Chapter 1

Introduction

> Wenn es eine Maschine mit [...] gäbe, hätte das Folgerungen von der größten Tragweite. Es würde offenbar bedeuten, daß man trotz der Unlösbarkeit des Entscheidungsproblems die Denkarbeit der Mathematiker bei ja-oder-nein Fragen vollständig (abgesehen von der Aufstellung der Axiome) durch eine Maschine ersetzen könnte.[1]
>
> Kurt Gödel

Proof complexity is the area of research within complexity theory whose main aim is to understand and classify the complexity of theorem-proving procedures. Proof complexity is a theory which provides a very promising approach based mainly on mathematical logic, on model theory, and on combinatorics to some of the main questions and problems in complexity theory, as for instance the exact relationship between the classes P and NP.

Since its origins in the late sixties, computational complexity considered as its main computational paradigm the classical computational model of the Turing machine, invented by Alan Turing in the thirties [Tur36]. In the last 25 years, complexity theory moved forward to expand the concept of computational model. New models of computation were introduced, also looking at and exploring other scientific disciplines as physics or advanced mathematics, like probability theory. The new computational models studied frequently involve alternative resources such as randomization, quantum computation, or even a limited amount non-computable information. The investigation of the main questions of complexity theory from the point of view of these new computational paradigms and models was very fertile and fruitful in the last twenty years. For instance, quantum algorithms have been proved to be strictly more efficient than classical algorithms on very important problems. The use of randomization, among many other

[1] Kurt Gödel in a letter to John von Neuman in 1956 (reprinted in [Göd93]).

aspects, has provided us with the new theory of probabilistically checkable proofs which represents one of the main attempts to tackle questions like P vs. NP.

Proof complexity is younger than complexity theory and so far its investigation has been essentially based only on the classical computational model of the Turing machine. Only in the very last years some leading research groups have initiated the study of proof complexity from the point of view of other models of computation. This thesis focuses on developing and contributing to these recent lines of research by considering two non-classical aspects from computational complexity for theorem proving: proof systems with advice and parameterized proof systems. In the following section we will give a brief overview of our main results on these two models.

1.1 Motivation, Models, and Main Results

1.1.1 Proof Systems with Advice

Complexity classes with advice were introduced by Karp and Lipton [KL80]. The idea is here to enhance efficient computations with a limited amount of non-uniform information: *the advice*. By using advice we leave the realm of effective computability because the advice can be arbitrarily complex, even non-computable. But we impose limits on the *amount* of advice that we are allowed to use and in this way obtain interesting computational models such as Boolean circuits [Pip79].

Recently, Cook and Krajíček [CK07] introduced *proof systems with advice* which—similarly as in the complexity approach mentioned above—may use a limited amount of non-uniform information for the verification of proofs. Their results show that, like in the classical Cook-Reckhow setting, these proof systems enjoy a close connection to theories of bounded arithmetic. Moreover, Cook and Krajíček obtained the surprising result that with only one bit of advice, an *optimal* proof system can be realized. The existence of such an optimal proof system, i. e., the strongest possible system, is not known in the classical model. Thus proof systems with advice appear to be a strictly more powerful model.

In this thesis we provide a rigorous development of the theory of proof systems with advice and investigate the following fundamental questions for this new model:

- Q1: Given a language L, do there exist polynomially bounded proof systems with advice for L?
- Q2: For propositional proof systems, does advice help to shorten proofs?
- Q3: Do there exist optimal proof systems with advice for L?

For question Q1, one of the major motivations for proof complexity [CR79], we obtain a complete complexity-theoretic characterization. The classical Cook-Reckhow Theorem states that NP = coNP if and only if the set of all tautologies

TAUT has a polynomially bounded proof system, i.e., there exists a polynomial p such that every tautology φ has a proof of size $\leq p(|\varphi|)$ in the system. Consequently, showing super-polynomial lower bounds to the proof size in propositional proof systems of increasing strength provides one way to attack the P vs. NP problem. This approach, also known as the Cook-Reckhow program, has led to a very fruitful research on the length of propositional proofs (cf. [Pud98]).

