Benedikt **Löwe**, Boris **Piwinger**, Thoralf **Räsch** (*eds.*)
Foundations of the Formal Sciences III
Complexity in Mathematics and Computer Science
Papers of the conference held in Vienna, September 21-24, 2001

Complexity hierarchies derived from reduction functions

Benedikt Löwe

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands

E-mail: `bloewe@science.uva.nl`

Abstract. This paper is an introduction to the entire volume: the notions of reduction functions and their derived complexity classes are introduced abstractly and connected to the areas covered by this volume.

1 Introduction

Logic is famous for what Hofstadter calls *limitative theorems*: Gödel's incompleteness theorems, Tarski's result on the undefinability of truth, and Turing's proof of the non-computability of the halting problem. For each of these limitative results we have three ingredients:

1. an informal notion to be investigated (*e.g.*, provability, expressibility, computability),
2. a formalization of that notion (*e.g.*, provability as formalized in Peano arithmetic PA, definability in a formal language, the Turing computable functions),

Received: February 26th, 2003;
In revised version: November 5th, 2003;
Accepted by the editors: November 21st, 2003.

2000 *Mathematics Subject Classification.* **03–02 03D55 03D15 68Q15** 03E15 03D28 68Q17.

3. and finally a theorem that there is some limitation to the formalized version of the informal notion (*e.g.*, the theorems of Gödel, Tarski and Turing).

The limitative theorems split the world into *cis* and *trans* relative to a barrier and show that some objects are beyond the barrier (*trans*). In our examples, the statement "PA is consistent" $\mathsf{Cons}(\mathsf{PA})$ is beyond the barrier marked by the formal notion of provability in PA, a truth predicate satisfying the convention (T) is beyond the barrier of definability, and the halting problem is "*beyond the Turing limit*" (Siegelmann). Even more importantly, these results relativize: the notion of formal provability in the stronger system $\mathsf{PA} + \mathsf{Cons}(\mathsf{PA})$ gives us another barrier, but for this barrier $\mathsf{Cons}(\mathsf{PA})$ lies *cis*, and $\mathsf{Cons}(\mathsf{PA} + \mathsf{Cons}(\mathsf{PA}))$ lies *trans*. This calls for iteration, and instead of seeing limitative theorems mainly as obstacles, we can see their barriers positively, as a means of defining relative complexity hierarchies.[1] This is the approach we chose for this introductory paper.

Let us illustrate the problems and methodology of limitative results in some examples:

> **(Example 1)** *Is there an effective algorithm to determine whether a number is prime?*

The informal notion involved in this question is the notion of "effective algorithm". In order to give a positive answer to the question in **(Example 1)**, you do not necessarily need a metatheoretical apparatus or even a proper formal definition of what an effective algorithm is. You have to display an algorithm, and convince people that it is effective.

But this pragmatic approach is not possible if you want to answer such a question in the negative. Consider the following statement:

> **(Example 2)** *There is no effective algorithm that determines whether a given graph has an independent subset of size k.*[2]

[1] In a much more general approach, Wolfram Hogrebe writes in his preface to the proceedings volume of the *XIX. Deutscher Kongreß für Philosophie* that was held in Bonn in September 2002 with the general topic *Grenzen und Grenzüberschreitungen*:

"Die limitativen Aspekte der *Condition Humaine* ... [sind] nie nur Beschränkungen, sondern immer auch Bedingungen der Konturfähigkeit im Ausdrucksraum." [Hog$_1$02]

[2] An independent subset of a graph is a set A such that for $a, b \in A$, there is no edge between a and b.

The statement of **(Example 2)** is a universally quantified statement, so in order to prove it, you have to show for each effective algorithm that it doesn't do the job. This requires a formalization of the notion of an "effective algorithm", the development of a metatheory of algorithms.

