A Tight Karp-Lipton Collapse Result in Bounded Arithmetic

OLAF BEYERSDORFF
Institut für Theoretische Informatik, Leibniz-Universität Hannover, Germany
and
SEBASTIAN MÜLLER
Institut für Informatik, Humboldt-Universität zu Berlin, Germany

Cook and Krajíček have recently obtained the following Karp-Lipton collapse result in bounded arithmetic: if the theory PV proves $\mathsf{NP} \subseteq \mathsf{P}/poly$, then the polynomial hierarchy collapses to the Boolean hierarchy, and this collapse is provable in PV. Here we show the converse implication, thus answering an open question posed by Cook and Krajíček. We obtain this result by formalizing in PV a hard/easy argument of Buhrman, Chang, and Fortnow.

In addition, we continue the investigation of propositional proof systems using advice, initiated by Cook and Krajíček. In particular, we obtain several optimality results for proof systems using advice. We further show that these optimal systems are equivalent to natural extensions of Frege systems.

Categories and Subject Descriptors: F.1.3 [**Complexity Measures and Classes**]: Relations among Complexity Classes; F.2.2 [**Nonnumerical Algorithms and Problems**]: Complexity of proof procedures; F.4.1 [**Mathematical Logic**]: Computational Logic

General Terms: Theory

Additional Key Words and Phrases: Karp-Lipton Theorem, Advice, Optimal Propositional Proof Systems, Bounded Arithmetic, Extended Frege

1. INTRODUCTION

The classical Karp-Lipton Theorem states that $\mathsf{NP} \subseteq \mathsf{P}/poly$ implies a collapse of the polynomial hierarchy PH to its second level [Karp and Lipton 1980]. Subsequently, these collapse consequences have been improved by Köbler and Watanabe [1998] to $\mathsf{ZPP}^{\mathsf{NP}}$ and by Sengupta and Cai to $\mathsf{S}_2^{\mathsf{p}}$ (cf. [Cai 2007]). This currently forms the strongest known collapse result of this kind.

Recently, Cook and Krajíček [2007] have considered the question which collapse consequences can be obtained if the assumption $\mathsf{NP} \subseteq \mathsf{P}/poly$ is provable in some weak arithmetic theory. This assumption seems to be stronger than in the classical

Email addresses: beyersdorff@thi.uni-hannover.de, smueller@informatik.hu-berlin.de

This research was supported by DFG grants KO 1053/5-1 and KO 1053/5-2.

An extended abstract of this article appeared in the *Proceedings of the 17th Annual Conference on Computer Science Logic (CSL)*. Lecture Notes in Computer Science, vol. 5213. Springer-Verlag, Berlin Heidelberg, 199 – 214.

Karp-Lipton results, because in addition to the inclusion $\mathsf{NP} \subseteq \mathsf{P}/poly$ we require an easy proof for it. In particular, Cook and Krajíček showed that if $\mathsf{NP} \subseteq \mathsf{P}/poly$ is provable in PV, then PH collapses to the Boolean hierarchy BH, and this collapse is provable in PV. For stronger theories, the collapse consequences become weaker. Namely, if PV is replaced by S^1_2, then $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}[O(\log n)]}$, and for S^2_2 one gets $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ [Cook and Krajíček 2007]. Still all these consequences are presumably stronger than in Sengupta's result above, because $\mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{S}^{\mathsf{p}}_2$.

Cook and Krajíček [2007] ask whether under the above assumptions, their collapse consequences for PH are optimal in the sense that also the converse implications hold. In this paper we give an affirmative answer to this question for the theory PV. Thus PV proves $\mathsf{NP} \subseteq \mathsf{P}/poly$ if and only if PV proves $\mathsf{PH} \subseteq \mathsf{BH}$. To show this result we use the assertion $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ as an intermediate assumption. Surprisingly, Cook and Krajíček [2007] have shown that provability of this assumption in PV is equivalent to the provability of $\mathsf{NP} \subseteq \mathsf{P}/poly$ in PV. While such a trade-off between nondeterminism and advice seems rather unlikely to hold unconditionally, Buhrman, Chang, and Fortnow [2003] proved that $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ holds if and only if PH collapses to BH. Their proof in [Buhrman et al. 2003] refines the hard/easy argument of Kadin [1988]. We formalize this technique in PV and thus obtain that $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ is provable in PV if and only if PV proves $\mathsf{PH} \subseteq \mathsf{BH}$. Combined with the mentioned results of Cook and Krajíček [2007], this implies that $PV \vdash \mathsf{PH} \subseteq \mathsf{BH}$ is equivalent to $PV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$.

Let us remark that this result can also be obtained in a less direct way by combining results of Zambella [1996] with recent advances of Jeřábek [2008]. The alternative argument proceeds as follows:[1] By a result of Zambella [1996] (cf. Theorem 4.2), if PV proves $\mathsf{PH} \subseteq \mathsf{BH}$, then Buss' hierarchy of arithmetic theories S_2 collapses to PV. Recently, Jeřábek [2008] proved that the assumption $S_2 = PV$ implies that $PV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/O(1)$. Using the above mentioned result by Cook and Krajíček [2007] it follows that $PV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$.

Comparing the two proofs, let us mention that Jeřábek's proof yields a more general result as it also holds for higher levels of the polynomial hierarchy (namely, if T^i_2 proves that the polynomial hierarchy collapses to the Boolean hierarchy over $\Sigma^{\mathsf{p}}_{\mathsf{i}+1}$, then T^i_2 proves $\Sigma^{\mathsf{p}}_{\mathsf{i}+1} \subseteq \Delta^{\mathsf{p}}_{\mathsf{i}+1}/poly$, cf. [Jeřábek 2008]). On the other hand, Jeřábek's result is reached via a new sophisticated and quite elaborate technique called "approximate counting by hashing", whereas our direct proof for the base case $i = 0$ is conceptually much more straightforward (it also uses the mentioned result of Zambella, though).

In addition, using Jeřábek's result one can even obtain the consequence $PV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ under the assumption $PV \vdash \mathsf{PH} \subseteq \Sigma^{\mathsf{p}}_2$, which at first sight seems to be weaker than $PV \vdash \mathsf{PH} \subseteq \mathsf{BH}$. This is so, because Zambella [1996] actually establishes $PV \vdash \mathsf{PH} \subseteq \Sigma^{\mathsf{p}}_2$ as a sufficient condition for the collapse $S_2 = PV$. Thus we conclude that PV proves $\mathsf{PH} \subseteq \mathsf{BH}$ if and only if PV proves $\mathsf{PH} \subseteq \Sigma^{\mathsf{p}}_2$. This is interesting, as such a result is not known to hold without reference to bounded arithmetic.

In summary, combining our results with previous results from [Cook and Krajíček

[1] We are grateful to an anonymous referee of the conference version of this paper for supplying this alternative argument.

2007; Jeřábek 2008; Zambella 1996], we obtain that provability in PV of each of the following four things is equivalent to the other three:

(1) $\mathsf{NP} \subseteq \mathsf{P}/poly$,
(2) $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$,
(3) $\mathsf{PH} = \mathsf{BH}$, and
(4) $\mathsf{PH} = \Sigma_2^p$.

Assumptions of the form $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ play a dominant role in the above Karp-Lipton results. These hypotheses essentially ask whether advice is helpful to decide propositional tautologies. Motivated by this observation, Cook and Krajíček [2007] started to investigate propositional proof systems taking advice. In the second part of this paper we continue this line of research. We give a quite general definition of functional propositional proof systems with advice. Of particular interest are those systems where the advice depends on the proof (input advice) or on the proven formula (output advice).

In our investigation we focus on the question whether there exist optimal proof systems for different advice measures. While the existence of optimal propositional proof systems without advice is a long-standing open question, posed by Krajíček and Pudlák [1989], Cook and Krajíček [2007] proved that there is a system with one bit of input advice which is optimal for all systems using up to logarithmically many advice bits. On the negative side, we show that this cannot be improved to a p-optimality result, where the simulation is computed in polynomial time without usage of advice. To obtain positive results in the spirit of p-optimality, we propose the less restrictive notion of a p-optimal machine, that allows the advice-free simulation of all systems from the respective class by one machine which, however, is allowed to use variable advice. By extending the proof method of Cook and Krajíček [2007], we obtain p-optimal machines for each class of proof systems with super-logarithmic advice.

These optimality results only leave open the question whether the classes of proof systems with constant advice admit p-optimal machines. We prove that for each constant k, there is a machine which p-simulates all systems with k advice bits, but itself uses $k + 1$ bits of advice. We also use a technique of Sadowski [2002] to show that the existence of p-optimal proof systems for SAT_2 implies the existence of p-optimal machines using k advice bits for each constant k.

In contrast to the optimality results for input advice, we show that we cannot expect similar results for proof systems with output advice, unless $\mathsf{PH} \subseteq \mathsf{BH}$ already implies $\mathsf{PH} \subseteq \mathsf{D}^\mathsf{P}$.

Finally, we consider classical proof systems like resolution or Frege systems using advice. We show that the optimal proof systems with advice are equivalent to extensions of Frege systems, thus demonstrating that these optimal proof systems admit a robust and meaningful definition.

2. PRELIMINARIES

Let $\Sigma = \{0, 1\}$. Σ^n denotes the set of strings of length n and $(\Sigma^n)^k$ the set of k-tuples of Σ^n. Let $\pi_i : \bigcup_{n\in\mathbb{N}}(\Sigma^*)^n \to \Sigma^*$ be the projection to the i^{th} string of some finite tuple and let $\pi_i^* : \Sigma^* \to \{0, 1\}$ be the projection to the i^{th} bit of a string.

As usual we enumerate the bits of a string starting with 0 and thus for example $\pi_0^*(a_0a_1a_2) = a_0$.

