A is $\leq^P_{bT(2)}$-complete for NP.

Proof: Since $G \in$ NP $\cap$ *co*-NP, $A \in$ NP by standard closure properties of NP. To see that $C \leq^{\mathrm{p}}_{bT(2)} A$, note that $C(x) = A(1(1 - i_n)x)$ for $x \in I_n$ and $i_n = G(x_n)$, by (5.39). Since $i_n = G(x_n) = A(x_n)$, $C(x)$ can be recovered from A by two adaptive queries. Moreover, on input $x \in I_n$, the corresponding x_n can be computed in polynomial time, whence $C \leq^P_{bT(2)} A$. □

Next we define the set $H \in$ NP and show that it is not $\leq^P_{posT}$-reducible to A. Let

$$H = \{1^{j+1}0x : j = G_1(x)\} \tag{5.43}$$

for $G_1 = \{x : 1x \in G\}$, and assume that $H \leq^P_{posT} A$ via M. Then we define a function $\gamma : \Sigma^* \to \Sigma^2$ by

$$\gamma^X(x) = \langle M^{A_n^x}(10x), M^{A_n^x}(110x)\rangle$$

for n such that $x \in I_n$, and A_n^X defined as A_n with X in place of G. Note that, for x of length $|\delta(n)|$ and n large enough, the influence of the set X on $\gamma^X(x)$ is determined by the values $X(x_0), \ldots, X(x_{n-1})$, and therefore by $X \restriction x_n$. Since M reduces H to A, the pair $\langle M^A(10x), M^A(110x)\rangle$ either equals $\langle 0, 1\rangle$ or $\langle 1, 0\rangle$. By definition of $A_n = A_n^G$, either $A \subseteq A_n$ or $A_n \subseteq A$. Moreover, this relation depends on $G(x_n)$:

$$A \subset A_n \;\Rightarrow\; G(x_n) = 0 \tag{5.44}$$

$$A_n \subset A \;\Rightarrow\; G(x_n) = 1 \tag{5.45}$$

Therefore, by positivity of M, the following implications hold:

$$\begin{aligned}
\gamma^G(x) = \langle 0,0\rangle \;&\Rightarrow\; A_n \subset A \;\Rightarrow\; G(x_n) = 1\\
\gamma^G(x) = \langle 1,1\rangle \;&\Rightarrow\; A \subset A_n \;\Rightarrow\; G(x_n) = 0\\
\gamma^G(x) = \langle 1,0\rangle \;&\Rightarrow\; G_1(x) = G(1x) = 0\\
\gamma^G(x) = \langle 0,1\rangle \;&\Rightarrow\; G_1(x) = G(1x) = 1
\end{aligned}$$

So let $f : \Sigma^* \to \Sigma$ be the extension function, which is defined on the strings $X\restriction x_n$ by

$$f(X\restriction x_n) = \begin{cases} (x_n, 0) & \text{if } \gamma^X(1^{\delta(n)}) = \langle 1,1\rangle \\ (x_n, 1) & \text{if } \gamma^X(1^{\delta(n)}) = \langle 0,0\rangle \\ (1^{\delta(n)+1}, 1) & \text{if } \gamma^X(1^{\delta(n)}) = \langle 1,0\rangle \\ (1^{\delta(n)+1}, 0) & \text{if } \gamma^X(1^{\delta(n)}) = \langle 0,1\rangle \end{cases} \tag{5.46}$$

Claim 2: G does not meet f

Proof: This is obvious by the above remarks about the relation between G and γ^G. □

Claim 3: f is an n^c-extension function for some c.

Proof: By definition of f and γ^X, on input $X\restriction x_n$, f has to simulate the computations of $M^{A_n^X}(101^{|x_n|})$ and $M^{A_n^X}(1101^{|x_n|})$. W.l.o.g. assume that the running time of M on these inputs is bounded by $\delta(n+1)$. Hence, for any query q made by M, the answer $A_n^X(q)$ depends on $X\restriction x_n$ and C.

(i) If $q = 0z$, then $A_n^X(q) = 1$ iff $z = x_j$ for some $j < n$ and $X(0z) = 1$.

(ii) If $q = 10z$, then $A_n^X(q) = 1$ iff $z \in I_j$ for some $j < n$ and $[X(x_j) = 1 \,\&\, C(z) = 1]$.