As in the Cook-Reckhow Theorem above, we obtain a series of results leading to a complete characterization for Q1. In particular, we show a tight connection of this problem to the notion of nondeterministic instance complexity. Similarly as Kolmogorov complexity, instance complexity measures the complexity of individual instances of a language [OKSW94]. In its nondeterministic version, Arvind, Köbler, Mundhenk, and Torán [AKMT00] used this complexity measure to show that, under reasonable complexity-theoretic assumptions, there are infinitely many tautologies that are hard to prove in every propositional proof system. In the light of our investigation, this connection between nondeterministic instance complexity and proof complexity is strengthened by results of the following form: *all elements of a given language L have small instance complexity if and only if L has a proof system with advice such that every $x \in L$ has a short proof.*

For question Q2 we concentrate on the most interesting case of propositional proof systems. Unfortunately, proof systems with advice do not constitute a feasible model for the verification of proofs in practice, as the non-uniform advice can be very complex (and even non-recursive). Approaching question Q2, we therefore investigate whether the advice can be simplified or even eliminated without increasing the proof length. Our first result in this direction shows that proving propositional tautologies does not require complicated or even non-recursive advice: every propositional proof system with up to logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse NP-oracle. Thus in propositional proof complexity, computation with advice can be replaced by a more realistic computational model.

While this result holds unconditionally, our next two results explore consequences of a positive or negative answer to question Q2. Assume first that advice helps to prove tautologies in the sense that proof systems with advice admit nontrivial upper bounds on the lengths of proofs. Then we show that the same upper bound can be achieved in a proof system with a simplified advice model. On the other hand, if the answer is negative in the sense that advice does not help to shorten proofs even for simple tautologies, then we obtain optimal propositional proof systems without advice.

This brings us to our last question Q3. While the existence of optimal proof systems in the classical model is a prominent open problem posed by Krajíček and Pudlák twenty years ago [KP89], question Q3 receives a surprising positive answer: optimal proof systems exist when a small amount of advice is allowed. For propositional proof systems this was already shown by Cook and Krajíček

[CK07]. Using the proof technique from [CK07], we show that for every language L, the class of all proof systems for L using logarithmic advice contains an optimal proof system and investigate whether the optimality result can be strengthened to it's efficient version of p-optimality. In addition, we show that the connection between optimal proof systems and promise classes also holds in the presence of advice.

Propositional proof systems enjoy a close connection to bounded arithmetic (cf. the monographs [Kra95, CN10] or the survey [Bey09]). Cook and Krajíček [CK07] use the correspondence between proof systems with advice and arithmetic theories to obtain a very strong Karp-Lipton collapse result in bounded arithmetic: if SAT has polynomial-size Boolean circuits, then the polynomial hierarchy collapses to the Boolean hierarchy. In Chapter 5 we show that this collapse consequence is in fact optimal with respect to the theory PV, thereby answering a question of Cook and Krajíček [CK07].

1.1.2 Parameterized Proof Systems

Parameterized complexity is widely considered one of the modern paradigms of computational complexity which considerably advances our understanding of intractable problems by offering a refined view on running times of algorithms. In proof complexity, this investigation has started recently with the work of Dantchev, Martin, and Szeider [DMS07]. There the authors introduce a general framework for parameterized proof complexity and consider a parameterized version of Resolution which is the best studied and most important propositional proof system in terms of applications.

In parameterized proof complexity, our main objective is to reach a more refined understanding of theorem proving by adapting concepts and techniques from parameterized complexity to proof complexity. Proof systems can be understood as non-deterministic algorithms for the tautology problem. Therefore, by considering parameterized proof systems we reach a better understanding of the borderline between efficiency and non-efficiency for *non-deterministic* algorithms. In proof complexity this condensates in a more refined classification of proof lengths. This view is supported by previous results from [DMS07, Gao09] and our investigation in this dissertation. For example, the hard case in the classical dichotomy for tree-like Resolution of Riis [Rii01] splits in the parameterized context into two cases: tautologies with fpt-bounded proofs and tautologies for which the shortest parameterized proof has size similar to exhaustive search, as shown in [DMS07].