Let $\langle\cdot\rangle$ be a polynomial-time computable function, mapping tuples of strings to strings. Its inverse will be denoted by *enc*.

2.1 Complexity Classes

We assume familiarity with standard complexity classes (cf. [Balcázar et al. 1988]). In particular, we will need the *Boolean hierarchy* BH which is the closure of NP under union, intersection, and complementation. The levels of BH are denoted BH_k and are inductively defined by $\mathsf{BH}_1 = \mathsf{NP}$ and

$$\mathsf{BH}_{k+1} = \{L_1 \setminus L_2 \mid L_1 \in \mathsf{NP} \text{ and } L_2 \in \mathsf{BH}_k\} \ .$$

The second level BH_2 is also denoted by $\mathsf{D^p}$. The Boolean hierarchy coincides with $\mathsf{P}^{\mathsf{NP}[O(1)]}$, consisting of all languages which can be solved in polynomial time with constantly many queries to an NP-oracle. For each level BH_k it is known that k non-adaptive queries to an NP-oracle suffice, i.e., $\mathsf{BH}_k \subseteq \mathsf{P}_{tt}^{\mathsf{NP}[k]}$ (cf. [Beigel 1991]).

Complete problems BL_k for BH_k are inductively given by $\mathrm{BL}_1 = \mathrm{SAT}$ and

$$\begin{aligned} \mathrm{BL}_{2k} &= \{\langle x_1,\dots,x_{2k}\rangle \mid \langle x_1,\dots,x_{2k-1}\rangle \in \mathrm{BL}_{2k-1} \text{ and } x_{2k} \in \overline{\mathrm{SAT}}\} \\ \mathrm{BL}_{2k+1} &= \{\langle x_1,\dots,x_{2k+1}\rangle \mid \langle x_1,\dots,x_{2k}\rangle \in \mathrm{BL}_{2k} \text{ or } x_{2k+1} \in \mathrm{SAT}\} \ . \end{aligned}$$

Observe that $\langle x_1,\dots,x_k\rangle \in \mathrm{BL}_k$ if and only if there exists an $i \le k$, such that x_i is satisfiable and the largest such i is odd.

Complexity classes with *advice* were first considered by Karp and Lipton [1980]. For each function $h : \mathbb{N} \to \Sigma^*$ and each language L we let

$$L/h = \{x \mid \langle x, h(|x|)\rangle \in L\} \ .$$

If C is a complexity class and F is a class of functions, then $\mathsf{C}/F = \{L/h \mid L \in \mathsf{C},\ h \in F\}$. Usually the family of functions F is defined by some bound on the length of the values in terms of the argument. Thus, for example, $\mathsf{NP}/O(1)$ denotes the class of languages recognized by NP machines with advice functions h where $|h(n)|$ is bounded by a constant (cf. [Balcázar et al. 1988]).

2.2 Propositional Proof Systems

Propositional proof systems were defined in a general way by Cook and Reckhow [1979] as polynomial-time computable functions P which have as their range the set of all tautologies. A string π with $P(\pi) = \varphi$ is called a P-proof of the tautology φ. Equivalently, propositional proof systems can be defined as polynomial-time computable relations $P(\pi,\varphi)$ such that φ is a tautology if and only if $(\exists\pi)P(\pi,\varphi)$ holds. A propositional proof system P is *polynomially bounded* if all tautologies have polynomial size P-proofs.

Proof systems are compared according to their strength by simulations introduced by Cook and Reckhow [1979] and Krajíček and Pudlák [1989]. A proof system S *simulates* a proof system P (denoted by $P \le S$) if there exists a polynomial p such that for all tautologies φ and P-proofs π of φ there is an S-proof π' of φ with $|\pi'| \le p(|\pi|)$. If such a proof π' can even be computed from π in polynomial time we say that S *p-simulates* P and denote this by $P \le_p S$. If the systems P and S

mutually (p-)simulate each other, they are called *(p-)equivalent.* A proof system is called *(p-)optimal* if it (p-)simulates all proof systems.

A prominent class of propositional proof systems is formed by *extended Frege systems EF* which are usual textbook proof systems based on axioms and rules, augmented by the possibility to abbreviate complex formulas by propositional variables to reduce the proof size (cf. [Cook and Reckhow 1979; Krajíček 1995]).

3. REPRESENTING COMPLEXITY CLASSES BY BOUNDED FORMULAS

The relations between computational complexity and bounded arithmetic are rich and varied, and we refer to [Krajíček 1995; Cook and Nguyen 2009] for background information. Here we will use the two-sorted formulation of arithmetic theories [Cook 2005; Cook and Nguyen 2009]. In this setting we have two sorts: numbers and finite sets of numbers, which are interpreted as strings. Number variables will be denoted by lower case letters $x, y, n, \ldots$ and string variables by upper case letters $X, Y, \ldots$ The two-sorted vocabulary includes the symbols $+, \cdot, \leq, 0, 1$, and the function $|X|$ for the length of strings.

Our central arithmetic theory will be the theory VPV, which is the two-sorted analogue of Cook's PV [Cook 1975]. In addition to the above symbols, the language of VPV contains names for all polynomial-time computable functions (where the running time is measured in terms of the length of the inputs with numbers coded in unary). The theory VPV is axiomatized by definitions for all these functions as well as by the number induction scheme for open formulas.

Bounded quantifiers for strings are of the form $(\forall X \leq t)\varphi$ and $(\exists X \leq t)\varphi$, abbreviating $(\forall X)(|X| \leq t \rightarrow \varphi)$ and $(\exists X)(|X| \leq t \wedge \varphi)$, respectively (where t is a number term not containing X). We use similar abbreviations for $=$ instead of $\leq$. By counting alternations of quantifiers, a hierarchy Σ_i^B, Π_i^B of bounded formulas is defined. The first level Σ_1^B contains formulas of the type $(\exists X_1 \leq t_1) \ldots (\exists X_k \leq t_k)\varphi$ with only bounded number quantifiers occurring in φ. Similarly, Π_1^B*-formulas* are of the form $(\forall X_1 \leq t_1) \ldots (\forall X_k \leq t_k)\varphi$.

As we want to investigate the provability of various complexity-theoretic assumptions in arithmetic theories, we need to formalize complexity classes within bounded arithmetic. To this end we associate with each complexity class C a class of arithmetic formulas $\mathcal{F}_{\mathsf{C}}$. The formulas $\mathcal{F}_{\mathsf{C}}$ describe C, in the sense that for each $A \subseteq \Sigma^*$ we have $A \in \mathsf{C}$ if and only if A is definable by an $\mathcal{F}_{\mathsf{C}}$-formula $\varphi(X)$ with a free string variable X.

It is well known that Σ_1^B-formulas describe NP-sets in this sense, and this connection extends to the formula classes Σ_i^B and Π_i^B and the respective levels $\mathsf{\Sigma_i^p}$ and $\mathsf{\Pi_i^p}$ of the polynomial hierarchy. Given this connection, we can model the levels BH_k of the Boolean hierarchy by formulas of the type

$$\varphi_1(X) \wedge \neg(\varphi_2(X) \wedge \ldots \neg(\varphi_{k-1}(X) \wedge \neg\varphi_k(X)) \ldots) \tag{1}$$

with Σ_1^B-formulas $\varphi_1, \ldots, \varphi_k$.

Another way to speak about complexity classes in arithmetic theories is to consider complete problems for the respective classes. For the satisfiability problem SAT we can build an open formula $Sat(T, X)$, stating that T codes a satisfying assignment for the propositional formula coded by X. In VPV we can prove that $(\exists T \leq |X|)\, Sat(T, X)$ is NP-complete, in the sense that every Σ_1^B-formula φ is provably equivalent to $(\exists T \leq t(|X|)\, Sat(T, F_\varphi(X))$ for some polynomial-time computable function F_φ and an appropriate number term t.

Using this fact, we can express the classes BH_k in VPV equivalently as:

LEMMA 3.1. *For every formula φ describing a language from BH_k as in* (1) *there is a polynomial-time computable function $F : \Sigma^* \to (\Sigma^*)^k$ such that VPV proves the equivalence of φ and*

$$\begin{aligned}(\exists T_1, T_3, \ldots, T_{2\cdot\lfloor k/2\rfloor+1} \leq t)(\forall T_2, T_4, \ldots, T_{2\cdot\lfloor k/2\rfloor} \leq t)\\ (\ldots((Sat(T_1, \pi_1(F(X))) \wedge \neg Sat(T_2, \pi_2(F(X))))\\ \vee Sat(T_3, \pi_3(F(X)))) \wedge \cdots \wedge_k \neg^{k+1} Sat(T_k, \pi_k(F(X))))\end{aligned} \tag{2}$$

where $\wedge_k = \wedge$ if k is even and $\vee$ otherwise, $\neg^k = \neg \ldots \neg$ (k-times), and t is a number term bounding $|F(X)|$. We will abbreviate (2) *by $BL_k(F(X))$.*

Similarly, we can define the class $\mathsf{P}_{tt}^{\mathsf{NP}[k]}$ by all formulas of the type

$$\begin{aligned}(\exists T_1 \ldots T_k \leq t)(Sat(T_1, F_1(X)) \wedge \cdots \wedge Sat(T_k, F_k(X)) \wedge \varphi_1(X)) \vee \cdots \vee\\ (\forall T_1 \ldots T_k \leq t)(\neg Sat(T_1, F_1(X)) \wedge \cdots \wedge \neg Sat(T_k, F_k(X)) \wedge \varphi_{2^k}(X))\end{aligned} \tag{3}$$

where $\varphi_1, \ldots, \varphi_{2^k}$ are open formulas, $F_1, \ldots, F_k$ are polynomial-time computable functions, and t is a number term bounding $|F_i(X)|$ for $i = 1, \ldots, k$. In (3), every combination of negated and unnegated *Sat*-formulas appears in the disjunction.

