A Tight Karp-Lipton Collapse Result in Bounded Arithmetic

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. Cook and Krajíček [9] have obtained the following Karp-Lipton result in bounded arithmetic: if the theory PV proves $\mathsf{NP} \subseteq \mathsf{P}/poly$, then PH collapses to BH, and this collapse is provable in PV. Here we show the converse implication, thus answering an open question from [9]. We obtain this result by formalizing in PV a hard/easy argument of Buhrman, Chang, and Fortnow [3].
In addition, we continue the investigation of propositional proof systems using advice, initiated by Cook and Krajíček [9]. In particular, we obtain several optimal and even p-optimal proof systems using advice. We further show that these p-optimal systems are equivalent to natural extensions of Frege systems.
Keywords: 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 [15]. Subsequently, these collapse consequences have been improved by Köbler and Watanabe [16] to $\mathsf{ZPP}^{\mathsf{NP}}$ and by Sengupta and Cai to $\mathsf{S}_2^{\mathsf{p}}$ (cf. [4]). This currently forms the strongest known collapse result of this kind.

Recently, Cook and Krajíček [9] 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 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_2^1, then $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}[O(\log n)]}$, and for S_2^2 one gets $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ [9]. Still all these consequences are presumably stronger than in Sengupta's result above, because $\mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{S}_2^{\mathsf{p}}$.

In [9] Cook and Krajíček ask whether under the above assumptions, their collapse consequences for PH are optimal in the sense that also the converse

⋆ Supported by DFG grants KO 1053/5-1 and KO 1053/5-2

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 [9] 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 [3] proved that $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$ holds if and only if PH collapses to BH. Their proof in [3] refines the hard/easy argument of Kadin [14]. 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 from [9], this implies that $PV \vdash \mathsf{PH} \subseteq \mathsf{BH}$ is equivalent to $PV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$.

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 [9] 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 [18], we obtain optimal proof systems with input advice for each advice class. Such a result was already obtained by Cook and Krajíček [9], who prove that there is a system with one bit of input advice which is optimal for all systems using up to logarithmically many advice bits. We extend the proof method from [9] to obtain even p-optimal systems with input advice within each class of systems with super-logarithmic advice function.

These optimality results only leave open the question whether the classes of proof systems with constant advice contain p-optimal systems. We prove that for each constant k, there is a proof system which p-simulates all systems with k advice bits, but itself uses $k+1$ bits of advice. We also use a technique of Sadowski [20] to show that the existence of p-optimal proof systems for SAT_2 implies the existence of p-optimal propositional proof systems using k advice bits for each constant k.

In contrast to these 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 Frege systems using advice. We show that our optimal and p-optimal proof systems with advice are p-equivalent to extensions of Frege systems, thus demonstrating that these p-optimal proof systems admit a robust and meaningful definition.

Due to space constraints, a number of proofs is omitted or only briefly sketched in this extended abstract.

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 $\pi_i : (\Sigma^*)^k \to \Sigma^*$ be the projection to the i^{th} string, and let $\pi_i^* : \Sigma^* \to \{0,1\}$ be the projection to the i^{th} bit of a string. Let π^*_{-i} and π_{-i} be projections deleting the i^{th} string from a tuple or the i^{th} bit from a string, respectively. Although we enumerate the bits of a string starting with 0, we will speak of the first bit, the second bit, etc. of a string, and thus for example $\pi_1^*(a_0a_1a_2) = a_0$ and $\pi^*_{-1}(a_0a_1a_2) = a_1a_2$.

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

Complexity Classes. We assume familiarity with standard complexity classes (cf. [1]). In particular, we will need the *Boolean hierarchy* BH which is the closure of NP under the Boolean operations $\cup$, $\cap$, and $^-$. 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. [2]).

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

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

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

Complexity classes with *advice* were first considered by Karp and Lipton [15]. For each function $k : \mathbb{N} \to \Sigma^*$ and each language L we let $L/k = \{x \mid \langle x, k(|x|)\rangle \in L\}$. If C is a complexity class and F is a class of functions, then $\mathsf{C}/F = \{L/k \mid L \in \mathsf{C}, k \in F\}$.

