Does Advice Help to Prove Propositional Tautologies?

Olaf Beyersdorff[1] and Sebastian Müller[2]*

[1] Institut für Theoretische Informatik, Leibniz-Universität Hannover, Germany
beyersdorff@thi.uni-hannover.de
[2] Institut für Informatik, Humboldt-Universität zu Berlin, Germany
smueller@informatik.hu-berlin.de

Abstract. One of the starting points of propositional proof complexity is the seminal paper by Cook and Reckhow [6], where they defined propositional proof systems as poly-time computable functions which have all propositional tautologies as their range. Motivated by provability consequences in bounded arithmetic, Cook and Krajíček [5] have recently started the investigation of proof systems which are computed by poly-time functions using advice. While this yields a more powerful model, it is also less directly applicable in practice.
In this note we investigate the question whether the usage of advice in propositional proof systems can be simplified or even eliminated. While in principle, the advice can be very complex, we show that proof systems with logarithmic advice are also computable in poly-time with access to a sparse NP-oracle. In addition, we show that if advice is "not very helpful" for proving tautologies, then there exists an optimal propositional proof system without advice. In our main result, we prove that advice can be transferred from the proof to the formula, leading to an easier computational model. We obtain this result by employing a recent technique by Buhrman and Hitchcock [4].

1 Introduction

Propositional proof complexity studies the question how difficult it is to prove propositional tautologies. In the classical Cook-Reckhow model, proofs are verified in deterministic polynomial time [6]. While this is certainly the most useful setting for practical applications, it is nevertheless interesting to ask if proofs can be shortened when their verification is possible with stronger computational resources. In this direction, Cook and Krajíček [5] have recently initiated the study of proof systems which use advice 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.

Subsequently, in [2, 3] we studied the questions whether there exist polynomially bounded or optimal proof systems with advice. For the first question, one of the major motivations for proof complexity [6], we obtained a complete complexity-theoretic characterization [2]. Unlike in the classical model, the second question receives a surprising positive answer: optimal proof systems exist when a small amount of advice is allowed [5, 3].

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). In this short paper we therefore investigate

* Supported by DFG grant KO 1053/5-2

the question whether the advice can be simplified or even eliminated while still preserving the same upper bounds on the lengths of proofs. Our first result shows that proving propositional tautologies does not require complicated 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 first result holds unconditionally, our next two results explore consequences of a positive or negative answer to our question in the title. Assume first that advice helps to prove tautologies in the sense that proof systems with advice admit non-trivial 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.

2 Proof Systems with Advice – and without

We start with a general semantic definition of proof systems:

Definition 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.

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.*

In the classical framework of Cook and Reckhow [6], proof systems are additionally required to be computable in polynomial time. Recently, Cook and Krajíček [5] have started to investigate propositional proof systems that are computable in polynomial time with the help of advice. This can be formalized as follows:

Definition 2 ([2]). *For a function $k : \mathbb{N} \to \mathbb{N}$, a proof system f for L is a* proof system with k bits of advice*, if there exist a polynomial-time Turing transducer M, an advice function $h : \mathbb{N} \to \Sigma^*$, and an advice selector function $\ell : \Sigma^* \to 1^*$ such that*

1. *ℓ is computable in polynomial time,*
2. *M computes the proof system f with the help of the advice h, i.e., for all $\pi \in \Sigma^*$, $f(\pi) = M(\pi, h(|\ell(\pi)|))$, and*
3. *for all $n \in \mathbb{N}$, the length of the advice $h(n)$ is bounded by $k(n)$.*

We say that f *uses k bits of input advice* if ℓ has the special form $\ell(\pi) = 1^{|\pi|}$. On the other hand, in case $\ell(\pi) = 1^{|f(\pi)|}$, then f is said to *use k bits of output advice.* The latter notion is only well-defined if we assume that the length of the output $f(\pi)$ (in case $f(\pi)$ is defined) does not depend on the advice. By this definition, proof systems with input advice use non-uniform information depending on the length of the proof, while proof systems with output advice use non-uniform information depending on the length of the proven formula.