With these arithmetic representations we can prove inclusions between complexity classes in arithmetic theories. Let A and B be complexity classes represented by the formula classes $\mathcal{A}$ and $\mathcal{B}$, respectively. Then we use $VPV \vdash \mathcal{A} \subseteq \mathcal{B}$ to abbreviate that for every formula $\varphi_\mathcal{A} \in \mathcal{A}$ there exists a formula $\varphi_\mathcal{B} \in \mathcal{B}$, such that $VPV \vdash \varphi_\mathcal{A}(X) \leftrightarrow \varphi_\mathcal{B}(X)$.

In the following, we will use the same notation for complexity classes and their respective representations. Hence we can write statements like $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$, with the precise meaning explained above. For example, using Lemma 3.1 it is straightforward to verify:

LEMMA 3.2. *For every number k we have $VPV \vdash \mathsf{BH}_k \subseteq \mathsf{P}_{tt}^{\mathsf{NP}[k]}$.*

Finally, we will consider complexity classes that take advice. Let $\mathcal{A}$ be a class of formulas. For a constant $k \geq 0$, $VPV \vdash \mathcal{A} \subseteq \mathsf{NP}/k$ abbreviates that, for every $\varphi \in \mathcal{A}$ there exist Σ_1^B-formulas $\varphi_1, \ldots, \varphi_{2^k}$, such that

$$VPV \vdash (\forall n) \bigvee_{1\leq i\leq 2^k} (\forall X)\,(|X| = n \to (\varphi(X) \leftrightarrow \varphi_i(X)))\ . \tag{4}$$

Similarly, using the self-reducibility of SAT, we can formalize the assertion $VPV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ as

$$VPV \vdash (\forall n)(\exists C \leq t(n))(\forall X \leq n)(\forall T \leq n)(Sat(T, X) \to Sat(C(X), X))$$

where t is a number term and $C(X)$ is a term expressing the output of the circuit C on input X (cf. [Cook and Krajíček 2007]).

4. THE KARP-LIPTON COLLAPSE RESULT IN VPV

In this section we will prove that, in VPV, the Karp-Lipton collapse $\mathsf{PH} \subseteq \mathsf{BH}$ of Cook and Krajíček [2007] is optimal in the sense that $VPV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ is equivalent to $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$. We will use the following complexity-theoretic theorem of Buhrman, Chang, and Fortnow [2003] to achieve the desired result.

Theorem 4.1 [Buhrman et al. 2003]. *For every constant k we have $\mathsf{coNP} \subseteq \mathsf{NP}/k$ if and only if $\mathsf{PH} \subseteq \mathsf{BH}_{2^k}$.*

While the forward implication of Theorem 4.1 is comparatively easy and was shown to hold relative to VPV by Cook and Krajíček [2007], the backward implication was proven in [Buhrman et al. 2003] by a sophisticated hard/easy argument. In the sequel, we will formalize this argument in VPV, thereby answering a question of Cook and Krajíček [2007], who asked whether $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$ already implies $VPV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.

The assumption of a provable collapse of PH to BH also allows us to use stronger tools in our arguments than are known to be available in VPV.

Theorem 4.2 [Zambella 1996]. *If $PV \vdash \mathsf{PH} \subseteq \mathsf{BH}$, then $PV = S_2$.*

This enables us to use the bounded replacement principle and bounded minimization properties which are presumably not available in VPV. The bounded replacement principle implies that each Σ_i^B class, defined in terms of alternating bounded string quantifiers, is up to provable equivalence closed under bounded number quantifiers (in the paper, we only need this for Σ_1^B, so we only require Σ_1^B or equivalently Σ_0^B replacement, cf. [Cook and Nguyen 2009]). The bounded minimization principle states that if a bounded property is ever satisfied, then there exists a lexicographically minimal satisfying element. We will frequently make use of these principles and their consequences later on.

Assuming $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$, we claim that there is some constant ℓ such that $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_\ell$. This follows, because $\mathsf{PH} \subseteq \mathsf{BH}$ implies $\mathsf{PH} = \mathsf{BH} = \mathsf{\Sigma}_2^{\mathsf{p}}$. Therefore every problem in PH can be reduced to a fixed $\mathsf{\Sigma}_2^{\mathsf{p}}$-complete problem. Since this problem is contained in some level BH_ℓ of BH, it can be reduced to an appropriate BH_ℓ-complete problem as well. Thus $\mathsf{PH} \subseteq \mathsf{BH}_\ell$.

Therefore, BH_ℓ is provably closed under complementation in VPV, i.e., there exists a polynomial-time computable function h such that

$$VPV \vdash BL_\ell(X_1, \dots, X_\ell) \leftrightarrow \neg BL_\ell(h(X_1, \dots, X_\ell)) \ . \tag{5}$$

Given such a function h, we define the notion of a *hard sequence.* This concept was defined by Chang and Kadin [1996] as a generalization of the notion of hard strings from [Kadin 1988]. Hard strings were first used to show that $\mathsf{BH} \subseteq \mathsf{D}^{\mathsf{p}}$ implies a collapse of PH [Kadin 1988].

Definition 4.3. *Let h be a polynomial-time computable function satisfying* (5)*. A sequence $\bar{x} = (x_1, \dots, x_r)$ of strings is a* hard sequence *of order r for length n, if $x_1, \dots, x_r$ are unsatisfiable formulas of length n, and for all $(\ell - r)$-tuples $\bar{u}$ of formulas of length n, the formula $\pi_{\ell-r+i}(h(\bar{u}, \bar{x}))$ is unsatisfiable for each $i = 1, \dots, r$.*

A hard sequence $\bar{x}$ of order r for length n is not extendable *if, for every unsatisfiable formula x of length n the sequence $x^\frown\bar{x}$ is not hard. Finally, a* maximal hard sequence *is a hard sequence of maximal order.*

Maximal hard sequences are obviously not extendable. Note that by definition, the empty sequence is a hard sequence for every length.

To use this definition in VPV, we have to formalize the notions of hard sequences, non-extendable hard sequences, and maximal hard sequences by bounded predicates HS, $NEHS$, and $MaxHS$, respectively.

Definition 4.4. *Let $VPV \vdash BL_\ell(\bar{X}) \leftrightarrow \neg BL_\ell(h(\bar{X}))$ for some polynomial-time computable function h and every ℓ-tuple $\bar{X}$ of strings.*

For r-tuples X of strings of length n, the following predicate $HS_\ell(X;n,r)$ expresses that X is a hard sequence. $HS_\ell(X;n,r)$ is defined as

$$(\forall i < r)(\forall T \leq n)\neg Sat(T, \pi_{i+1}(X)) \wedge$$
$$(\forall U \in (\Sigma^n)^{\ell-r})(\forall i < r)(\forall T \leq n)(\neg Sat(T, \pi_{\ell-r+i+1}(h(U,X)))) \ .$$

Similarly, we formalize non-extendable hard sequences by the following predicate $NEHS_\ell(X;n,r)$, defined as

$$HS_\ell(X;n,r) \wedge (\forall S = n)\neg HS_\ell(S^\frown X; n, r+1) \ .$$

Finally, maximal hard sequences are expressed via the following bounded formula $MaxHS_\ell(X;n,r)$

$$HS_\ell(X;n,r) \wedge (\forall S \in (\Sigma^n)^{r+1})\neg HS_\ell(S; n, r+1) \ .$$

We remark that HS_ℓ is a Π_1^B-predicate, while $NEHS_\ell$ and $MaxHS_\ell$ are Π_2^B-formulas. Maximal hard sequences allow us to define the unsatisfiability of propositional formulas by a Σ_1^B-formula, as stated in the following lemma.

Lemma 4.5. *Let h be a polynomial-time computable function which for some constant ℓ satisfies* (5). *Then VPV proves the formula*

$$(\forall n)(\forall X = n)(\forall r < \ell)(\forall H \in (\Sigma^n)^{\ell-r-1})\ (MaxHS_\ell(H; n, \ell-r-1) \rightarrow$$
$$[(\forall T \leq n)\neg Sat(T,X) \leftrightarrow (\exists T \leq n)(\exists U \in (\Sigma^n)^r) Sat(T, \pi_{r+1}(h(U,X,H)))]) \ .$$

Proof. We will argue in the theory VPV. Let $H \in (\Sigma^n)^{\ell-r-1}$ be given such that $MaxHS(H; n, \ell-r-1)$ is fulfilled. Assume $(\forall T \leq n)\neg Sat(T,X)$. Then

$$(\forall T \leq n)(\forall U \in (\Sigma^n)^r)\neg Sat(T, \pi_{r+1}(h(U,X,H)))$$

implies $\neg NEHS(H; n, \ell-r-1)$, which in turn implies $\neg MaxHS(H; n, \ell-r-1)$. Thus it holds that

$$(\exists T \leq n)(\exists U \in (\Sigma^n)^r) Sat(T, \pi_{r+1}(h(U,X,H))) \ . \tag{6}$$

On the other hand, assume that (6) holds. Then we obtain

$$(\forall T \leq n)\neg Sat(T,X)$$

from $BL_\ell(X_1,\dots,X_\ell) \leftrightarrow \neg BL_\ell(h(X_1,\dots,X_\ell))$ in a straightforward calculation showing that by the maximality of H, the formula X cannot be satisfiable if $\pi_{r+1}(h(U,X,H))$ is. □

By the preceding lemma, given maximal hard sequences we can describe Π_1^B-formulas by Σ_1^B-formulas. Most of the proof of the main theorem (Theorem 4.8) will go into the construction of such sequences. As we want to use a similar technique as in Theorem 4.1, we will now only consider levels of the Boolean hierarchy that are powers of 2. Thus we assume that $\ell = 2^k$ for some k. It will turn out that, assuming $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$, we can construct 2^k Σ_1^B-formulas, whose disjunction decides the elements of a maximal hard sequence as in (4).