The usual formalization of "effective algorithm" in computer science is "polynomial time algorithm".[3] Using that formalization, Agrawal, Saxena, and Kayal have answered the question in **(Example 1)** positively in their already famous [AgrKaySax∞]: they give an $O(\log^{7.5} n)$-time algorithm for determining whether a given number is prime. Looking at **(Example 2)**, the question of whether a given graph has an independent subset of size k is $\mathbf{NP}$-complete; *cf.* [Pap94, Theorem 9.4]. Consequently, the validity of the statement of **(Example 2)** is equivalent to $\mathbf{P} \neq \mathbf{NP}$. In light of this, a proof of $\mathbf{P} \neq \mathbf{NP}$ would be another limitative theorem, locating all problems solvable in polynomial time *cis* and the $\mathbf{NP}$-complete problems like TSP, SAT and the problem mentioned in **Example 2** *trans* of the barrier determined by our formalization of "effective algorithm".[4]

Let us move from the examples from computer science to an example from pure mathematics: In classical geometry, you are interested in constructions with ruler and compass: what lengths can be constructed from a given unit length by drawing auxiliary lines with ruler and compass. *E.g.*, can we give a geometric construction of $\sqrt{2}$, of $\sqrt[3]{2}$, of π?

Again, you can prove existential statements of the form "There is a ruler-and-compass construction of X" by just displaying the construction, *e.g.*, for $\sqrt{2}$.

On the other hand, there is no way to prove a statement of the form "X cannot be constructed with ruler and compass" without a means of proving things about arbitrary ruler-and-compass constructions, *i.e.*, of talking about ruler-and-compass constructions as mathematical entities. In fact, you can do this, and prove non-constructibility (as is taught in algebra courses) by interpreting ruler-and-compass constructions as a certain class of field extensions. This is (a small fragment of) what is called *Galois theory*, and it would deserve the name *metageometry*.

[3] *Cf.* the introduction of Engebretsen's paper in this volume: "It has been widely accepted that a running time that can be bounded by a function that is a polynomial in the input length is a robust definition of 'reasonable running time' (Engebretsen, p. 78)".

[4] Here, TSP is the **Traveling Salesman Problem** [Pap94, § 1.3] and SAT is the **Satisfiability Problem** [Pap94, § 4.2]; *cf.* also Section 6.

For classical construction problems like squaring the circle, doubling the cube (the Delic problem), triangulation of arbitrary angles, and construction methods for the regular p-gon, humankind had been seeking unsuccessfully for positive answers (*i.e.*, proofs of the existential formula) for centuries. Mathematicians embedded ruler-and-compass construction into the richer theory of field extensions by associating to a given real r a field extension and proving that r being ruler-and-compass constructible is equivalent to a property of the associated field extension.[5]

Thus, the universally quantified negative forms of the classical problems mentioned above are transformed into simple statements about algebraic objects: the cube cannot be doubled by ruler and compass because $\sqrt[3]{2}$ has degree 3 over $\mathbb{Q}$; the angle of 60° cannot be triangulated, since the polynomial $X^3 - \frac{3}{4}X - \frac{1}{8}$ is the minimal polynomial of $\cos(20^\circ)$ over $\mathbb{Q}$; the regular 7-gon can not be constructed since the 7th cyclotomic polynomial has degree 6 over $\mathbb{Q}$ and is the minimal polynomial of a 7th root of unity.

In the cases discussed, the limitative results relativize: similar to the relativization of Gödel's incompleteness theorem for Peano Arithmetic PA, we get notions of relative computability and relative ruler-and-compass constructibility. These give rise to problems that are not even computable if we have the halting problem as an oracle, or numbers that are not ruler-and-compass constructible even if the ruler has $\sqrt[3]{2}$ marked on it. As soon as a limitative theorem has told us that there are things on the other side of the barrier, we are not content anymore with the simple dichotomy of *cis* and *trans*. We want to compare those that are beyond the barrier according to their complexity. This leads to questions of the following type:

> **(Example 3)** *Is it harder to determine whether a Π_2 expression is satisfiable than to determine whether a quantifier free expression is satisfiable?*

The first problem is known as Schönfinkel-Bernays SAT, the latter is SAT itself. Both are **NP**-hard by Cook's Theorem [Pap94, Theorem

[5] *Cf., e.g.*, [Lan93, Chapter VI].