(iii) And if $q = 11z$, then $A_n^X(q) = 1$ iff either $z \in I_n$, or $z \in I_j$ for some $j < n$, and $[C(z) = 1 \vee X(x_j) = 1]$.

Hence, in dependence of the prefix of q, f can compute $A_n^X(q)$ by scanning the input and by computing $C(z)$ if necessary, in linear time, which shows that f is an n^2-extension function. □

Since G is a general p-generic set, these claims are contradictory, which completes the proof of Theorem 5.6.2. □

Considering the set A given in [BuhHT91] to separate $\le_T^P$- from $\le_{tt}^P$-completeness for EXP, we easily obtain a separation of $\le_{posT}^P$-completeness from $\le_{tt}^P$-completeness for EXP by the set $A \oplus \overline{A}$. Though the construction of A may be carried out in NP, this second step fails for NP since $A \oplus \overline{A}$ may not be a member of NP unless we assume NP $=$ *co*-NP. Therefore, to obtain the desired separation, we give a new set B in NP with the required properties by a more direct construction.

Theorem 5.6.3. *Assume ($GG_\cap$). Then there is a set B which is $\leq_{posT}^P$-complete but not $\leq_{tt}^P$-complete for* NP.

Proof. Let G be a general p-generic set in NP $\cap$ *co*-NP and C as in the preceding proof. B will be constructed in such a way that C is recognized by a positive Turing machine M with oracle B which acts as follows.

On input x of length n, M performs $n+1$ rounds of querying B. In the first n rounds M chooses two queries depending on the answers to the previous queries and accepts (rejects) if both answers are 1 (0). Otherwise, M starts the next round. If the final round is reached, only one query is chosen and M accepts iff the answer is 1.

Since for oracles $A \subseteq B$, $M^A(x) = 1$ implies that the computation of M_B on input x either follows the same path and accepts, or there is a minimal round in the computation of $M^A(x)$, where the answers of the oracles A and B differ. But then the behaviour of M implies that A gives two different answers, while both answers of B are 1, whence $M^B(x) = 1$. Therefore M is indeed positive and the computation tree of M on input x may contain a maximal number of $2(2^{|x|} - 1) + 2^{|x|}$ different queries. This fact provides enough flexibility to diagonalize against polynomial time truth-table reductions.

First we define for every string z the tree $T(z)$ of queries to be chosen by M. Consider a balanced tree of depth $|z|$ where the root is labeled by $\{\langle z, 0\rangle, \langle z, 1\rangle\}$ and if a node is labeled by $\{\langle z, y0\rangle, \langle z, y1\rangle\}$ then its left son is labeled by $\{\langle z, y00\rangle, \langle z, y01\rangle\}$ and its right son by $\{\langle z, y10\rangle, \langle z, 11\rangle\}$. Finally, we remove from each leaf the pair $\langle z, y\rangle$ where the last bit of y is one.

The resulting labeled tree will guide M in the following way: the first round performed by M on input z will consist of querying $\langle z, 0\rangle, \langle z, 1\rangle$, i.e., the strings that are labeled to the root of $T(z)$. As described above, M accepts (rejects) in round m if all queries made in this round are answered by 1 (0). Otherwise, if M has reached the node n_m of $T(z)$ in this round, then M chooses the left (right) son of n_m if only the first (resp. the second) answer is 1 and starts the next round by querying the strings that are labeled to that node.

An easy computation shows that for any set S with $|S| < 2 \cdot 2^{|z|} - 1$, there exists a node n in $T(z)$ such that S and the label of n are disjoint. Let $N_z(S)$ denote the first such node and $P_z(S)$ the path in $T(z)$ that leads to it.

Now we are ready to give the definition of B. Fix a set X. First we inductively assoziate to each string w a formula ϕ_w^X with free variable x by

$$\begin{aligned} \phi_\lambda^X(x) &= TRUE, \\ \phi_{w0}^X(x) &= \phi_w^X(x) \wedge 0^{|x|+1}1xw0 \in X, \\ \phi_{w1}^X(x) &= \phi_w^X(x) \wedge 0^{|x|+1}1xw0 \notin X \wedge 0^{|x|+1}1xw1 \in X. \end{aligned}$$