Therefore our aim is to give an NP/k definition of a maximal hard sequence. We will give such a definition for a bitwise encoding of a maximal hard sequence below.

Definition 4.6. *Let h be a polynomial-time computable function which for some constant $\ell = 2^k$ satisfies* (5). *We define the predicate $HardSeqBits_{\ell,1}(\langle 1^n, i\rangle)$ by*

$$\begin{aligned}&(\exists H \in (\Sigma^n)^{<\ell})[MaxHS_\ell(H; n, |H|)\wedge\\ &\quad(\forall S \in (\Sigma^n)^{<\ell})(MaxHS_\ell(S; n, |S|) \rightarrow \langle H\rangle \leq_{lex} \langle S\rangle) \wedge \pi_i^*(\langle H\rangle) = 1]\ .\end{aligned}$$

Here $\leq_{lex}$ denotes the lexicographic ordering on the strings.

Analogously we define $HardSeqBits_{\ell,0}(\langle 1^n, i\rangle)$ with a 0 substituted for the 1 in the last line of the above formula.

Informally, $HardSeqBits_{\ell,1}(\langle 1^n, i\rangle)$ holds, if the i^{th} bit of the encoding of the lexically smallest maximal hard sequence for length n is 1. $HardSeqBits_{\ell,0}(\langle 1^n, i\rangle)$ holds, if the i^{th} bit of the encoding of the lexically smallest maximal hard sequence for length n is 0. Observe that we need the Π_2^B minimization principle, as stated after Theorem 4.2, to prove the existence of such a minimal H. Let $s_\ell(n)$ be a number term such that sequences with at most ℓ elements from Σ^n are coded by strings of size $\leq s_\ell(n)$ via the tupling function $\langle\cdot\rangle$. We choose this function in such a way that for all bit positions $1 \leq i, j \leq s_\ell(n)$ we get $|\langle 1^n, i\rangle| = |\langle 1^n, j\rangle|$. Thus the length of $\langle 1^n, i\rangle$ only depends on n.

Lemma 4.7. *For every k, n, and i, if $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$, then*

$$VPV \vdash HardSeqBits_{2^k,1}(\langle 1^n, i\rangle) \in \mathsf{NP}/k\ .$$

The same holds for $HardSeqBits_{2^k,0}$.

Proof. We will only argue for $HardSeqBits_{2^k,1}$ as the proof for $HardSeqBits_{2^k,0}$ follows along the same lines. As $HardSeqBits_{2^k,1}$ is definable by a bounded formula, the assumption $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$ together with Lemma 3.2 guarantees that the predicate $HardSeqBits_{2^k,1}(\langle 1^n, i\rangle)$ is VPV-provably equivalent to the $\mathsf{P}_{tt}^{\mathsf{NP}[2^k]}$-formula

$$\begin{aligned}&(\exists T_1 \dots T_{2^k} \leq t_\ell(n))\\ &[Sat(T_1, F_1(\langle 1^n, i\rangle)) \wedge \dots \wedge Sat(T_{2^k}, F_{2^k}(\langle 1^n, i\rangle)) \wedge \varphi_1(\langle 1^n, i\rangle)]\ \vee\\ &\qquad\qquad\qquad\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (7)\\ &(\forall T_1 \dots T_{2^k} \leq t_\ell(n))\\ &[\neg Sat(T_1, F_1(\langle 1^n, i\rangle)) \wedge \dots \wedge \neg Sat(T_{2^k}, F_{2^k}(\langle 1^n, i\rangle)) \wedge \varphi_{2^{2^k}}(\langle 1^n, i\rangle)]\ ,\end{aligned}$$

for appropriate polynomial-time computable functions $F_1, \dots, F_{2^k}$, open formulas $\varphi_1, \dots, \varphi_{2^{2^k}}$, and the VPV-number term $t_\ell(n) = |F_1(\langle 1^n, 1\rangle)|$. By padding and because $|\langle 1^n, i\rangle|$ is already determined by n, we can choose the F_j in such a way that the size of the formulas $F_j(\langle 1^n, i\rangle)$ is also determined by n. Thus every formula $F_j(\langle 1^n, i\rangle)$ is of size $t_\ell(n)$.

Our goal is to find Σ_1^B-formulas $\psi^0_{HSB,1}, \dots, \psi^{2^k-1}_{HSB,1}$, such that for every n there is a z, such that for every i, $\mathit{HardSeqBits}_{2^k,1}(\langle 1^n, i\rangle) \leftrightarrow \psi^z_{HSB,1}(\langle 1^n, i\rangle)$. Let Φ be the set of all formulas $F_j(\langle 1^n, i\rangle)$ where $i, n \geq 0$ and $1 \leq j \leq 2^k$. Since we have to evaluate the Sat-formulas only for arguments from Φ, by the proof of Lemma 4.5 it suffices to consider only sequences H with elements from Φ. The parameter z in the formulas $\psi^z_{HSB,1}$ will be the order of a maximal hard sequence for length n, if we only allow formulas from Φ in the sequence.

Let now n be given and let H be a tuple of sequences with elements from Φ. Assume further that H contains a hard sequence of order z. Then we can give a Σ_1^B-formula $\psi^z_{HB,1}(\langle 1^n, i\rangle, H)$ that is VPV-equivalent to $\mathit{HardSeqBits}_{2^k,1}(\langle 1^n, i\rangle)$ for every i of suitable length by $\psi^z_{HB,1}(\langle 1^n, i\rangle, H) =_{def}$

$$\begin{aligned}
&(\exists I = 2^k)(\forall i_H \leq |H|)(((|\pi_{i_H}(H)| = z \wedge HS_{2^k}(\pi_{i_H}(H); n, z)) \rightarrow [\\
&\quad(\exists \bar{U} \in (\Sigma^{t_\ell(n)})^{2^k-z-1})(\forall j < 2^k)(\exists T \leq t_\ell(n))[\\
&\quad(\pi^*_j(I) = 1 \rightarrow \mathit{Sat}(T, F_{j+1}(\langle 1^n, i\rangle))) \wedge\\
&\quad(\pi^*_j(I) = 0 \rightarrow \mathit{Sat}(T, \pi_{2^k-z}(h(\bar{U}, F_{j+1}(\langle 1^n, i\rangle), \pi_{i_H}(H))))) \wedge\\
&\quad\varphi_{\ell(I)}(X)]]) \ .
\end{aligned} \tag{8}$$

Here, ℓ is a polynomial-time computable function that takes I to the number of the respective line in (7). I codes the satisfiability of that line, i.e., $\pi^*_j(I) = 1$ if and only if $F_j(\langle 1^n, i\rangle)$ is satisfiable. This is verified in lines 3 and 4 of (8) by a maximal hard sequence. Line 5 then queries the appropriate $\varphi_{\ell(I)}$. The verification is due to Lemma 4.5, because we only consider maximal hard sequences in line 4 (by line 1 and the assumption that z is the proper advice). Observe that HS_{2^k} is Π_1^B and therefore, using replacement, $\psi^z_{HB,1}$ is equivalent to a Σ_1^B-formula. Abusing notation we will identify $\psi^z_{HB,1}$ with its equivalent Σ_1^B-formula.

Due to (8) we will focus on the definition of such a tuple H of sequences, one of which is maximal. First, observe that there are only few possible elements of the sequences to be included in H. Namely, for each n there are just polynomially many propositional formulas coded by the $F_j(\langle 1^n, i\rangle)$. Let $p_F(n)$ be a polynomial bounding this number. Thus, there exist at most $q_F(n) = 2^k \cdot p_F(n)^{2^k}$ sequences of length at most 2^k with elements among the $F_j(\langle 1^n, i\rangle)$. Therefore, even if H contains all such sequences, it will still be polynomial in size. So, we will just give a definition of H that guarantees that H contains every sequence of order less than 2^k. Then H trivially contains every maximal hard sequence.

To this end let $\psi_{all}(H) =_{def}$

$$\begin{aligned}
&(\exists i_\varepsilon \leq |H|)(\varepsilon = \pi_{i_\varepsilon}(H)) \ \wedge\\
&(\forall i_H \leq |H|)(\forall i_F \leq p_F(n))\\
&\quad(|\pi_{i_H}(H)| < 2^k \rightarrow (\exists j \leq |H|)F_{p(i_F)}(\langle 1^n, q(i_F)\rangle)^\frown\pi_{i_H}(H) = \pi_j(H)) \ .
\end{aligned}$$

The formula $\psi_{all}(H)$ states in the first line, that H includes the empty sequence ε. The next two lines ensure that, if some sequence s in H does not have maximal length, then H includes every sequence of the type $F_j(\langle 1^n, i\rangle)^\frown s$. Thus, H contains every sequence of length less than 2^k, in particular every maximal hard sequence. Here, p and q are polynomial-time computable functions, such that $F_{p(i)}(\langle 1^n, q(i)\rangle)$ is an enumeration of

$$F_1(\langle 1^n, 0\rangle), \ldots, F_1(\langle 1^n, s_{2^k}(n)\rangle), \ldots, F_{2^k}(\langle 1^n, 0\rangle), \ldots, F_{2^k}(\langle 1^n, s_{2^k}(n)\rangle) \enspace .$$

By the arguments above, we can define $\mathit{HardSeqBits}_{2^k,1}(\langle 1^n, i\rangle)$ by using $\psi_{all}(H)$ in addition to $\psi^z_{HB,1}(\langle 1^n, i\rangle, H)$. Thus let $\psi^z_{HSB,1}(\langle 1^n, i\rangle) =_{def}$

$$(\exists H \in ((\Sigma^{t_\ell(n)})^{\leq 2^k})^{q_F(n)})\ \psi_{all}(H) \wedge \psi^z_{HB,1}(\langle 1^n, i\rangle, H) \enspace .$$

Then it holds, that

$$VPV \vdash (\forall n) \bigvee_{0 \leq z < 2^k} (\forall i \leq s_{2^k}(n))(\mathit{HardSeqBits}_{2^k,1}(\langle 1^n, i\rangle) \leftrightarrow \psi^z_{HSB,1}(\langle 1^n, i\rangle)))$$

which concludes the proof of the lemma. □

The above lemma provides the appropriate tools to prove the converse implication to the Karp-Lipton collapse result of Cook and Krajíček [2007].