8.2]. Are they equally hard, or is the first one harder than the second one?[6]

In order to talk about questions like this, we need a relation "is harder than" or "is at least as hard as" and a corresponding complexity hierarchy. In this paper, we shall restrict our attention to a special class of complexity hierarchies, *viz.* those induced by reduction functions. This choice is motivated by the fact that the hierarchies investigated in computer science are of this type, and some of the most famous hierarchies in mathematical logic (*e.g.*, the Wadge hierarchy, one-reducibility and many-one-reducibility) are as well. We shall introduce a notion of complexity hierarchy in an abstract way in Section 2 and then specialize in the sections to follow.

Before starting with our abstract account, we should add a disclaimer: this paper is not a survey of notions of complexity and definitely not a history of complexity. Its purpose is to describe a general feature of several complexity notions that occur in mathematics and computer science and use that to tie together the papers in this volume. We shall give some pointers to the literature for the interested reader but there is no intention to be comprehensive.

2 Abstract notions

Let K be some basic set of objects. This can be the set of graph structures on a given set, of natural numbers, of real numbers etc. Suppose that we fixed some set F of functions from K to K that is closed under compositions (*i.e.*, if f and g are from F, then $f \circ g$ is from F as well) and contains the identity function. We will call these functions **reductions**. Depending on the context, we can choose different sets F.

We can now use F to define a partial preorder (*i.e.*, a reflexive and transitive relation) on the power set of K, $\wp(K)$:

$$A \leq_F B :\iff \exists f \in F(A = f^{-1}[B]).$$

The notion of a reduction and the derived partial preorder occur in many areas of mathematics and computer science: the "efficient reduc-

[6] Schönfinkel-Bernays SAT is $\mathbf{NEXP}$-complete [Pap94, Theorem 20.3] and SAT is in $\mathbf{NP}$. Hence, the answer to the question in **(Example 3)** is 'Yes'.

tions" of theoretical computer science[7], the continuous and Borel reductions of descriptive set theory[8], the computable reductions from recursion theory[9], and also the Rudin-Keisler ordering of the theory of ultrafilters.[10] In this paper, we will only talk about comparing sets of objects of the same type. If you want to compare problems from different areas, the corresponding asymmetric version of the reductions mentioned here are the **Galois-Tukey reductions** or **Galois correspondences**.

Given our reducibility relation $\leq_F$ defined from F, we can now define the corresponding equivalence relation

$$A \equiv_F B :\Longleftrightarrow\ A \leq_F B \wedge A \leq_F B,$$

and then look at the set of equivalence classes $C_F := \wp(K)/\equiv_F$. The reducibility relation $\leq_F$ transfers directly to a relation $\leq_F$ on the equivalence classes where it is a partial order. We denote the equivalence class of a set A by $[A]_{\equiv_F}$. The elements of C_F can now be called the F**-complexity classes** or F**-degrees**.

In the following, we shall give examples of K and F and the derived notions of complexity classes.

3 Sets of Reals

In descriptive set theory, the set of objects K is the set of real numbers $\mathbb{R}$. This set is naturally endowed with the canonical topology.

Traditionally, the topological space $\mathbb{R}$ is endowed with several complexity hierarchies of sets of real numbers that –at least *prima facie*– are not derived from complexity functions: the Borel hierarchy and the projective hierarchy.

For the Borel hierarchy, we set $\mathbf{\Sigma}^0_1$ to be the set of all open sets. For any ordinal α, we set $\mathbf{\Pi}^0_\alpha$ to be the collection of complements of sets in $\mathbf{\Sigma}^0_\alpha$ (so $\mathbf{\Pi}^0_1$ is the set of closed sets) and $\mathbf{\Sigma}^0_\alpha$ is the collection of all sets that are countable unions $\bigcup_{i\in\mathbb{N}} A_i$, where each A_i is in some $\mathbf{\Pi}^0_{\alpha_i}$ for $\alpha_i < \alpha$.