Then we use these formulas to define sets X_w, B_w^X and C_w^X. Let $z(x,w) = 0^{|x|+1}1xw$, and let

$$\begin{aligned} X_w &= \{x : \phi_w^X(x) \wedge z(x,w0) \notin X \wedge z(x,w1) \notin X\} \\ B_{wi}^X &= \{x : \phi_{wi}^X(x) \vee (x \in C \wedge x \in X_w)\} \quad (i \leq 1) \\ C_{w0}^X &= \{x : \phi_w^X(x) \wedge x \in C\} \end{aligned}$$

Finally, let

$$\begin{aligned} B = \{\langle z,w\rangle &: 1 \leq |w| \leq |z| \ \wedge \ z \in B_w^G\} \cup \\ \cup\{\langle z,w0\rangle &: |w| = |z| \ \wedge \ z \in C_{w0}^G\}. \end{aligned}$$

Consider x, w and X such that $\phi_w^X(x)$ is true. Then at most one of the formulas ϕ_{wi}^X for $i \leq 1$ can hold for x, and $x \in X_w$ iff none of the two holds for x. This implies that

$$B_{w0}^X(x) = B_{w1}^X(x) = 0 \Longrightarrow x \notin C, \tag{5.47}$$

and

$$B_{w0}^X(x) = B_{w1}^X(x) = 1 \Longrightarrow x \in C. \tag{5.48}$$

Moreover, in case that $B_{w0}^X(x) = 1 - B_{w1}^X(x)$, then $\phi_{wi}^X(x)$ holds for the unique $i \leq 1$ such that $B_{wi}^X(x) = 1$.

That B has the desired properties, is shown in the following claims.

Claim 1: $B \in$ NP

Proof: As in the previous theorem, this follows by assumption about G and the closure properties of NP. □

Claim 2: C is $\leq_{posT}^P$-reducible to B via M_T.

Proof: Let x of length n be given. By the definition of ϕ, there is a maximal string w of lenth $\leq n$ such that $\phi_w^G(x)$w holds. Then $\phi_{w'i}^G(x) = 1$ iff $w'i \sqsubseteq w$ and $G_{w'}(x) = 0$ for all $w' \sqsubset w$ and $i \leq 1$. This implies that for every $w' \sqsubset w$ and $i \leq 1$,

$$B_{w'i}^G(x) = 1 \iff w'i \sqsubseteq w.$$

Hence $M_T^B(x)$ performs $|w|$ rounds of querying B and proceeds either by querying $\langle x,w0\rangle$ and $\langle x,w1\rangle$, or by querying $\langle x,w0\rangle$ in case that $|w| = n$. In the first case, by maximality of w

$$B(\langle x,w0\rangle) = B(\langle x,w1\rangle) = C(x) = M_T^B(x)$$

by (5.47) and (5.48). In the second case,

$$B(\langle x,w0\rangle) = C_{w0}^G(x) = C(x) = M_T^B(x)$$

since $\phi_w^G(x)$ holds. □

Claim 3: B is dominated by G

Proof: Consider a pair $\langle x, w\rangle$ such that $1 \leq |w| \leq |x|$. By definition of B_w^G, $B(\langle x, w\rangle) = 1$ whnenever the following holds:

$$\begin{aligned}
&\forall w' \in I_0(w) \; : \; z(x, w'0) \in G \\
&\forall w' \in I_1(w) \; : \; z(x, w'0) \notin G \,\&\, z(x, w'1) \in G \\
&z(x, w) \in G \\
&\exists w' : w = w'1 \; : \Longrightarrow z(x, w'0) \notin G.
\end{aligned}$$

On the other hand, $B(\langle x, w\rangle) = 0$ whenever $z(x, w) \in G$, but one of the above conditions fails.

For pairs $\langle x, w0\rangle$ where $|w| = |x|$, only $B(\langle x, w0\rangle) = 0$ can be forced. Namely, whenever one of the above conditions fails, this implies that $\phi_w^G(x)$ does not hold, whence $B(\langle x, w0\rangle) = C_{w0}^G(x) = 0$. □

Claim 4: B is not NP-*tt*-complete.

Proof: Let $G_1 = \{x \, : \, 1x \in G\}$, and assume that G_1 is P-*tt*-reducible to B via M. Fix n and $X \restriction 0^n$, and let $QUERY_n = Q(M, 0^n)$ be the set of queries made by M on input 0^n. Again, we have to fix the function α_n^X of intended oracle answers and the condition set $COND_n^X$ such that (5.12) implies

$$M^B(0^n) \neq G_1(0^n) \tag{5.49}$$

for some n, and such that the induced extension function is a p-extension function.