Theorem 4.8. *If $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$, then $VPV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/k$.*

Proof. Assuming $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$, there exists a polynomial-time computable function h, such that for tuples $\bar{X} = (X_1, \ldots, X_{2^k})$ we have $VPV \vdash BL_{2^k}(\bar{X}) \leftrightarrow \neg BL_{2^k}(h(\bar{X}))$. Thus, by Lemma 4.5, given a maximal hard sequence for length n, we can define $(\forall T \leq n)\neg \mathit{Sat}(T, X)$ by a Σ^B_1-formula. In Lemma 4.7 we constructed such a sequence using k bits of advice.

Let $\psi^z_{HSB,1}$ be the Σ^B_1-formula from the proof of Lemma 4.7 and let $\psi^z_{HSB,0}$ be its counterpart coding the zeros of the hard sequence.

By Lemma 4.7 the theory VPV proves the formulas

$$(\forall n) \bigvee_{0 \leq z < 2^k} (\forall i \leq s_{2^k}(n))\, (\mathit{HardSeqBits}_{2^k,j}(\langle 1^n, i\rangle) \leftrightarrow \psi^z_{HSB,j}(\langle 1^n, i\rangle, Y))$$

for $j \in \{0, 1\}$. As in Lemma 4.7, z is the order of a maximal hard sequence for length n. Observe that z, acting as the advice, can be nonuniformly obtained from n.

Provided the right z, there is a Σ^B_1-formula $\mathit{EasyUnSat}_z(X)$ that, for every X of length n, is VPV-equivalent to $(\forall T \leq n)\neg \mathit{Sat}(T, X)$. This is due to Lemma 4.5. The formula $\mathit{EasyUnSat}_z(X)$ is defined as

$$(\exists C \leq s_{2^k}(|X|))\ (\forall i < |C|)[\bigwedge_{j \in \{0,1\}} (\pi^*_i(C) = j \rightarrow \varphi^z_{HSB,j}(\langle 1^{|X|}, i\rangle, Y)) \wedge$$

$$(\exists T \leq |X|)(\exists \bar{U} \in (\Sigma^{|X|})^{2^k - 1 - |enc(C)|})\ \mathit{Sat}(T, \pi_{2^k - |enc(C)|}(h(\bar{U}, X, enc(C))))] \enspace .$$

By the first line of this formula, C is the encoding of some maximal hard sequence. As in Lemma 4.5, C is used to define $\neg \mathit{Sat}$ by a Σ^B_1-formula (second line). Thus, we have

$$VPV \vdash (\forall n) \bigvee_{0 \leq z < 2^k} (\forall X = n)[(\forall T \leq n)\neg \mathit{Sat}(T, X) \leftrightarrow \mathit{EasyUnSat}_z(X)] \enspace .$$

This concludes the proof. □

With this result we can now prove the optimality of the following Karp-Lipton collapse result of Cook and Krajíček [2007]:

THEOREM 4.9 [COOK AND KRAJÍČEK 2007]. *If VPV proves* $\mathsf{NP} \subseteq \mathsf{P}/poly$*, then* $\mathsf{PH} \subseteq \mathsf{BH}$*, and this collapse is provable in VPV.*

To show the converse implication, we use the following surprising trade-off between advice and nondeterminism in VPV:

THEOREM 4.10 [COOK AND KRAJÍČEK 2007]. *$VPV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ if and only if $VPV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.*

We remark that the proof of Theorem 4.10 uses strong witnessing arguments in form of the Herbrand Theorem and the KPT witnessing theorem [Krajíček et al. 1991]. Thus it seems unlikely that a similar result holds without assuming provability of $\mathsf{NP} \subseteq \mathsf{P}/poly$ and $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ in some weak arithmetic theory. Theorem 4.9 can be obtained as a consequence of Theorem 4.10 and a complexity-theoretic proof of $\mathsf{coNP} \subseteq \mathsf{NP}/O(1) \Rightarrow \mathsf{PH} \subseteq \mathsf{BH}$ (cf. [Buhrman et al. 2003; Cook and Krajíček 2007]).

Combining Theorems 4.8, 4.9, and 4.10 we can now state the optimality of the Karp-Lipton collapse $\mathsf{PH} \subseteq \mathsf{BH}$ in VPV.

THEOREM 4.11. *The theory VPV proves $\mathsf{NP} \subseteq \mathsf{P}/poly$ if and only if VPV proves that the polynomial hierarchy collapses to the Boolean hierarchy.*

5. PROPOSITIONAL PROOF SYSTEMS WITH ADVICE

Motivated by the dominant role of the condition $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ in the above arguments (cf. in particular Theorem 4.10 and its proof), Cook and Krajíček [2007] defined propositional proof systems with advice. In contrast to the classical setting of [Cook and Reckhow 1979] where proof systems are computed by deterministic polynomial-time Turing machines, Cook and Krajíček [2007] allow the Turing machines to take advice. Cook and Krajíček [2007] consider this model of computation both in the functional and in the relational setting for propositional proof systems. For both models, different concepts of proof systems with advice arise that not only differ in the amount of advice, but also in the way the advice is used by the proof system. In the following sections we continue the investigation of proof systems with advice of Cook and Krajíček [2007]. Before considering concrete proof systems with advice, two interesting general questions appear in connection with this new computational model: first, whether there exist optimal proof systems with advice and second, whether there exist polynomially bounded proof systems in this model. Here we will concentrate on the first question. Different complexity-theoretic characterizations for the second question have been obtained in [Beyersdorff et al. 2009].

Our general model of computation for functional proof systems with advice is a Turing transducer with several tapes: an input tape containing the proof, possibly several work tapes for the computation of the machine, an output tape where we output the proven formula, and an advice tape containing the advice. We start with a quite general definition for functional proof systems with advice which subsumes the definitions given by Cook and Krajíček [2007].

Definition 5.1. *Let $k : \mathbb{N} \to \mathbb{N}$ be a function on natural numbers. A surjective function $f : \Sigma^* \to \mathrm{TAUT}$ is a* general functional propositional proof system with k bits of advice*, abbreviated general fpps/k, if there exists 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*) *$f(\pi)$ is computable in polynomial time with advice $h(|\ell(\pi)|)$, i.e., for some fixed polynomial-time Turing machine P_f, $f(\pi) = P_f(\pi, h(|\ell(\pi)|))$, 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.* Note that the latter notion is only well-defined if we assume that the length of the output $f(\pi)$ does not depend on the advice. We remark that Cook and Krajíček [2007] defined a more restrictive concept of proof systems with output advice, which they called length-determined functional proof systems.

The notions of (p-)simulations and (p-)optimality are easily generalized to proof systems with advice. For p-simulations we will use polynomial-time computable functions without advice (unless stated otherwise). We say that a proof system f is (p-)optimal for some class $\mathcal{F}$ of proof systems if f (p-)simulates every system in $\mathcal{F}$ and $f \in \mathcal{F}$.

In the next proposition we observe that *fpps*/k with input advice are already as strong as any general *fpps*/k (Definition 5.1).

Proposition 5.2. *Let $k : \mathbb{N} \to \mathbb{N}$ be a monotone function and f be a general fpps/k. Then there exists a functional proof system f' with k bits of input advice such that f and f' are p-equivalent.*

Proof. We choose a polynomial-time computable bijective pairing function $\langle \cdot, \cdot \rangle$ on $\mathbb{N}$ such that $\langle n_1, n_2 \rangle \geq n_1 + n_2$ for all numbers n_1 and n_2. Let f be an *fpps*/k computed by P_f with advice function h and advice selector ℓ. We define a proof system f' with input advice as follows: on input π' of length n the Turing machine $P_{f'}$ first computes the two unique numbers n_1 and n_2 such that $n = \langle n_1, n_2 \rangle$. It then interprets the first n_1 bits $\pi'_1 \dots \pi'_{n_1}$ of π' as an f-proof π and checks whether $\ell(\pi) = 1^{n_2}$. If this is the case, $P_{f'}$ outputs $P_f(\pi)$, otherwise $P_{f'}(\pi') = \top$. Obviously, $P_{f'}(\pi')$ uses advice $h(|\ell(\pi)|) = h(n_2)$ whose length is bounded by $k(n_1) \leq k(n)$. This shows that f' is an *fpps*/k with input advice.

The p-simulation of f by f' is computed by the function $\pi \mapsto \pi' = \pi 1^m$ where $m = \langle |\pi|, |\ell(\pi)| \rangle - |\pi|$. The converse simulation $f' \leq_p f$ is given by

$$\pi' \mapsto \begin{cases} \pi = \pi'_1 \dots \pi'_{n_1} & \text{if } |\pi'| = \langle n_1, n_2 \rangle \text{ and } \ell(\pi) = 1^{n_2} \\ \pi_0 & \text{otherwise} \end{cases}$$

where π_0 is a fixed f-proof of $\top$. □

In the relational setting for propositional proof systems, advice can be easily implemented as follows:

Definition 5.3 [Cook and Krajíček 2007]. *A* propositional proof system with $k(n)$ bits of advice*, abbreviated pps/k, is a relation P such that for all $x \in \Sigma^*$*

we have $x \in \text{TAUT}$ if and only if $(\exists y)P(y,x)$ and P can be decided by a polynomial-time (in $|x|+|y|$) algorithm which uses $k(|x|)$ bits of advice.