In Chapter 7 we show that in contrast to classical Resolution, *Parameterized Resolution* appears to be a relatively powerful proof system as a number of classically hard principles admit fpt-bounded proofs even in *tree-like* Parameterized Resolution. We show this by transferring the concept of a kernel from parameterized complexity to proof complexity and constructing kernelizations

for many classically hard principles as the class of all CNF's of bounded width. Specific examples of formulas which are hard for classical Resolution, but possess fpt-bounded proofs even in tree-like Parameterized Resolution include the linear ordering principle, pebbling tautologies, coloring principles, and Tseitin tautologies.

For hardness results we introduce a powerful two-player game to model and study the complexity of proofs in tree-like Parameterized Resolution. Our game refines the Prover-Delayer game of Pudlák and Impagliazzo [PI00] and makes it applicable in situations where the proof trees are very unbalanced. This technique also yields improved lower bounds for non-parameterized proof systems as we show in Chapter 6.

Although the Prover-Delayer game is a very general technique, it cannot be used for *dag-like* proofs. In Section 7.8 we obtain the first lower bound for dag-like Parameterized Resolution for the pigeonhole principle. For this lower bound we again use a game-theoretic argument originating in Pudlák's work [Pud00].

1.2 Organization of the Thesis and Published Parts

This thesis is organized as follows. Chapters 2 and 3 contain background information on proof complexity and computational complexity, respectively. These two chapters are largely of preliminary nature. Apart from known definitions and results, Sections 3.4 to 3.6 contain some new results on non-deterministic instance complexity, promise classes, and optimal proof systems which we apply in Chapter 4.

In Chapters 4 and 5 we investigate proof systems with advice, first from the perspective of computational complexity (Chapter 4) and then with respect to their relation to bounded arithmetic (Chapter 5).

In Chapter 6 we introduce a new technique for lower bounds in tree-like proof systems—the asymmetric Prover-Delayer game—and apply it to classical Resolution. Chapter 7 then contains our investigation of parameterized proof complexity and in particular of Parameterized Resolution where we again use the game of Chapter 6.

Chapter 8 concludes with a discussion of our two non-classical aspects that we investigate here and puts this work into a broader context.

Part of the results from this thesis are already published in journals or in conference proceedings. The relevant publications are

- [BKM11] containing Section 3.4 and most of Chapter 4;
- [BS11] containing Sections 3.5 and 3.6;
- [BM10b] containing Section 4.4.2 and all of Chapter 5;

- [BGL10] containing Chapter 6;
- [BGL11] and [BGLR11] containing most of the material in Chapter 7.

Chapter 2

Proof Complexity

> Nach diesen Bemerkungen sei es dem Verfasser noch erlaubt, einige Worte für sich anzuführen. Er hat sich bemüht, so kurz zu schreiben, als es ihm möglich war und es diese Gattung von Arbeiten erfordert. Es wäre zu wünschen, daß man sich dieses Gesetz der Kürze bei allen Büchern über das Altertum, die doch nicht unser ganzes Leben beschäftigen sollen, vorhalten möchte. Die meisten antiquarischen Schriftsteller gleichen durch ihre Weitschweifigkeit den Flüssen, die anschwellen, wenn man ihres Wassers nicht bedarf, und trocken bleiben, wo eben Wasser nötig wäre.[1]
>
> Johann Joachim Winckelmann

One of the starting points of propositional proof complexity is the seminal paper of Cook and Reckhow [CR79] where they formalized propositional proof systems as polynomial-time computable functions which have as their range the set of all propositional tautologies. In that paper, Cook and Reckhow also observed a fundamental connection between lengths of proofs and the separation of complexity classes: they showed that there exists a propositional proof system which has polynomial-size proofs for all tautologies (a *polynomially bounded* proof system) if and only if the class NP is closed under complementation. From this observation the so called *Cook-Reckhow program* was derived which serves as one of the major motivations for propositional proof complexity: to separate NP from coNP (and hence P from NP) it suffices to show super-polynomial lower bounds to the size of proofs in all propositional proof systems.