So consider $p \in QUERY_n$. Call p *relevant* if it is of the form $\langle q, w\rangle$ for some string w of length $1 \leq |w| \leq |q| + 1$, and the last bit of w is zero for $|w| = |x| + 1$. Fix $\alpha_n^X(p) = 0$ for any irrelevant query p. For relevant queries $p = \langle q, w\rangle$ such that $2|q| + 3 < n$, let

$$\alpha_n^X(\langle q, w\rangle) = \begin{cases} B_w^Y(q) & \text{if } 1 \leq |w| \leq |q| \wedge |z(q, w)| < n \\ C_w^Y(q) & \text{if } w = w'0 \wedge |w'| = |q| \wedge |z(q, w)| < n \\ 0 & \text{otherwise} \end{cases}$$

where Y denotes the minimal set extending $X \restriction 0^n$, i.e., $Y \restriction 0^n = X \restriction 0^n$ and $Y(z) = 0$ for all $z \geq 0^n$. For every q such that there is a relevant query $\langle q, w\rangle$ in $QUERY_n$, let

$$COND_q = \{(\langle q, w\rangle, 0) \; : \; 1 \leq |w| \leq |q| \wedge 2|q| + 2 + |w| = n\}$$

In case that $2|q| + 3 \geq n$, w.l.o.g. we may choose w_q of length $|q|$ minimal such that $\langle q, w_q 0\rangle \notin Q(M, 0^n)$. For each such q we want to fix α_n^X such that

$M_T^{\alpha_n^X}(q)$ would perform the maximal number $|q|+1$ rounds, finally querying $\langle q, w_q\rangle$. So let

$$\alpha_n^X(\langle q, w\rangle) = \begin{cases} 1 & \text{if } 1 \le |w| \wedge w \sqsubset w_q \\ 0 & \text{otherwise.} \end{cases}$$

To force these intended oracle answers, fix

$$\begin{aligned} COND_q = & \{(z(q, wi), 1) \,:\, w \in I_i(w_q)\} \\ & \cup\{(z(q, wi), 0) \,:\, w \in I_{1-i}(w_q)\} \\ & \cup\{(z(q, w_q), 1)\} \\ & \cup\{(z(q, w0), 0) \,:\, w_q = w1\} \end{aligned}$$

Finally, let $COND_n^X$ consist of the parts $COND_q$ and the final condition

$$(10^n, 1 - M^{\alpha_n^X}(0^n)).$$

Then the extension function induced by $COND_n^X$ on $\{0^n \,:\, n \ge 1\}$ is an n^k-extension function for some $k \ge 1$, whence G meets f. It remains to show that, if G meets f at 0^n, then B agrees with α_n^G on the set $QUERY_n$. By the definition of α_n^X, this is obvious for irrelevant queries and for queries $\langle q, w\rangle$ such that $2|q| + 2 + |w| < n$. So consider $p = \langle q, w\rangle \in QUERY_n$ such that $3|q| + 2 = |z(q, q)| \ge n$ and assume (5.12).
Case 1: $2|q| + 3 < n$ and $1 \le |w| \le |q|$.

Then $\alpha_n^X(p) = 0$, and there is a prefix w' of w such that $|z(q, w'0)| = n$. Thus (5.12) and the definition of $COND_q$ imply that neither $\phi_{w'}^G(q)$ nor $\phi_w^G(q)$ hold, whence $B_w^G(q) = A(p) = 0$.
Case 2: $2|q|+3 < n$ and $|w| = |q|+1$. In this case $w = w'0$, and the analog argument as in Case 1 for w' shows that $\phi_{w'}^G(q)$ does not hold, whence $A(p) = C_{w'0}^G = 0 = \alpha_n^X(p)$. □

Case 3: $2|q| + 3 \ge n$ and $1 \le |w| \le |q|$.

If $w <_L w_q$ then there is a maximal common prefix z of w and w_q. Then neither $\phi_{z0}^G(q)$ holds, nor $q \in G_{z0}$ be the first two parts of $COND_q$. Hence $B_w^G(q) = \alpha_n^G(p) = 0$. And a similar argument applies for the case $w_q <_L w$. So consider $w \sqsubseteq w_q$. Then again by the first two parts of $COND_q$ $\phi_w^G(q)$ holds, whence $B_w^G(q) = \alpha_n^G(p) = 1$.
Case 4: $2|q| + 3 \ge n$ and $|w| = |q| + 1$.