In [2] we have shown that every proof system with advice is equivalent to a proof system with the same amount of input advice, whereas output advice seems to yield a strictly less powerful model. Yet, even output advice can be arbitrarily complex and thus computing proofs with advice does not form a feasible model to use in practice. Our first result shows that instead of possibly complex non-uniform information we can also use sparse NP-oracles to achieve the same proof lengths as in proof systems with advice. The qualification "same proof length" is made precise by the notion of simulation [8]: a proof system g *simulates* a proof system f, denoted $f \leq g$, if there is a polynomial p such that for every f-proof π there exists a g-proof π' of size $\leq p(|\pi|)$ with $f(\pi) = g(\pi')$.

Theorem 3.

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*
2. *Conversely, every propositional proof system computable in polynomial time with access to a sparse* NP*-oracle is simulated by a propositional proof system with logarithmic advice.*

Proof. For the first claim, let f be a propositional proof system computed by the polynomial-time Turing transducer M_f with advice function h_f where $|h_f(n)| \leq c \cdot \log n$ for some constant c. Without loss of generality, we may assume that f uses input advice (cf. [2]). We choose a length-injective polynomial-time computable pairing function $\langle \cdot \rangle$ and consider the set

$$A = \Big\{ \langle 1^n, a \rangle \mid a \in \Sigma^{\leq c \cdot \log n} \text{ and for some } \pi \in \Sigma^n,\ M_f(\pi, a) \notin \mathrm{TAUT} \Big\} \quad ,$$

where $M_f(\pi, a)$ denotes the computation of M_f on input π under advice a. Intuitively, A collects all incorrect advice strings for M_f on length n. By construction, A is sparse. Further, $A \in \mathsf{NP}$ because on input $\langle 1^n, a \rangle$ we can guess $\pi \in \Sigma^n$ and nondeterministically verify $M_f(\pi, a) \notin \mathrm{TAUT}$ by guessing a satisfying assignment for $\neg M_f(\pi, a)$.

We now construct a polynomial-time oracle Turing transducer M_g which under oracle A computes a proof system $g \geq f$. Proofs in g will be of the form $\langle \pi, \varphi \rangle$. On such input, M_g queries all strings $\langle 1^{|\pi|}, a \rangle$, $a \in \Sigma^{\leq c \cdot \log |\pi|}$. For each negative answer, M_g simulates M_f on input π using a as advice. If any of these simulations outputs φ, then M_g also outputs φ, otherwise $g(\langle \pi, \varphi \rangle)$ is undefined. Because M_g performs at most polynomially many simulations of M_f, the machine M_g runs in polynomial time. Correctness and completeness of g follow from the fact that M_f is simulated with all correct advice strings, and the original advice used by M_f is among these (as also other advice strings are used, g might have shorter proofs than f, though).

For the second claim, let f be a propositional proof system computed by the oracle transducer M_f under the sparse NP-oracle A. Let M_A be an NP-machine for A and let $p(n)$ be a polynomial bounding the cardinality of $A \cap \Sigma^{\leq n}$ as well as the running times of M_A and M_f. With these conventions, there are at most $q(n) = p(p(n))$ many strings in A that M_f may query on inputs of length n.

We now define a machine M_g, an advice function h_g, and an advice selector ℓ_g which together yield a propositional proof system $g \geq f$ with logarithmic advice. The advice function will be $h_g(n) = |A \cap \Sigma^{\leq p(n)}|$. As A is sparse this results in logarithmic advice. Proofs in the system g are of the form

$$\pi_g = \langle a_1, w_1, \ldots, a_{q(n)}, w_{q(n)}, \pi_f \rangle$$

where $\pi_f \in \Sigma^n$ (an f-proof), $a_1, \ldots, a_{q(n)} \in \Sigma^{\leq p(n)}$ (elements from A), and $w_1, \ldots, w_{q(n)} \in \Sigma^{\leq q(n)}$ (computations of M_A). At such a proof π_g, the advice selector chooses the advice corresponding to $|\pi_f|$, i.e., we set $\ell_g(\pi_g) = |\pi_f|$. The machine M_g works as follows: it first uses its advice to obtain the number $m = h_g(|\pi_f|)$ and checks whether $a_1, \ldots, a_m$ are pairwise distinct and for each $i = 1, \ldots, m$, the string w_i is an accepting computation of M_A on input a_i. If all these simulations succeed, then we know that $A \cap \Sigma^{\leq p(n)} = \{a_1, \ldots, a_m\}$. Hence M_g can now simulate M_f on π_f and give correct answers to all oracle queries made in this computation. □