[7] *Cf.* [Pap94, Definition 8.1] and Section 6.
[8] *Cf.* Sections 3 and 4.
[9] *Cf.* Section 5.
[10] Note that the Rudin-Keisler ordering is of slightly different type because ultrafilters are sets of sets of numbers: for ultrafilters U and V on $\mathbb{N}$, we say that U is **Rudin-Keisler reducible** to V ($U \leq_{\mathrm{RK}} V$) if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $U := \{f^{-1}[X]\,;\, X \in V\}$.

The index α indicates how often you have to iterate the operations "complementation" and "countable union" to get a set which is in $\mathbf{\Sigma}^0_\alpha$. Thus, in a preconceived notion of complexity, we could call

$$c_{\mathrm{Borel}}(A) := \min\{\alpha\,;\, A \in \mathbf{\Sigma}^0_\alpha \cup \mathbf{\Pi}^0_\alpha\}$$

a complexity measure for Borel sets and can derive a complexity relation

$$A \leq_{\mathrm{Borel}} B :\iff c_{\mathrm{Borel}}(A) \leq c_{\mathrm{Borel}}(B)$$

from it.

Similarly, we can look at the projective hierarchy (defined on finite Cartesian products of $\mathbb{R}$). We let $\mathbf{\Sigma}^1_0$ be the set of Borel sets, and for each n, we let $\mathbf{\Pi}^1_n$ be the collection of complements of sets in $\mathbf{\Sigma}^1_n$. Then $\mathbf{\Sigma}^1_{n+1}$ is defined to be the set of projections of sets in $\mathbf{\Pi}^1_n$, where $A \subseteq \mathbb{R}^n$ is a projection of $B \subseteq \mathbb{R}^{n+1}$ if

$$\vec{x} \in A \iff \exists y(\langle y, \vec{x}\rangle \in B).$$

Again, if we set $c_{\mathrm{Proj}}(A)$ to be the least n such that $A \in \mathbf{\Sigma}^1_n \cup \mathbf{\Pi}^1_n$, we can call the following relation a complexity relation:

$$A \leq_{\mathrm{Proj}} B :\iff c_{\mathrm{Proj}}(A) \leq c_{\mathrm{Proj}}(B).$$

These two hierarchies are proper hierarchies, *i.e.*, for $\alpha < \omega_1$ and $1 \leq n < \omega$, the following inclusions are all proper:[11]

$$\mathbf{\Delta}^0_\alpha \subsetneqq \mathbf{\Sigma}^0_\alpha \cup \mathbf{\Pi}^0_\alpha \subsetneqq \mathbf{\Delta}^0_{\alpha+1} \subsetneqq \mathbf{\Delta}^1_n \subsetneqq \mathbf{\Sigma}^1_n \cup \mathbf{\Pi}^1_n \subsetneqq \mathbf{\Delta}^1_{n+1}.$$

By the technique used to prove this chain of strict inclusions, it is connected to the limitative theorems of logic: As in the limitative theorems, the non-equality of the complexity classes (often called a **hierarchy theorem**) is proved using the method of diagonalization via universal sets.

Even more connections to logic emerge: if you look at the well-known **Lévy hierarchy** of formulas where you count alternations of quantifiers[12], then sets definable with a real number parameter and first-order quantifiers over the standard model of (second-order) arithmetic

[11] Here, $\mathbf{\Delta}^0_\alpha := \mathbf{\Sigma}^0_\alpha \cap \mathbf{\Pi}^0_\alpha$ and $\mathbf{\Delta}^1_n := \mathbf{\Sigma}^1_n \cap \mathbf{\Pi}^1_n$.