As in the classical case without advice, functional proof systems with output advice and relational proof systems with advice are two formulations of the same concept:

PROPOSITION 5.4. *Let $k : \mathbb{N} \to \mathbb{N}$ be any function. Then every fpps/k with output advice is p-equivalent to some pps/k. Conversely, every pps/k is p-equivalent to an fpps/k with output advice.*

PROOF. For the left-to-right direction, let f be an *fpps*/k with output advice and let M_f be a deterministic polynomial-time Turing machine that computes f using the advice function h. Define a Turing machine M that computes a relational proof system as follows. On input (π, φ) the machine M uses the advice $h(|\varphi|)$ and determines whether $f(\pi) = \varphi$. If true, then M accepts, otherwise M rejects.

For the converse direction, let P be a *pps*/k. We define an *fpps*/k with output advice as follows. Let

$$f(\pi, \varphi) = \begin{cases} \varphi & \text{if } P(\pi,\varphi) \text{ holds} \\ \top^{|\varphi|} & \text{otherwise,} \end{cases}$$

where $\top^n$ denotes a fixed tautology of length n which can be computed in polynomial-time from 1^n. Such a sequence $\top^n$ can be constructed by appropriately padding an easy tautology (e.g. $\top$). □

As in the classical theorem of Cook and Reckhow [1979], Cook and Krajíček [2007] showed the following equivalence:

THEOREM 5.5 [COOK AND KRAJÍČEK 2007]. *Let k be any function. Then there exists a polynomially bounded fpps/k with output advice if and only if $\text{TAUT} \in \mathsf{NP}/k$.*

In the case of polynomial, logarithmic, or constant advice we can formulate Theorem 5.5 as follows.

COROLLARY 5.6. *Let F be a class of functions on $\mathbb{N}$ such that for each $f \in F$ and each polynomial p, $f \circ p \in F$. Then there exists a polynomially bounded fpps/F with output advice if and only if $\mathsf{coNP} \subseteq \mathsf{NP}/F$.*

PROOF. For the left-to-right direction, let f be an *fpps*/k with output advice and polynomial bound p. Guessing at input φ an f-proof π of size $\le p(|\varphi|)$ and verifying $f(\pi) = \varphi$ yields a nondeterministic polynomial-time algorithm for TAUT which uses $k(|\varphi|)$ bits of advice. As TAUT is coNP-complete, every language L in coNP can be reduced to TAUT by a length-respecting polynomial-time reduction t (i.e. if $|x_1| = |x_2|$, then also $|t(x_1)| = |t(x_2)|$). Therefore L has a nondeterministic polynomial-time algorithm which uses $k(q(n))$ bits of advice, where $q(n) = |t(1^n)|$ is the length increase of the reduction t. As $k \in F$ there is another function $k' \in F$ such that $k(q(n)) = k'(n)$ for all n. Thus $\mathsf{coNP} \subseteq \mathsf{NP}/F$.

Conversely, an NP/k-procedure M for TAUT immediately gives a polynomially bounded *fpps*/k with output advice, where proofs are accepting paths of the machine M. □

6. OPTIMAL PROOF SYSTEMS WITH ADVICE

In this section we will investigate the question whether there exist optimal or p-optimal propositional proof systems with advice. With respect to p-optimality, there are two natural options how to define this concept in the presence of advice: we may also allow the simulation functions to take advice or we can consider advice-free simulations. With respect to the first option, a strong positive result was shown by Cook and Krajíček [2007].

Theorem 6.1 [Cook and Krajíček 2007]. *There exists a functional propositional proof system P with one bit of input advice which p-simulates all functional propositional proof systems with $k(n)$ bits of input advice for $k(n) = O(\log n)$. The p-simulation is computed by a polynomial-time algorithm using $k(n)$ bits of advice.*

In terms of simulations rather than p-simulations this result yields:

Corollary 6.2. *The class of all general fpps/$O(\log n)$ contains an optimal functional proof system with one bit of input advice.*

In the above theorem, in addition to the proof system also the simulation functions are allowed to use advice. Our next result shows that for advice-free simulation functions such a result does not hold.

Proposition 6.3. *Let $k : \mathbb{N} \to \mathbb{N}$ be an arbitrary function. Then there does not exist a general fpps/k which p-simulates every fpps/1 with output advice.*

Proof. Let f be a propositional proof system without advice. We will define an uncountable family of proof systems with one bit of output advice. With each infinite sequence $a = (a_i)_{i \in \mathbb{N}}$ with $a_i \in \{0, 1\}$, we associate the following proof system

$$f_a(\pi) = \begin{cases} f(\pi') & \text{if } \pi = 0\pi' \\ f(\pi') \wedge \top & \text{if } \pi = 1\pi' \text{ and } a_{|f(\pi') \wedge \top|} = 1 \\ f(\pi') \vee \bot & \text{if } \pi = 1\pi' \text{ and } a_{|f(\pi') \vee \bot|} = 0. \end{cases}$$

Because of the first line of its definition, f_a is a complete proof system. Further, f_a uses one bit of output advice, as the length of $f_a(\pi)$ does not depend on the advice bit (because $f(\pi') \wedge \top$ and $f(\pi') \vee \bot$ are of the same length). As all advice bits from the sequence a are coded into the proof system f_a according to lines 2 and 3 of its definition, different sequences a and b also yield different proof systems f_a and f_b. Therefore there exist uncountably many different *fpps*/1 with output advice.

On the other hand, there are only countably many Turing machines which can compute potential p-simulations between proof systems. Simulating two different proof systems f_a and f_b by one fixed proof system g requires two different simulation functions. Hence the claim follows. □

Proposition 6.3 immediately yields that none of the classes of proof systems with advice can have a p-optimal proof system.

Corollary 6.4. *Let $k : \mathbb{N} \to \mathbb{N}$ be a function such that $k(n) > 0$ for infinitely many $n \in \mathbb{N}$. Then the class of all general fpps/k does not contain a p-optimal proof system. Similarly, the class of all fpps/k with output advice does not contain a p-optimal proof system.*

The previous corollary contains strong negative information on the existence of p-optimal proof systems with advice. In order to still obtain positive results in the spirit of p-optimality, we make the following less restrictive definition.

Definition 6.5. *Let $k : \mathbb{N} \to \mathbb{N}$ be any function. Then the class of all general fpps/k has a* p-optimal machine *if there exists a deterministic polynomial-time Turing machine M and a polynomial-time computable advice selector function $\ell : \Sigma^* \to 1^*$ such that for all general fpps/k f there exists an advice function $h : \mathbb{N} \to \Sigma^*$ and a polynomial-time computable function t such that for all $\pi \in \Sigma^*$*

(1) $f(\pi) = M(t(\pi), h(|\ell(t(\pi))|))$ (the p-simulation),

(2) for all $n \in \mathbb{N}$, $|h(n)| \le k(n)$ (the advice bound), and

(3) $M(\pi, h(|\ell(\pi)|)) \in \mathrm{TAUT}$ (the correctness).

Let us provide some motivation for this definition. Proof systems with advice essentially consist of three components: the uniform polynomial-time Turing machine, the uniform advice selector function, and the nonuniform advice. As we cannot control the nonuniform component (which causes the absence of p-optimal proof systems by Proposition 6.3), it makes sense to ask for a p-optimal system where only the uniform part is fixed, but the nonuniform advice remains variable. This constellation is precisely described by the above notion of a p-optimal machine. In the remaining part of this section we will investigate the question whether p-optimal machines exist for several measures of advice.

In the next definition we single out a large class of natural functions which we will use as advice bounds in Theorem 6.7 below.

Definition 6.6. *A monotone function $k : \mathbb{N} \to \mathbb{N}$ is* polynomially monotone *if there exists a polynomial p, such that for each $m, n \in \mathbb{N}$, $m \ge p(n)$ implies $k(m) > k(n)$.*

Monotone polylogarithmic functions and monotone polynomials (non-constant) are examples for polynomially monotone functions. If we consider proof systems with a polynomially monotone amount of advice, then we obtain p-optimal machines for each such class. This is the content of the next theorem which we prove by the same technique as was used for Theorem 6.1.

Theorem 6.7. *Let $k(n)$ be a polynomially monotone function. Then the class of all general fpps/k has a p-optimal machine.*

Proof. Let k be a function as above. Since k is polynomially monotone we can find a polynomial-time computable function $\ell : \Sigma^* \to 1^*$ such that for each $x \in \Sigma^*$ we have $k(|\ell(x)|) \ge k(|x|) + 1$. Moreover, we can choose the function ℓ such that ℓ is injective on lengths, i.e., for all $x, y \in \Sigma^*$, $|\ell(x)| = |\ell(y)|$ implies $|x| = |y|$. Let $\|\cdot\|$ be an encoding of deterministic polynomial-time clocked Turing transducers by natural numbers. Without loss of generality we may assume that every machine M has running time $|x|^{\|M\|}$. Further, we need a polynomial-time computable function $\langle \cdot, \cdot, \cdot \rangle$ mapping triples of $\mathbb{N}$ bijectively to $\mathbb{N}$.

We will construct a polynomial-time Turing machine P which together with the above advice selector function ℓ serves as a p-optimal machine for the class of all general *fpps/k*. Let Q be a system from the class of all general *fpps/k* with advice function h_Q. By Proposition 5.2 we may assume that Q has input advice. First we will define a polynomial-time computable function t_Q translating Q-proofs into P-proofs and then we will describe how P works. We set $t_Q(\pi) = \pi 1^m$ where m is determined from the equation $m + |\pi| = \langle |\pi|, 1^{\|Q\|}, |\pi|^{\|Q\|} \rangle$.