Let us remark that Balcázar and Schöning [1] have shown a similar tradeoff between advice and oracle access in complexity theory: $\mathsf{coNP} \subseteq \mathsf{NP/log}$ if and only if $\mathsf{coNP} \subseteq \mathsf{NP}^S$ for some sparse $S \in \mathsf{NP}$. We complete the picture by showing that the simulations in the previous theorem cannot be strengthened to a full equivalence between the two concepts:

Proposition 4. *There exist propositional proof systems with constant advice which cannot be computed with access to a recursive oracle.*

Proof. Let f be a polynomial-time computable propositional proof system. With each infinite sequence $a = (a_i)_{i \in \mathbb{N}}$, $a_i \in \{0, 1\}$, we associate the system

$$f_a(\pi) = \begin{cases} f(\pi') & \text{if either } \pi = 0\pi' \text{ or } (\pi = 1\pi' \text{ and } a_{|\pi|} = 0) \\ \text{undefined} & \text{if } \pi = 1\pi' \text{ and } a_{|\pi|} = 1. \end{cases}$$

As different sequences a and b yield different proof systems f_a and f_b, there exist uncountably many different propositional proof systems with one bit of advice. On the other hand, there are only countably many proof systems computed by oracle Turing machines under recursive oracles. Hence the claim follows. □

3 Optimal Proof Systems

A propositional proof system which simulates every other propositional proof system is called *optimal.* While in the classical setting, the existence of optimal proof systems is a prominent open question [8], Cook and Krajíček [5] have shown that there exists a propositional proof system with one bit of input advice which simulates all classical Cook-Reckhow proof systems. Combining this result with Theorem 3 yields:

Corollary 5. *There exists a propositional proof system f which simulates every polynomial-time computable propositional proof system. The system f is computable in polynomial time under a sparse* NP*-oracle.*

Our next result shows that if advice does not help to shorten proofs even for easy languages, then optimal propositional proof systems exist.

Theorem 6. *If every polynomially bounded proof system with logarithmic output advice for some $L \in \mathsf{coNP}$ can be simulated by a proof system without advice, then the class of all polynomial-time computable propositional proof systems contains an optimal system.*

Proof. The classical Cook-Reckhow Theorem characterizes the existence of polynomially bounded proof systems: a language L has a polynomially bounded poly-time computable proof system if and only if $L \in \mathsf{NP}$. This result also holds in the presence of advice (cf. [5, 2]): L has a polynomially bounded proof system with logarithmic output advice if and only if $L \in \mathsf{NP/log}$. For languages from coNP, this equivalence even holds for input instead of output advice [2]. Therefore, we can restate the hypothesis of the theorem as $(\mathsf{coNP} \cap \mathsf{NP/log}) \subseteq \mathsf{NP}$.

Now we can apply a result of Balcázar and Schöning [1] who have shown that $(\mathsf{coNP} \cap \mathsf{NP/log}) \subseteq \mathsf{NP}$ holds if and only if $\mathsf{NE} = \mathsf{coNE}$. The latter condition, however, is known to imply the existence of optimal propositional proof systems in the classical sense, as shown by Krajíček and Pudlák [8] (cf. also [7]). □

Let us remark that the hypothesis in Theorem 6 does not refer to TAUT, but only to some of its subsets which are easy to prove with the help of advice.

4 Simplifying the Advice

There are two natural ways to enhance proof systems with advice by either supplying non-uniform information to the proof (input advice) or to the proven formula (output advice). Intuitively, input advice is the stronger model: proofs can be quite long and formulas of the same size typically require proofs of different size. Hence, supplying advice depending on the proof size is not only more flexible, but also results in more advice per formula. This view is also supported by previous results: there exist optimal proof systems with input advice [5, 2], whereas for output advice a similar result cannot be obtained by current techniques [3]. Further evidence is provided by the existence of languages that have polynomially bounded proof systems with logarithmic input advice, but do not have such systems with output advice [2].