[12] A formula with n alternations of quantifiers starting with $\exists$ is called a Σ_{n+1} formula, a formula with n alternations starting with $\forall$ is called Π_{n+1}. The Lévy hierarchy was introduced in [Lév65].

with a Σ_n (Π_n) formula are exactly the $\mathbf{\Sigma}^0_n$ ($\mathbf{\Pi}^0_n$) sets, and those definable with a real number parameter and both first- and second-order quantifiers over the standard model of (second-order) arithmetic with a Σ_n (Π_n) formula are exactly the $\mathbf{\Sigma}^1_n$ ($\mathbf{\Pi}^1_n$) sets.

Thus, Borel and projective complexity fit well into the concept of formula (Lévy) complexity.

These complexity relations are related to reduction functions in the following way: Let F_{W} be the set of continuous functions from $\mathbb{R}$ to $\mathbb{R}$. Then the relation $\leq_{F_{\mathrm{W}}}$ defined on $\wp(\mathbb{R})$ is called **Wadge reducibility** $\leq_{\mathrm{W}}$. If you take a set $A \in \mathbf{\Sigma}^0_\alpha \backslash \mathbf{\Pi}^0_\alpha$, then (using Wadge's Lemma and Borel Determinacy)

$$\{B\,;\, B \leq_{\mathrm{W}} A\} = \mathbf{\Sigma}^0_\alpha.$$

Similarly, (under additional assumptions) we get a description of the projective classes as initial segments of the Wadge hierarchy: $\leq_{\mathrm{W}}$ is a refinement of both $\leq_{\mathrm{Borel}}$ and $\leq_{\mathrm{Proj}}$.

The Wadge hierarchy is one of the most fundamental hierarchies in the foundations of mathematics. Under the assumption of determinacy of games, the Wadge hierarchy is an almost linear and well-founded backbone of the class of sets of real numbers.[13] Since the theory of real numbers and sets of reals is intricately connected to questions about the axiomatic framework of mathematics as a whole, the Wadge hierarchy serves as a stratification of an important part of the logical strength of set-theoretic axiom systems for mathematics.

For a basic introduction into the theory of Wadge degrees, we refer the reader to Van Wesep's basic paper [Van78].

4 Equivalence Classes

Closely connected to the Wadge hierarchy is the area called *Descriptive Set Theory of Borel Equivalence Relations*. Riccardo Camerlo's paper "Classification Problems in Algebra and Topology" in this volume gives an overview of this area of research.

Take two equivalence relations E and F on the set of real numbers $\mathbb{R}$. As relations, they are subsets of $\mathbb{R}\times\mathbb{R}$. A Borel function $f : \mathbb{R} \to \mathbb{R}$ gives

[13] By results of Wadge, Martin, Monk, Steel and van Wesep.

rise to a Borel function $\widehat{f} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}$ by $\widehat{f}(x, y) := \langle f(x), f(y)\rangle$. Look at the class F_{Borel} of functions like this and call $\leq_{F_{\mathrm{Borel}}}$ **Borel reducibility** $\leq_{\mathrm{B}}$. Note that this means that $E \leq_{\mathrm{B}} F$ if and only if there is a Borel function $f : \mathbb{R} \to \mathbb{R}$ such that

$$x\, E\, y \iff f(x)\, F\, f(y).$$

This hierarchy (as opposed to the Borel fragment of the Wadge hierarchy) is not at all linear: Camerlo mentions the Adams-Kechris theorem on p. 73 of his paper in this volume; you can embed the Borel sets (partially ordered by inclusion) into this hierarchy.