Then $\alpha_n^G(p) = 0$. By the choice of w_q, $w \ne w_q0$. Note that for w' with $|w'| = |q|$, $\phi_{w'}^G(q)$ holds only if $w' = w_q$ by the last two parts of $COND_q$. Hence $C_w^G = A(p) = 0$.
This completes the proof of Theorem 5.6.3. □

Chapter 6

Summary

The true localization of the class NP within the exponential time hierarchy is probably the most famous open problem of structural complexity theory. Complexity classes like E or EXP, which contain a universal function for P, allow for diagonalizations against polynomial time-bounded reducibilities. This results in a very rich internal structure under such reducibilities.

Interesting structural properties claimed for the class NP, however, immediately would settle the P = NP question to the negative. So, for the time being, the investigation of the internal structure of NP has to be based on assumptions about the localization of NP within the exponential time hierarchy.

Lutz [Lut90] proposed to study the internal structure of NP under the hypothesis that NP does not have p-measure 0 and under related measure assumptions like NP not having measure 0 in E. There are many properties which can be shown under these assumptions but which so far could not be obtained from the weaker assumption that $\mathrm{P} \neq \mathrm{NP}$ (see Lutz and Mayordomo [LutM96]).

Ambos-Spies [Amb94a] proposed to consider not only Lutz's measure axiom for NP but also weaker hypotheses. Especially category based assumptions, that are more directly related to diagonalizations than the measure theoretic approach of Lutz, are mentioned there: the assumption that NP is not p-meager, and the variant that NP is assumed to be not general p-meager in the sense given in [Amb96]. All these assumptions are *consistent* in the sense that they hold for some relativizations. In fact Kautz and Miltersen [KauM94] have shown that the class of oracles A such that $\mu_{p^A}(\mathrm{NP}^A) \neq 0$ has measure 1. Though there is no evidence that this relativization to a random oracle describes the unrelativized case, working with strong hypotheses are of interest by pointing out the relations between possible structural properties of NP. So far, the plausibility of such a hypothesis can only be judged by its consequences. It is easy to see that the Genericity Hypothesis suffices to distinguish many of the standard polynomial time reducibilities on NP: So $\leq_1^p$, $\leq_m^p$, $\leq_{btt(k)}^p$ for $k \geq 2$, $\leq_{btt}^p$ and $\leq_{tt}^p$ are mutually different on NP. (The relations between $\leq_m^p$ and $\leq_{btt(1)}^p$ and between $\leq_{tt}^p$ and $\leq_T^p$ on NP under the hypothesis (G) are not known.)

One of the most beautiful implications of Lutz's measure axiom is the separation of p-m-completeness and p-T-completeness (Lutz and Mayordomo [LutM96]). As Ambos-Spies [Amb94b] has shown, in fact the Genericity Hypothesis is sufficient for this separation.

This thesis extends the ideas of these two results to other NP-completeness notions. Completeness notions corresponding to reductions of bounded norm are studied under the hypothesis (G). It turned out that this hypothesis suffices to establish most of the natural instances of separation results known for the class EXP. Some of the remaining gaps can be filled under the hypotheses ($G_\cap$) and ($GG_\cap$) . The analogous scenery emerges for completeness notions corresponding to reductions of unbounded norm under the hypothesis (GG). However, the differences in the bounds on the number of queries are not as tight as in the bounded case. Nevertheless, it turns out that the hypothesis (GG) allows for separations that are not limited to reductions with slowly growing bounds on the number of queries, the way hypothesis (R) seems to be. Finally, the relation between completeness notions induced by positiv and nonpositive reductions is studied, but again, the results that are possible for the class EXP seem to require the strong genericity hypotheses ($G_\cap$) and ($GG_\cap$) to allow negative access to the assumed (general) p-generic set.

So far, there is no generally accepted attitude towards the hypotheses considered in this thesis. On one hand, it is argued in [LutM96] that a martingale witnessing that NP *does have* p-measure zero, i.e., a betting strategy that is computable in polynomial time and succeeds in every language in NP would be a remarkable algorithm. (In the sense that the strategy would have to make a bet without having the possibility of totally examining the "field of possible solutions".) On the other hand, it is hard to imagine an NP-algorithm that would lead to a p-random (or p-generic) set. One might also doubt the existence of a DTIME(2^n)-bi-immune, or even a p-bi-immune set in NP. This thesis does not discuss this question, but makes contributions to the list of consequences of this kind of strong hypotheses.