Although the first super-polynomial lower bound to the lengths of proofs had already been shown by Tseitin in the late 60's for a sub-system of Resolution [Tse68], the first major achievement in this program was made by Haken in 1985

[1] Winckelmann in der Vorrede der *Beschreibung der geschnittenen Steine des seligen Baron Stosch* (Florenz, 1760)

when he showed an exponential lower bound to the proof size in Resolution for a sequence of propositional formulas describing the pigeonhole principle [Hak85]. In the last two decades these lower bounds were extended to a number of further propositional systems such as the Nullstellensatz system [BIK+96], Cutting Planes [BPR97, Pud97], Polynomial Calculus [CEI96, Raz98], or bounded-depth Frege systems [Ajt94, BIK+92, BPI93, KPW95]. For all these proof systems we know exponential lower bounds to the lengths of proofs for concrete sequences of tautologies arising mostly from natural propositional encodings of combinatorial statements.

For proving these lower bounds, a number of generic approaches and general techniques have been developed. Most notably, there is the method of feasible interpolation developed by Krajíček [Kra97], the size-width trade-off introduced by Ben-Sasson and Wigderson [BSW01], and the use of pseudorandom generators in proof complexity [ABSRW04, Kra01, Kra04a].

Despite this enormous success many questions still remain open. In particular Frege systems currently form a strong barrier [BBP95], and all current lower bound methods seem to be insufficient for these strong systems. A detailed survey of recent advances in propositional proof complexity is contained in [Seg07].

Let us mention that the separation of complexity classes is not the only motivation for studying lengths of proofs. In particular for strong systems like Frege and its extensions there is a fruitful connection to bounded arithmetic which adds insight to both subjects (cf. [Kra95]). Further, understanding weak systems as Resolution is vital to applications as the design of efficient SAT solvers (see e. g. [PS10] for a more elaborate argument). Last not least, propositional proof complexity has over the years grown into a mature field and many researchers believe that understanding propositional proofs and proving lower bounds—arguably the hardest task in complexity—is a very important and beautiful field of logic which is justified in its own right.

2.1 Proof Systems

We start with a general semantic definition of proof systems:

Definition 2.1.1 *A* proof system *for a language L is a (possibly partial) surjective function $f : \Sigma^* \to L$. For $L = \mathrm{TAUT}$, f is called a* propositional proof system.

In the classical framework of Cook and Reckhow [CR79], proof systems are additionally required to be computable in polynomial time. As we are relaxing this definition in subsequent chapters we have chosen the more general semantic definition above where the computational resources to compute f are not specified.

We review important notions concerning proof systems. A string w with $f(w) = x$ is called an *f-proof* of x. Proof complexity studies lengths of proofs, so we use the following notion: for a function $t : \mathbb{N} \to \mathbb{N}$, a proof system f for L is *t-bounded* if every $x \in L$ has an f-proof of size $\leq t(|x|)$. If t is a polynomial, then f is called *polynomially bounded*. We recall the classical theorem of Cook and Reckhow on polynomially bounded proof systems:

Theorem 2.1.2 (Cook, Reckhow [CR79]) *A language L has a polynomially bounded proof system if and only if $L \in \mathsf{NP}$.*

2.2 Simulations and Optimal Proof Systems

Proof systems are compared according to their strength by simulations as introduced in [CR79] and [KP89]. If f and g are proof systems for L, we say that g *simulates* f (denoted $f \leq g$), if there exists a polynomial p such that for all $x \in L$ and f-proofs w of x there is a g-proof w' of x with $|w'| \leq p(|w|)$. If such a proof w' can even be computed from w in polynomial time, we say that g *p-simulates* f and denote this by $f \leq_p g$. If the systems f and g mutually (p-)simulate each other they are called *(p-)equivalent*, denoted by $f \equiv_{(p)} g$. A proof system for L is *(p-)optimal* if it (p-)simulates all proof systems for L.