Equivalence relations on the real numbers (or on other Polish spaces) play a rôle in classification projects. We will give an example how complexity enters the discussion here (the example is from Simon Thomas' survey article [$\mathrm{Tho}_0$01]):

In 1937, Reinhold Baer had given a (simple) complete invariant for additive subgroups of $\mathbb{Q}$, thus classifying these groups up to isomorphism.[14] Kurosh and Mal'cev gave complete invariants for additive subgroups of $\mathbb{Q}^n$, but "the associated equivalence relation is so complicated that the problem of deciding whether two [invariants are the same] ... is as difficult as that of deciding whether the corresponding groups are isomorphic. It is natural to ask whether the classification problem for [additive subgroups of $\mathbb{Q}^n$] ... is genuinely more difficult when $n \geq 2$. [$\mathrm{Tho}_0$01, p. 330]".

This problem has been solved by Hjorth [Hjo99] by showing that the isomorphism equivalence relation for subgroups of $\mathbb{Q}^2$ has strictly higher complexity than the isomorphism equivalence relation for subgroups of $\mathbb{Q}$ and in fact, that it is strictly more complicated than E_0, the first non-smooth Borel equivalence relation (*cf.* Camerlo's Theorem 2.2).[15]

This complexity result can be seen as a metatheoretical explanation for the fact that Kurosh and Mal'cev couldn't come up with better invariants for these classes of groups.

[14] Subgroups of $\mathbb{Q}$ can be seen as subsets of $\mathbb{N}$ via a bijection between $\mathbb{Q}$ and $\mathbb{N}$. Via characteristic functions, subsets of $\mathbb{N}$ can be interpreted as infinite 0-1-sequences, and those can be seen as a real number (*e.g.*, via the binary expansion). Thus, the isomorphism relation on subgroups of $\mathbb{Q}$ can be investigated as an "equivalence relation on the real numbers".

[15] Hjorth comments on the connection between this theorem and certain pretheoretical notions of classifiability in his [Hjo00, p. 55, fn. 2].

5 Recursion Theory

From sets of reals, we move to sets of integers now. In Recursion Theory, K is the set of natural numbers $\mathbb{N}$. We shall look at two different kinds of reduction functions: the set F_{t} of total recursive functions, and the set F_1 of total injective recursive functions.

Each of these sets of reductions gives rise to recursion-theoretic reducibility relations: $\leq_{F_{\mathrm{t}}}$ is known as **many-one reducibility** $\leq_{\mathrm{m}}$, and $\leq_{F_1}$ is known as **1-reducibility** $\leq_1$.[16]

One of the mentioned three classical limitative theorems stems from Recursion Theory: Let us assume that we're looking at computer programs in binary code. We can ask whether a program, given its own code as an input, will eventually halt or run into an infinite loop. Call the former programs **halting** and the latter **looping**. Now, is there a program that, given any binary code b, can determine whether the program with code b is halting or looping? Diagonalization easily shows that there can't be such a program. The set

$$H := \{b\,;\text{ the machine with code } b \text{ is halting}\}$$

is called **Turing's halting problem**: in terms of complexity the diagonalization argument shows that $H \not\leq_{\mathrm{m}} A$ for any computable set A (and so also $H \not\leq_1 A$).

Hence the derived sets of degrees $\langle C_{F_{\mathrm{t}}}, \leq_{\mathrm{m}}\rangle$, and $\langle C_{F_1}, \leq_1\rangle$ are non-trivial, and, in fact, as the degrees of Borel equivalence relations, they are very far from being linear orders. A huge part of the literature on recursion-theoretic degrees investigates the structure of the complexity hierarchies of recursion theory. Even more prominently than $\leq_1$ and $\leq_{\mathrm{m}}$ features **Turing reducibility** $\leq_{\mathrm{T}}$ defined by the notion of **computation in an oracle**. The literature on complexity hierarchies in recursion theory also includes Sacks' seminal [Sac71] in which he introduced Sacks forcing. The properties of Sacks forcing used in recursion theory are similar to the minimality property proved in Lemma 2.8 of the survey paper on set-theoretic properties of Sacks forcing by Geschke and Quickert in this volume.