In our next result we show how input advice can be transformed into output advice. We obtain this simplification of advice under the assumption of weak, but non-trivial upper bounds to the proof size. More precisely, from a proof system which uses logarithmic input advice and has sub-exponential size proofs of all tautologies, we construct a system with polynomial output advice which obeys almost the same upper bounds. For the proof we use a new technique by Buhrman and Hitchcock [4] who show that sets of sub-exponential density are not NP-hard unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$.

Theorem 7. *Let $t(n) \in 2^{O(\sqrt{n})}$ and assume that there exists a $t(n)$-bounded propositional proof system f with polylogarithmic input advice. Then there exists an $s(n)$-bounded propositional proof system g with polynomial output advice where $s(n) \in O(t(d \cdot n^2))$ with some fixed constant d independent of f.*

Proof. Let $t(n) \leq 2^{c\cdot\sqrt{n}}$ for some constant c and let f be a $t(n)$-bounded propositional proof system with polylogarithmic input advice. We say that π is a *conjunctive f-proof* for a tautology φ if there exist tautologies $\psi_1, \dots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \dots \wedge \psi_n$ and φ is among the ψ_i. For a number $m \geq 1$, we denote by $\sharp^n_m$ the number of tautologies $\varphi \in \mathrm{TAUT}^{=n}$ which have conjunctive f-proofs of size exactly m. By counting we obtain

$$(\sharp^n_m)^n \geq |\{(\varphi_1, \dots, \varphi_n) \mid \varphi_1 \wedge \dots \wedge \varphi_n \text{ has an } f\text{-proof of size } m \text{ and } |\varphi_i| = n \text{ for } 1 \leq i \leq n\}| \ . \tag{1}$$

As f is t-bounded, every $\varphi \in \mathrm{TAUT}^{=n}$ has a conjunctive f-proof of size at most $t(d \cdot n^2)$ where d is a constant such that for each sequence $\psi_1, \dots, \psi_n$ of formulas of length n, $|\psi_1 \wedge \dots \wedge \psi_n| \leq d \cdot n^2$. Let $\sharp^n = \max\{\sharp^n_m \mid m \leq t(d \cdot n^2)\}$. Using (1) we obtain

$$\begin{aligned} |\mathrm{TAUT}^{=n}|^n &\leq \sum_{m=1}^{t(d\cdot n^2)} (\sharp^n_m)^n \ \leq \ (\sharp^n)^n \cdot t(d \cdot n^2) \\ &\leq (\sharp^n)^n \cdot 2^{c\cdot\sqrt{d\cdot n^2}} \ = \ (\sharp^n \cdot 2^{c\cdot\sqrt{d}})^n \ . \end{aligned}$$

Thus, setting $\delta = 2^{-c\cdot\sqrt{d}}$, we get $\sharp^n \geq \delta \cdot |\mathrm{TAUT}^{=n}|$. Therefore, by definition of $\sharp^n$ there exists a number $m_{n,0} \leq t(d \cdot n^2)$ such that $\sharp^n_{m_{n,0}} \geq \delta \cdot |\mathrm{TAUT}^{=n}|$, i.e., a δ-fraction of all tautologies of length n has a conjunctive f-proof of size $m_{n,0}$.

Consider now the set $\mathrm{TAUT}^{=n}_0$ of all tautologies of length n which do not have conjunctive f-proofs of size $m_{n,0}$. Repeating the above argument for $\mathrm{TAUT}^{=n}_0$ yields a proof length $m_{n,1}$ such that $\sharp^n_{m_{n,1}} \geq \delta \cdot |\mathrm{TAUT}^{=n}_0|$. Iterating this argument we obtain a sequence

$$m_{n,0}, m_{n,1}, \dots, m_{n,\ell(n)} \qquad \text{with } \ell(n) = \left\lceil \frac{\log |\mathrm{TAUT}^{=n}|}{\log(1-\delta)^{-1}} \right\rceil \leq \left\lceil \frac{n}{\log(1-\delta)^{-1}} \right\rceil$$

such that every $\varphi \in \mathrm{TAUT}^{=n}$ has a conjunctive f proof of size $m_{n,i}$ for some $i \in \{0, \dots, \ell(n)\}$.