Now we define the machine P: upon input x we first compute the unique numbers m_1, m_2, m_3 such that $|x| = \langle m_1, m_2, m_3 \rangle$. Let $\pi = x_1 \dots x_{m_1}$ be the first m_1 bits of x. Then we determine the machine Q from the encoding $|m_2| = \|Q\|$. By the construction of ℓ, the machine P receives at least one more bit of advice than Q. For the p-simulation of Q, the machine P uses the advice function $h_{P,Q}(|\ell(t_Q(\pi))|) = \mathit{Correct}^\frown h_Q(|\pi|)$, where *Correct* is a bit certifying that under the advice $h_Q(|\pi|)$, the machine Q encoded by $|m_2|$ is indeed a correct propositional proof system on proof length $|\pi|$. Because ℓ is injective on lengths, the bit *Correct* can indeed refer to the correctness of Q on proof length $|\pi|$. Therefore, if the first advice bit of P is 1, P simulates Q on input π for m_3 steps, where it passes the last $k(|\pi|)$ advice bits of P to Q. Otherwise, if the first advice bit of P is 0, P outputs $\top$. Except for the first bit, P receives the same advice as Q. Further, the machine P p-simulates every *fpps*$/k$ Q with input advice via the polynomial-time computable function t_Q. By Proposition 5.2, P also p-simulates every general *fpps*$/k$. Thus, P and ℓ yield a p-optimal machine. □

In a similar way we get:

Proposition 6.8. *For each constant $k \geq 0$ there exists a machine P using $k+1$ bits of input advice that p-simulates every fpps with k bits of input advice.*

Proof. The proof uses the same construction as in the proof of Theorem 6.7 where the last k advice bits of the new machine P are the advice bits for the machine Q which we simulate if the first of the $k+1$ advice bits certifies that Q is correct, i.e., it only produces tautologies. □

Regarding the two previous results there remains the question whether for constant k the class of all general *fpps*$/k$ also has a p-optimal machine with exactly k bits. Going back to the proof of Proposition 6.8, we observe that the machine with $k+1$ advice bits, which p-simulates each *fpps*$/k$, does not really need the full power of these $k+1$ bits, but in fact only needs $2^k + 1$ different advice strings. Assuming the existence of a p-optimal proof system without advice, we can manage to reduce the amount of the necessary advice to exactly k bits, thus obtaining a p-optimal machine for the class of all general *fpps*$/k$.

Theorem 6.9. *Assume that there exists a p-optimal proof system. Then for each constant $k \geq 1$ the class of all general fpps$/k$ has a p-optimal machine.*

Proof. Sadowski [2002] proved that the existence of p-optimal propositional proof systems can be characterized as follows:

> *There exists a p-optimal proof system if and only if there exists a recursive enumeration M_i, $i \in \mathbb{N}$, of deterministic polynomial-time clocked Turing machines such that*
> (*1*) *for every $i \in \mathbb{N}$ we have $L(M_i) \subseteq$ TAUT and*
> (*2*) *for every polynomial-time decidable subset $L \subseteq$ TAUT there exists an index i such that $L \subseteq L(M_i)$.*

Assume now that M_i is an enumeration of the easy subsets of TAUT as above. For every proof system Q with k bits of input advice we construct a sequence of propositional formulas

$$\mathit{Prf}^{Q}_{m,n,k}(\pi, \varphi, a) \ ,$$

asserting that the computation of Q at input π of length m leads to the output φ of length n under the k advice bits of a. We also choose a propositional formula $\mathit{Taut}_n(\varphi)$ stating that the formula encoded by φ is a propositional tautology. As Q is an *fpps*/k, the formulas

$$\mathit{Correct}^{Q}_{m,n,k} = (\exists a)(\forall \pi, \varphi) \left(\mathit{Prf}^{Q}_{m,n,k}(\pi, \varphi, a) \rightarrow \mathit{Taut}_n(\varphi) \right)$$

are true quantified Boolean formulas for every $n, m \geq 0$. Since the advice length k is a constant, the quantifier $(\exists a)$ can be replaced by a constant-size disjunction, making it Π^q_1; by prenexing and stripping the universal quantifiers, we obtain a usual Boolean formula. Because the resulting formulas can be constructed in polynomial time from Q, there exists an index $i \in \mathbb{N}$ such that M_i accepts the set of propositional translations of $\{\mathit{Correct}^{Q}_{m,n,k} \mid m, n \geq 0\}$.

Now we construct a p-optimal machine P with k advice bits as follows: at input x we compute the unique numbers $m_1, \dots, m_5$ such that $|x| = \langle m_1, \dots, m_5 \rangle$. As in the proof of Theorem 6.7, we set $\pi = x_1 \dots x_{m_1}$ and $\|Q\| = m_2$. The machine P then simulates $Q(\pi)$ with its own k advice bits for m_3 steps. If the simulation does not terminate, then P outputs $\top$. Otherwise, let φ be the output of this simulation. But before also P can output φ, we have to check the correctness of Q for the respective input and output length. To do this, P simulates the machine M_{m_4} on input $\mathit{Correct}^{Q}_{m_1,|\varphi|,k}$ for at most m_5 steps. If M_{m_4} accepts, then we output φ, and $\top$ otherwise.

The advice which P receives is the correct advice for Q, in case that M_{m_4} certifies that such advice indeed exists. To show the p-optimality of P, let Q be an *fpps*/k with input advice and let M_i be the machine accepting $\{\mathit{Correct}^{Q}_{m,n,k} \mid m, n \geq 0\}$. Then the system Q is p-simulated by the machine P via the mapping $\pi \mapsto \pi 1^m$ where $m = \langle |\pi|, \|Q\|, p(|\pi|), i, p(\ell) \rangle - |\pi|$, where p is a polynomial bounding the running time of both M_i and Q, and $\ell = \max_{i \leq p(|\pi|)}(|\mathit{Correct}^{Q}_{|\pi|,i,k}|)$. □

All the optimal proof systems and p-optimal machines that we have so far constructed were using input advice. It is a natural question whether we can improve these constructions to obtain proof systems with output advice that still have the same optimality conditions. While we must leave this question open, our next result shows that it seems unlikely to give an affirmative answer with currently available techniques, as otherwise collapse assumptions of presumably different strength would be equivalent. This result indicates that, by current knowledge, input advice for propositional proof systems is indeed a more powerful concept than output advice.

THEOREM 6.10. *Let $k \geq 1$ be a constant and assume that there exists an fpps/k with output advice that simulates every fpps/1. Then the following conditions are equivalent:*

(*1*) *The polynomial hierarchy collapses to BH_{2^k}.*

(*2*) *The polynomial hierarchy collapses to* BH.

(*3*) $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n)$.

(*4*) $\mathsf{coNP} \subseteq \mathsf{NP}/k$.

Proof. The equivalence of 1 and 4 was shown by Buhrman, Chang, and Fortnow (Theorem 4.1), and clearly, item 1 implies item 2. It therefore remains to prove the implications $2 \Rightarrow 3$ and $3 \Rightarrow 4$.

For the implication $2 \Rightarrow 3$, let us assume $\mathsf{PH} \subseteq \mathsf{BH}$. We choose a Σ_2^{p}-complete problem L, which by assumption is contained in $\mathsf{BH}_{k'}$ for some number k'. By Theorem 4.1 this implies $\mathsf{coNP} \subseteq \mathsf{NP}/k'$ and hence $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n)$.

For the final implication $3 \Rightarrow 4$, we assume $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n)$. By Theorem 5.5 this guarantees the existence of a polynomially bounded system P with $O(\log n)$ bits of output advice. By Theorem 6.1, P is simulated by a proof system P' with only one bit of input advice. Hence also P' is polynomially bounded. Now we use the hypothesis of the existence of a functional proof system Q with k bits of output advice which simulates all *fpps*/1. In particular, $P' \leq Q$ and therefore Q is a polynomially bounded *fpps*/k with output advice. Using again Theorem 5.5 we obtain $\mathsf{coNP} \subseteq \mathsf{NP}/k$. □

With respect to the optimal proof system from Corollary 6.2 we obtain:

Corollary 6.11. *The optimal fpps*/1 *from Corollary 6.2 is not equivalent to an fpps*/1 *with output advice, unless* $\mathsf{PH} \subseteq \mathsf{BH}$ *implies* $\mathsf{PH} \subseteq \mathsf{D^p}$.

Of course, rather than indicating that proof systems with constant output advice cannot be optimal for the class of all general *fpps*/$O(\log n)$, this corollary points towards our current limitations to disprove such a result.

7. CLASSICAL PROOF SYSTEMS WITH ADVICE

Let us now outline how one can define classical proof systems that use advice. A priori it is not clear how systems like resolution or Frege can sensibly use advice, but a canonical way to implement advice into them is to enhance these systems by further axioms which can be decided in polynomial time with advice. Cook and Krajíček [Cook and Krajíček 2007] have defined the notion of extended Frege systems using advice. We give a more general definition.

Definition 7.1. *Let* Φ *be a set of tautologies that can be decided in polynomial time with* $k(n)$ *bits of advice. We define the system* $EF + \Phi/k$ *as follows. An* $EF + \Phi/k$*-proof of a formula* φ *is a pair* $\langle \pi, \psi_0 \rangle$, *where* π *is an* EF*-proof of an implication* $\psi \to \varphi$ *and* ψ *is a simple substitution instance of* $\psi_0 \in \Phi$ *(simple substitutions only replace some of the variables by constants).*

If π is an $EF + \Phi/k$-proof of a formula φ, then the advice used for the verification of π neither depends on $|\pi|$ nor on $|\varphi|$, but on the length of the substitution instance ψ from Φ, which is used in π. As $|\psi|$ can be easily determined from π, $EF + \Phi/k$ are systems of the type *fpps*/k (in fact, this was the motivation for our general Definition 5.1).