Of course, the above definitions are not restricted to the classical notion of computation. When you change the model of computation and

[16] *Cf.* [Hin78] and [Soa87].

replace "recursive"/"computable" in the above definitions by some predicate connected to a generalized model of computation, you get new complexity hierarchies with a lot of structure to investigate. One instance of this is the notion of the **Infinite Time Turing Machine** introduced by Hamkins and Kidder. These machines have the same architecture as normal Turing machines but can go on through the ordinals in their computations. Being computable by an Infinite Time Turing Machine is a much more liberal property of functions, so the hierarchies we get are much coarser.

In this volume, there are two papers dealing with Infinite Time Turing Machines: a gentle introduction by Joel Hamkins entitled "Supertask Computation" and a paper by Philip Welch that discusses supertasks on sets of reals, thus connecting Infinite Time recursion theory to descriptive set theory (*cf.* Section 3).

Having seen the rich structure theory of recursion theoretic hierarchies for the classical notion of computability, one can ask with Hamkins (Question 2 of his contribution to this volume):

> *What is the structure of infinite time Turing degrees? To what extent do its properties mirror or differ from the classical structure?*[17]

6 Complexity Theory in Computer Science

Computable functions, the reductions used in recursion theory, are not good enough in theoretical computer science: a computable function in general has no bound on computing time or storage space that is used while it's being computed. The fine hierarchies of computer science that want to distinguish between problems that are solvable in a realistic time-frame and those that require too much time cannot be content with reductions of that sort.

The class of reductions in theoretical computer science is the class of functions computable with logarithmic space usage (**logspace reductions**). *I.e.*, there is an algorithm that computes $f(x)$ from x and satisfies

[17] In his [$\text{Wel}_0$99], Philip Welch discusses an aspect of Hamkins' Question 2: minimality in the Infinite Time Turing degrees. Again, this is connected to minimality of Sacks reals as discussed in the survey of Geschke and Quickert, and Welch uses a variant of Sacks' argument from [Sac71].

the following condition: if the input has n bits, the program will never use more than $c \cdot \log(n)$ bits of storage space during the computation (where c is a constant).[18]

If F_ℓ is the class of logspace reductions, then $\langle C_{F_\ell}, \leq_{F_\ell}\rangle$ is the usual world of complexity classes: first of all the deterministic ones, the class of polynomial time decidable sets $\mathbf{P}$, those decidable with polynomial scratch space usage $\mathbf{PSPACE}$, those decidable in exponential time $\mathbf{EXP}$, and also their nondeterministic counterparts $\mathbf{NP}$, $\mathbf{coNP}$, $\mathbf{NPSPACE}$, $\mathbf{NEXP}$, and others. The usual inclusion diagram of complexity classes (*cf.* [Pap94, §§ 7.2 & 7.3]) can be verified easily:

$$\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP} \subseteq \mathbf{NEXP}.$$

While there are some non-equality results, one puzzling and titillating feature of the complexity classes of computer science is that we don't know for sure that all of the inclusions are proper. In particular, we are lacking the limitative theorem $\mathbf{P} \neq \mathbf{NP}$. For the reader unfamiliar with structural complexity theory, Rod Downey's "Invitation to structural complexity" [Dow92] will be a nice way to approach the subject.

As in recursion theory, also in complexity theory, a new model of computation gives rise to new classes of reduction functions, and thus, *a fortiori*, to different hierarchies.[19] Examples are the Blum-Shub-Smale theory of computation with real numbers[20] or quantum computability as discussed in Ambainis' survey in this volume.

7 Other Complexity Hierarchies

We abstractly discussed hierarchies derived in a particular way from reduction functions. Of course, there are more complexity hierarchies in logic than that.

Fundamental are the hierarchies of proof theory that are related to reduction functions in a less direct way: Fix a language $\mathcal{L}$ and a collection

[18] *Cf.* [Pap94, Chapter 8].

[19] Interesting to note is the connection to Infinite Time Turing Machines: Schindler opened the discussion on analogues of the $\mathbf{P} = \mathbf{NP}$ question for Infinite Time Turing Machines in his [Sch$_2\infty$]. *Cf.* also [HamWel$_0$03] and [DeoHamSch$_2\infty$].