Whether or not there exist optimal propositional proof system is open. Posed by Krajíček and Pudlák [KP89], this question has remained unresolved for more than twenty years. Sufficient conditions were established by Krajíček and Pudlák [KP89] by $\mathsf{NE} = \mathsf{coNE}$ for the existence of optimal and $\mathsf{E} = \mathsf{NE}$ for p-optimal propositional proof systems, and these conditions were subsequently weakened by Köbler, Messner, and Torán [KMT03]. Necessary conditions for the existence of optimal proof systems are tightly linked to the following question for promise complexity classes lacking an easy syntactic machine model:

Problem 2.2.1 *Do there exist complete problems for a given promise class C?*

Like the first question of the existence of optimal proof systems also Problem 2.2.1 has a long research record, dating back to the 80's when Kowalczyk [Kow84] and Hartmanis and Hemachandra [HH88] considered this question for $\mathsf{NP} \cap \mathsf{coNP}$ and UP. This research agenda continues to recent days where, due to cryptographic and proof-theoretic applications, disjoint NP-pairs have been intensively studied (cf. [GSS05, GSSZ04, GSZ07, Bey07] and [GSZ06] for a survey). Very recently, Itsykson has shown the surprising result that AvgBPP, the average-case version of BPP, has a complete problem [Its09].

Understanding these questions better through characterizations is an important problem with consequences to seemingly unrelated areas such as descriptive complexity: very recently, Chen and Flum [CF10] have shown that the existence of an optimal propositional proof system is equivalent to the open problem

whether $L_{\leq}$ is a P-bounded logic for P. Other recent research concentrated on modified versions of Q1, where a number of surprising positive results have been obtained. Cook and Krajíček [CK07] have shown that optimal propositional proof systems exist under non-uniform information (advice), and even one bit of advice suffices (we will discuss this result in detail in Section 4.4). In another direction, Hirsch and Itsykson [HI10, Hir10] considered randomized proof systems and showed the existence of an optimal system in the class of all automatizable heuristic proof systems (cf. Chapter 8). Still another positive result was very recently obtained by Pitassi and Santhanam [PS10] who show that there exists an optimal quantified propositional proof system under a weak notion of simulation.

2.3 Two Examples of Proof Systems

We give two important examples of propositional proof systems which we will need later on: Resolution and Frege systems.

We start with Resolution. A *literal* is a positive or negated propositional variable and a *clause* is a set of literals. The *width* of a clause is the number of its literals. A clause is interpreted as the disjunctions of its literals and a set of clauses as the conjunction of the clauses. Hence clause sets correspond to formulas in CNF. The *Resolution system* is a refutation system for the set of all unsatisfiable CNF. Resolution uses as its only rule the *Resolution rule*

$$\frac{\{x\} \cup C \qquad \{\neg x\} \cup D}{C \cup D}$$

for clauses C, D and a variable x. The aim in Resolution is to demonstrate unsatisfiability of a clause set by deriving the empty clause. If in a derivation every derived clause is used at most once as a prerequisite of the Resolution rule, then the derivation is called *tree-like*, otherwise it is *dag-like*. The *size* of a Resolution proof is the number of its clauses where multiple instances of the same clause are counted separately. Undoubtedly, Resolution is the most studied and best-understood propositional proof system (cf. [Seg07]).

Our second example are Frege systems. Frege systems derive formulas using axioms and rules. In texts on classical logic these systems are usually referred to as Hilbert-style systems, but in propositional proof complexity it has become customary to call them Frege systems [CR79].

A *Frege rule* is a $(k+1)$-tuple $(\varphi_0, \varphi_1 \ldots, \varphi_k)$ of propositional formulas such that

$$\{\varphi_1, \varphi_2, \ldots, \varphi_k\} \models \varphi_0 \ .$$

The standard notation for rules is

$$\frac{\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_k}{\varphi_0} \ .$$