We will now define a proof system g which uses polynomial output advice and obeys the same proof lengths as f. Assume that f is computed by the polynomial-time Turing transducer M_f with advice function h_f. The system g will be computed by a polynomial-time Turing transducer M_g using the advice function

$$h_g(n) = \left\langle m_{n,0}, h_f(m_{n,0}), \dots, m_{n,\ell(n)}, h_f(m_{n,\ell(n)}) \right\rangle \ .$$

The machine M_g works as follows: first M_g checks whether the proof has the form

$$\langle \varphi, \psi_1, \dots, \psi_n, \pi, i \rangle$$

where $\varphi, \psi_1, \dots, \psi_n$ are formulas of length n such that $\varphi \in \{\psi_1, \dots, \psi_n\}$, π is a string (an f-proof), and i is a number $\leq \ell(n)$. If this test fails, then M_g outputs $\top^n$ (an easy tautology of length n). Then M_g uses its advice to check whether $|\pi| = m_{n,i}$. If so, then M_g simulates M_f on input π using advice $h_f(m_{n,i})$ (which is available through the advice function h_g). If the output of this simulation is $\psi_1 \wedge \dots \wedge \psi_n$, then M_g outputs φ, otherwise it outputs $\top^n$.

By our analysis above, g is a propositional proof system (it is correct and complete). The advice only depends on the length n of the proven formula, so g uses output advice. To estimate the advice length, let $|h_f(m)| \leq \log^a m$ for some constant a. Then

$$|h_g(n)| \leq \sum_{i=0}^{\ell(n)} (|m_{n,i}| + |h(m_{n,i}|) \leq (\ell(n)+1)\left(c\sqrt{n} + \log^a(2^{c\sqrt{n}})\right) \in n^{O(1)} \ .$$

The size of a g-proof $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ for $\varphi \in \mathrm{TAUT}^{=n}$ is dominated by $|\pi| \leq t(d \cdot n^2)$, and therefore g is $s(n)$-bounded for some $s(n) \in O(t(d \cdot n^2))$. □

5 Conclusion

Does advice help to prove propositional tautologies? In this generality, we leave open the question – but our results provide partial answers. On the one hand, when proving tautologies "very complicated" advice is not necessary – it suffices to use a "small amount of simple" advice (Theorem 3). Further, if advice is helpful to prove tautologies in the sense that proofs become shorter in general, then again the advice can be simplified (Theorem 7).

On the other hand, if advice is not at all useful to prove tautologies, then optimal propositional proof systems exist (Theorem 6), a consequence which is considered unlikely by many researchers (cf. [7]). For further research, it seems interesting to explore how natural proof systems like resolution can facilitate advice. Is it possible to shorten proofs in such systems by using advice?

Acknowledgement. The first author wishes to thank Uwe Schöning for suggesting to apply results from [1] in the context of proof systems with advice.

References

1. J. Balcázar and U. Schöning. Logarithmic advice classes. *Theoretical Computer Science*, 99:279–290, 1992.
2. O. Beyersdorff, J. Köbler, and S. Müller. Nondeterministic instance complexity and proof systems with advice. In *Proc. 3rd International Conference on Language and Automata Theory and Applications*, volume 5457 of *Lecture Notes in Computer Science*, pages 164 – 175. Springer-Verlag, Berlin Heidelberg, 2009.
3. O. Beyersdorff and S. Müller. A tight Karp-Lipton collapse result in bounded arithmetic. In *Proc. 17th Annual Conference on Computer Science Logic*, volume 5213 of *Lecture Notes in Computer Science*, pages 199 – 214. Springer-Verlag, Berlin Heidelberg, 2008.
4. H. Buhrman and J. M. Hitchcock. NP-hard sets are exponentially dense unless coNP $\subseteq$ NP/poly. In *Proc. 23rd Annual IEEE Conference on Computational Complexity*, pages 1–7, 2008.
5. S. A. Cook and J. Krajíček. Consequences of the provability of NP $\subseteq$ P/poly. *The Journal of Symbolic Logic*, 72(4):1353–1371, 2007.
6. S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.
7. J. Köbler, J. Messner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184(1):71–92, 2003.
8. J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, 1989.