[20] *Cf.* [Blu+98]; in Footnote 4 of his article in this volume, Hamkins mentions that the Blum-Shub-Smale theory was the key motivation for the development of Infinite Time Turing Machines.

Φ of $\mathcal{L}$-sentences. From now on, we mean by an **$\mathcal{L}$-theory** a primitively recursive set of $\mathcal{L}$-sentence (understood as an axiom system of the theory). Let $\mathsf{Proof}(v_0, v_1, v_2)$ be the predicate "v_0 is the Gödel number of a proof of the sentence with the Gödel number v_1 in the theory with the index v_2". If S and T are $\mathcal{L}$-theories, we call a function f a **Φ-reduction of S to T**, if it is primitively recursive and

$$\mathsf{PRA} \vdash \forall\varphi \in \Phi \, \forall n \; (\, \mathsf{Proof}(n, \ulcorner\varphi\urcorner, \ulcorner S\urcorner) \;\rightarrow\; \mathsf{Proof}(f(n), \ulcorner\varphi\urcorner, \ulcorner T\urcorner)\,)\,.$$

We write $S \leq_\Phi T$ if such a Φ-reduction exists.[21]

As a famous special case, you can look at $\Phi_* = \{\bot\}$. Then the existence of a Φ_*-reduction says that every proof of an inconsistency in S can be effectively transformed into a proof of an inconsistency in T. In other words, if T is consistent, then so is S. This gives rise to the **consistency strength hierarchy** $\leq_{\mathrm{Cons}}$.[22] Gödel's second incompleteness theorem states that $\mathsf{PA} <_{\mathrm{Cons}} \mathsf{PA} + \mathsf{Cons}(\mathsf{PA})$, so the consistency strength hierarchy is nontrivial. Extending Peano arithmetic to second-order arithmetic or even set theory gives a multitude of interesting new systems that form an increasing hierarchy in the ordering $\leq_{\mathrm{Cons}}$. It is known [Rat99, Proposition 2.18] that the consistency strength hierarchy in general is not linearly ordered. Yet, surprisingly, for a large class of axiom systems (and, importantly, all of the axiom systems considered to be natural are among them) the relation has been empirically established to be a linear order, actually a well-ordering.[23] As such, it has served as the measuring rod of logical strength in foundations of mathematics for many years and is subtly connected to some of the mentioned other complexity hierarchies as the recursion theoretic hierarchies and the Wadge hierarchy. Some subareas of logic have specialized on certain fragments of the consistency strength hierarchy, among them are systems of second order arithmetic investigated by reverse mathematics, these and other systems of proof theory by proof theoretic ordinal analysis, and systems of higher set theory by large cardinal theory[24].

---

[21] Here, PRA is **primitive recursive arithmetic**, a weak subsystem of second-order arithmetic; *cf.* [$\mathrm{Sim}_2$99].

[22] *Cf.* [Rat99, §§ 2.5 & 2.7].

[23] *Cf.* [$\mathrm{Ste}_0$82a] for a mathematical discussion of the connections between the theory of inner models of set theory and this remarkable fact.

[24] *Cf.* [$\mathrm{Sim}_2$99], [Poh96], [Kah02], and [$\mathrm{Kan}_0$94]. For an example of a strength analysis by large cardinals, see Schindler's paper in this volume where he computes the consistency strength

The relations between the different aspects and contexts of complexity are large in number, and a short introductory article to this volume can't list them all. The fact that we can only provide a very passing glance at the fascinating subject of complexity stresses the importance, the liveliness and vigour of the topic of the conference FotFS III. The slightly different but deeply related notions of complexity in the several areas discussed in this volume are entrenched within the research communities of mathematical logic and computer science and will serve as a bridge between the two subject areas for years to come.

of the theory $\mathsf{BPFA}+$"every projective set of reals is Lebesgue measurable" in terms of large cardinals.