Bibliography

[Amb94a] K. Ambos-Spies. Largeness axioms for NP. Lecture at the *Annual Meeting of the Special Interest Group* Logik in der Informatik of the Gesellschaft fuer Informatik (GI), May 1994.

[Amb94b] K. Ambos-Spies. Some notes on genericity and measure for exponential time. unpublished manuscript, 1994.

[Amb96] K. Ambos-Spies. Resource-bounded genericity. In *Computability, Enumerability, Unsolvability: Directions in Recursion Theory*, Cooper et al., Eds., London Math. Soc. Lect. Notes Series 224, pages 1–59. Cambridge University Press, 1996.

[AmbB97] K. Ambos-Spies and L. Bentzien. Separating NP-completeness notions under strong hypotheses. In *Proc. 12th Conference on Computational Complexity*, pages 121–127. IEEE Computer Society Press, 1997.

[AmbFH87] K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizations over polynomial time computable sets. *Theoretical Computer Science* **51**, 177–204, 1987.

[AmbFH88] K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizing over deterministic polynomial time. In *Proceedings of the First Workshop on Computer Science Logic, CSL '87*, Lecture Notes in Computer Science 329, pages 1–16. Springer Verlag, 1988.

[AmbMW96] K. Ambos-Spies, E. Mayordomo, Y. Wang, and X. Zheng. Resource-bounded balanced genericity, stochasticity, and weak randomness. In *Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science, STACS 96*, Lecture Notes in Computer Science 1046, pages 63–74. Springer Verlag, 1996.

[AmbNT96] K. Ambos-Spies, H.-C. Neis, and S. A. Terwijn. Genericity and measure for exponential time. *Theoretical Computer Science* **168**, 3–19, 1996.

[AmbTZ97] K. Ambos-Spies, S. A. Terwijn, and X. Zheng. Resource bounded randomness and weakly complete problems. *Theoretical Computer Science* **172**, 195–207, 1997.

[BakGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P =? NP question. *SIAM Journal on Computing* **4**, 431–442, 1975.

[BalDG88] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer Verlag, 1988.

[BalS85] J. L. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory* **18**, 1–10, 1985.

[BelG94] M. Bellare and S. Goldwasser. The complexity of decision versus search. *SIAM Journal on Computing* **23**, 97–119, 1994.

[Ber76] L. Berman. On the structure of complete sets: Almost everywhere complexity and infinitely often speedup. In *Proceedings of the Seventeenth Annual Symposium on Foundations of Computer Science*, pages 76–80. IEEE Computer Society Press, 1976.

[BovC94] D. P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity.* Prentice Hall, 1994.

[Buh93] H. Buhrman. *Resource Bounded Reductions.* PhD thesis, Universiteit van Amsterdam, Amsterdam, 1993.

[BuhHT91] H. Buhrmann, S. Homer, and L. Torenvliet. Completeness for nondeterministic complexity classes. *Mathematical Systems Theory* **24**, 179–200, 1991.

[BuhT94] H. Buhrmann and L. Torenvliet. On the structure of complete sets. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, pages 118–133. IEEE Computer Society Press, 1994.

[Chu33] A. Church. A set of postulates for the foundation of logic. *Annals of Mathematics* **25**, 839–864, 1933.

[Chu36] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics* **58**, 345–363, 1936.

[Coo71] S. COOK. The complexity of theorem proving procedures. In *Proc. 3rd. ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[Edm65] J. EDMONDS. Paths, trees, and flowers. *Canada Journal of Mathematics* **17**, 449–467, 1965.

[Fel86] W. FELLER. *Introduction to Probability Theory and its Applications.* Wiley & Sons, 1986.

[Fen91] S. A. FENNER. Notions of resource-bounded category and genericity. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 196–212. IEEE Computer Society Press, 1991.

[Göd31] K. GÖDEL. On formally undecidable propositions of principia mathematica and related systems. *Monatshefte für Mathematik und Physik* **38**, 173–198, 1931.

[HomKR93] S. HOMER, S. KURTZ, AND J. ROYER. A note on 1-truth-table complete sets. *Theoretical Computer Science* **115**, 383–389, 1993.

[HopU79] J. E. HOPCROFT AND J. D. ULLMAN. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, 1979.