If we require that the length of ψ in the implication $\psi \to \varphi$ is determined by the length of the proven formula φ, then the advice only depends on the output and hence we get an *fpps*$/k$ with output advice. This is the case for a collection of extensions of *EF* defined by Cook and Krajíček [2007], which are motivated by the proof of Theorem 4.10. Cook and Krajíček proved that these systems, which use constant advice, are polynomially bounded if *VPV* proves $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.

Our next result shows that the optimal proof system from Corollary 6.2 is equivalent to an extended Frege system with advice as in Definition 7.1.

Theorem 7.2. (*1*) *There exists a set* $\Psi \in \mathsf{P}/1$ *such that* $EF + \Psi/1$ *is optimal for the class of all general fpps*$/O(\log n)$.

(*2*) *In contrast, none of the constant advice extensions of EF as defined in [Cook and Krajíček 2007] simulates every general fpps*$/1$, *unless items 1 to 4 from Theorem 6.10 are equivalent.*

Proof. For item 1 we choose the system P using 1 bit of input advice which is optimal for the class of all *fpps*$/O(\log n)$ according to Corollary 6.2. We define the set $\Psi \in \mathsf{P}/1$ as the collection of all formulas

$$RFN^P_{m,n,1} = Prf^P_{m,n,1}(\pi, \varphi, a) \to Taut_n(\varphi)$$

which describe the correctness of P, similarly as in the proof of Theorem 6.9. In contrast to the formulas $Correct^P_{m,n,k}$ from the proof of Theorem 6.9, the correct advice bit a is already substituted into the formula $Prf^P_{m,n,1}(\pi, \varphi, a)$. Therefore, the set

$$\Psi = \{RFN^P_{m,n,1}(\pi, \varphi, a) \mid m, n \geq 0\}$$

is not necessarily in P, but only in $\mathsf{P}/1$.

To show the optimality of $EF + \Psi/1$ it suffices to prove $P \leq EF + \Psi/1$. For this let π be a P-proof of φ. Substituting the propositional encodings of π and φ into $RFN^P_{|\pi|,|\varphi|,1}$, we obtain the formula

$$Prf^P_{m,n,1}(\pi, \varphi, a) \to Taut_n(\varphi) \enspace .$$

Now $Prf^P_{m,n,1}(\pi, \varphi, a)$ is a tautological formula, where all relevant variables have been substituted by constants (only auxiliary variables describing the computation of P remain free, but these variables are determined by π). Therefore, we can derive $Prf^P_{m,n,1}(\pi, \varphi, a)$ in a polynomial-size *EF*-proof, and modus ponens yields $Taut_n(\varphi)$. By induction on the formula φ, we can devise polynomial-size *EF*-proofs of

$$Taut_n(\varphi) \to \varphi \enspace .$$

Hence one further application of modus ponens gives the formula φ, and thus we have constructed a polynomial-size $EF + \Psi/1$-proof of φ.

As the extensions of *EF* defined by Cook and Krajíček [2007] use a constant amount of output advice, the second item follows by Theorem 6.10. □

Comparing the definition of *EF* with advice from [Cook and Krajíček 2007] with our Definition 7.1, we remark that both definitions are parametrized by a set of tautologies Φ, and hence they both lead to a whole class of proof systems rather than *the* extended Frege system with advice. The drawback of our Definition 7.1 is, that even in the base case, where no advice is used, we do not get *EF*, but again all extensions $EF + \Phi$ with polynomial-time computable $\Phi \subseteq \mathrm{TAUT}$. It is known that each advice-free propositional proof system is p-simulated by such an extension of EF [Krajíček 1995]. In contrast, Cook and Krajíček's extended Frege systems with advice lead exactly to EF, if no advice is used. On the other hand, even with advice these systems appear to be strictly weaker than the systems from Definition 7.1, as indicated by item 2 of Theorem 7.2.

Finally, we will outline how other classical proof systems like resolution can be equipped with advice. Let $\Phi = \{\varphi_n \mid n \geq 0\}$ be a sequence of tautologies in conjunctive normal form. Then φ_n can be written as a set of clauses Δ_n. Assume further that Φ can be decided in polynomial time with $k(n)$ bits of advice. A *resolution system with advice* $Res + \Phi$ is then defined as follows: Let ψ be a formula in disjunctive normal form and let Γ be the set of clauses for $\neg\psi$. A $Res + \Phi$-proof of ψ is a resolution refutation of the set $\Delta \cup \Gamma$ where Δ is some simple substitution instance of Δ_n for some n.

8. DISCUSSION AND OPEN PROBLEMS

In this paper we have shown that $\mathsf{PH} \subseteq \mathsf{BH}$ is the optimal Karp-Lipton collapse within the theory PV. It remains as an open problem whether also $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}[O(\log n)]}$ and $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ are optimal within S^1_2 and S^2_2, respectively (cf. [Cook and Krajíček 2007]). For S^1_2 this corresponds to the problem whether $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n)$ is equivalent to $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}[O(\log n)]}$. Buhrman, Chang, and Fortnow [2003] conjecture $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n) \iff \mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ (cf. also [Fortnow and Klivans 2005]). This seems unlikely, as Cook and Krajíček [2007] noted that $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n)$ implies $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}[O(\log n)]}$. However, it does not seem possible to extend the technique from [Buhrman et al. 2003] to prove the converse implication. Is even $\mathsf{coNP} \subseteq \mathsf{NP}/poly \iff \mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ true, possibly with the stronger hypothesis that both inclusions are provable in S^2_2? Currently, $\mathsf{coNP} \subseteq \mathsf{NP}/poly$ is only known to imply $\mathsf{PH} \subseteq \mathsf{S}_2^{\mathsf{NP}}$ [Cai et al. 2005].

With respect to the proof systems with advice we remark that all advice information we have used for our optimal systems in Sects. 6 and 7 can be decided in coNP. It would be interesting to know whether we can obtain stronger proof systems by using more complicated advice.

ACKNOWLEDGMENTS

We are grateful to Emil Jeřábek, Johannes Köbler, Jan Krajíček, and the anonymous referees of both the conference and the journal version for very helpful comments and detailed suggestions on how to improve this paper.

REFERENCES

Balcázar, J. L., Díaz, J., and Gabarró, J. 1988. *Structural Complexity I.* Springer-Verlag, Berlin Heidelberg.

Beigel, R. 1991. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science 84*, 199–223.

Beyersdorff, O., Köbler, J., and Müller, S. 2009. Nondeterministic instance complexity and proof systems with advice. In *Proc. 3rd International Conference on Language and Automata Theory and Applications.* Lecture Notes in Computer Science, vol. 5457. Springer-Verlag, Berlin Heidelberg, 164 – 175.

Buhrman, H., Chang, R., and Fortnow, L. 2003. One bit of advice. In *Proc. 20th Symposium on Theoretical Aspects of Computer Science.* Lecture Notes in Computer Science, vol. 2607. Springer-Verlag, Berlin Heidelberg, 547–558.

Cai, J.-Y. 2007. $S_2^p \subseteq ZPP^{NP}$. *Journal of Computer and System Sciences 73,* 1, 25–35.

Cai, J.-Y., Chakaravarthy, V. T., Hemaspaandra, L. A., and Ogihara, M. 2005. Competing provers yield improved Karp-Lipton collapse results. *Information and Computation 198,* 1, 1–23.

Chang, R. and Kadin, J. 1996. The Boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing 25,* 2, 340–354.

Cook, S. A. 1975. Feasibly constructive proofs and the propositional calculus. In *Proc. 7th Annual ACM Symposium on Theory of Computing.* 83–97.

Cook, S. A. 2005. Theories for complexity classes and their propositional translations. In *Complexity of Computations and Proofs*, J. Krajíček, Ed. Quaderni di Matematica, 175–227.

Cook, S. A. and Krajíček, J. 2007. Consequences of the provability of NP $\subseteq$ P/poly. *The Journal of Symbolic Logic 72,* 4, 1353–1371.

Cook, S. A. and Nguyen, P. 2009. *Logical Foundations of Proof Complexity.* Cambridge University Press. To appear.

Cook, S. A. and Reckhow, R. A. 1979. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic 44,* 1, 36–50.

Fortnow, L. and Klivans, A. R. 2005. NP with small advice. In *Proc. 20th Annual IEEE Conference on Computational Complexity.* 228–234.

Jeřábek, E. 2008. Approximate counting by hashing in bounded arithmetic. *Journal of Symbolic Logic*. To appear.

Kadin, J. 1988. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing 17,* 6, 1263–1282.

Karp, R. M. and Lipton, R. J. 1980. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symposium on Theory of Computing.* ACM Press, 302–309.

Köbler, J. and Watanabe, O. 1998. New collapse consequences of NP having small circuits. *SIAM Journal on Computing 28,* 1, 311–324.

Krajíček, J. 1995. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Encyclopedia of Mathematics and Its Applications, vol. 60. Cambridge University Press, Cambridge.

Krajíček, J. and Pudlák, P. 1989. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic 54,* 3, 1063–1079.

Krajíček, J., Pudlák, P., and Takeuti, G. 1991. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic 52*, 143–153.

Sadowski, Z. 2002. On an optimal propositional proof system and the structure of easy subsets of TAUT. *Theoretical Computer Science 288,* 1, 181–193.

Zambella, D. 1996. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic 61,* 3, 942–966.

Received September 2008; revised April 2009; accepted April 2009