[Imm88] N. IMMERMAN. Nondeterministic space is closed under complementation. *SIAM Journal on Computing* **17**, 935–938, 1988.

[JueL95] D. W. JUEDES AND J. H. LUTZ. The complexity and distribution of hard problems. *SIAM Journal on Computing* **24**, 279–295, 1995.

[Kar72] R. M. KARP. Reducibilities among combinatorial problems. In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., pages 85–103. Plenum, 1972.

[KauM94] S. M. KAUTZ AND P. B. MILTERSEN. Relative to a random oracle, NP is not small. In *Proceedings of the 9th Annual Structure in Complexity Theory Conference*, pages 162–174. IEEE Computer Society Press, 1994.

[Kle36] S. C. KLEENE. Gneral recursive functions of natural numbers. *Mathematische Annalen* **112**, 727–742, 1936.

[KoM81] K. KO AND D. MOORE. Completeness, approximation and density. *SIAM Journal on Ccomputing* **10**, 787–796, 1981.

[Lad75] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM* **22**, 155–171, 1975.

[LadLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial-time reducibilities. *Theoretical Computer Science* **1**, 103–123, 1975.

[Lev73] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission* **9**, 265–266, 1973.

[Lut90] J. H. Lutz. Category and measure in complexity classes. *SIAM Journal on Computing* **19**, 1100–1131, 1990.

[Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences* **44**, 220–258, 1992.

[Lut97] J. H. Lutz. The quantitative structure of exponential time. In *Complexity Theory Retrospective II*, L. A. Hemaspaandra and A. L. Selman, Eds. Springer Verlag, 1997.

[LutM96] J. H. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science* **164**, 141–163, 1996.

[LutZ97] J. Lutz and Z. Zhao. The density of weakly complete problems under adaptive reductions. In *Proc. 12th Conference on Computational Complexity*, pages 111–120. IEEE Computer Society Press, 1997.

[Mah82] S. R. Mahaney. Sparse complete sets for NP: Solution for a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences* **25**, 130–143, 1982.

[Mat70] Y. Matijasevič. Enumerable sets are diophantine. *Soviet Math. Dokl.* **11**, 354–357, 1970.

[May94a] E. Mayordomo. Almost every set in exponential time is P-bi-immune. *Theoretical Computer Science* **136**, 487–506, 1994.

[May94b] E. Mayordomo. *Contributions to the Study of Resource-Bounded Measure.* PhD thesis, Universitat Polytècnica de Catalunya, Barcelona, Spain, 1994.

[Oxt80] J. C. Oxtoby. *Measure and Category.* Springer Verlag, 2 edition, 1980.

[Pap94] C. M. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Pos36] E. L. POST. Finite combinatory processes—formulation I. *Journal of Symbolic Logic* **1**, 103–105, 1936.

[Sch71a] C. P. SCHNORR. *A Unified Approach to the Definition of Random Sequences.* Lecture Notes in Mathematics 218. Springer Verlag, 1971.

[Sch71b] C. P. SCHNORR. Zufälligkeit und Wahrscheinlichkeit. *Mathematical Systems Theory* **5**, 246–258, 1971.

[Sch82] U. SCHÖNING. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science* **18**, 95–103, 1982.

[Sel79] A. L. SELMAN. P-selective sets, tally languages, and the behaviour of polynomial time reducibilities on NP. *Mathematical Systems Theory* **13**, 55–65, 1979.

[Sze88] R. SZELEPCSÉNYI. The method of forced enumeration for nondeterministic automata. *Acta Informatica* **26**, 279–284, 1988.

[Tur36] A. TURING. On computable numbers with an application to the Entscheidungsproblem. *Proc. London Mathematical Society* **2**, 230–265, 1936.

[Tur37] A. TURING. Rectification to On Computable Numbers .... *Proc. London Mathematical Society* **4**, 544–546, 1937.

[Vil39] J. VILLE. *Etude Critique de la Notion de Collectif.* Gauthiers-Villars, 1939.

[WagW86] K. WAGNER AND G. WECHSUNG. *Computational Complexity.* VEB Deutscher Verlag der Wissenschaften, 1986.

[Wan96] Y. WANG. *Randomness and Complexity.* PhD thesis, Universität Heidelberg, 1996.

[Wat87] O. WATANABE. A comparison of polynomial time completeness notions. *Theoretical Computer Science* **54**, 249–265, 1987.