Propositional Proof Systems. Propositional proof systems were defined in a general way by Cook and Reckhow [11] 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 in [11] and [18]. A proof system S *simulates* a proof system P (denoted by $P \leq S$) 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 $|\pi'| \leq p(|\pi|)$. 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 $P \leq_p S$. 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. [11, 17]).

3 Representing Complexity Classes by Bounded Formulas

The relations between computational complexity and bounded arithmetic are rich and varied, and we refer to [17, 10] for background information. Here we will use the two-sorted formulation of arithmetic theories [8, 10]. 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 letter $x, y, n, \ldots$ 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 [7]. 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 $\Sigma_\mathsf{i}^\mathsf{p}$ and $\Pi_\mathsf{i}^\mathsf{p}$ 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 prob-

lem 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 $(\exists T \leq |X|)Sat(T, F_\varphi(X))$ for some polynomial-time computable function F_φ.

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

Lemma 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 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 are 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 1 it is straightforward to verify:

Lemma 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. Then $VPV \vdash \mathcal{A} \subseteq \mathsf{NP}/k$ 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.[9]).

4 The Karp-Lipton Collapse Result in *VPV*

In this section we will prove that the Karp-Lipton collapse $\mathsf{PH} \subseteq \mathsf{BH}$ from [9] is optimal in VPV, in the sense that $VPV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ is equivalent to $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$. For this we will use the following complexity-theoretic result.

Theorem 3 (Buhrman, Chang, Fortnow [3]). *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 3 is comparatively easy, and was shown to hold relative to VPV by Cook and Krajíček [9], the backward implication was proven in [3] by a sophisticated hard/easy argument. In the sequel, we will formalize this argument in VPV, thereby answering a question of Cook and Krajíček [9], who asked whether $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$ already implies $VPV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.

Assuming $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}$, we claim that there is some constant k such that $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_k$. This follows, because $\mathsf{PH} \subseteq \mathsf{BH}$ implies $\mathsf{PH} = \mathsf{BH} = \Sigma_2^{\mathsf{p}}$. Therefore every problem in PH can be reduced to a fixed Σ_2^{p}-complete problem. Since this problem is contained in some level BH_k of BH, it can be reduced to an appropriate BH_k-complete problem as well. Thus $\mathsf{PH} \subseteq \mathsf{BH}_k$.

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

$$VPV \vdash BL_k(X_1, \ldots, X_k) \leftrightarrow \neg BL_k(h(X_1, \ldots, X_k)) \ . \tag{5}$$

Given h, we define the notion of a *hard sequence*. This concept was defined in [6] as a generalization of the notion of hard strings from [14]. Hard strings were first used to show that $\mathsf{BH} \subseteq \mathsf{D}^{\mathsf{p}}$ implies a collapse of PH [14].

Definition 4. *Let h be a function as in* (5)*. A sequence $\bar{x} = (x_1, \ldots, x_r)$ of strings is a* hard sequence *of order r for length n, if for all $i \leq r$, x_i is an unsatisfiable formula of length n, and for all $(k-r)$-tuples $\bar{u}$ of formulas of length n, the formula $\pi_{k-r+i}(h(\bar{u}, \bar{x}))$ is unsatisfiable.*

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 the empty sequence is a hard sequence for every length.*

To use this definition in VPV, we we note that the notion of a maximal hard sequence can be formalized by a bounded predicate $MaxHS$. 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 5. *Assume that h is a polynomial-time computable function which for some constant k satisfies* (5)*. Then VPV proves the formula*

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

By the preceding lemma, given maximal hard sequences, we can describe Π_1^B-formulas by Σ_1^B-formulas. Most part of the proof of the next theorem will go into the construction of such sequences. 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).

Theorem 6. *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 5, given a maximal hard sequence for length n, we can define $(\forall T \leq n)\neg Sat(T, X)$ by a Σ_1^B-formula. Therefore, our aim is to construct such a sequence using k bits of advice.

To this end, for $i > 0$ let $\mathit{HardSeqBits}(1^n, i)$ hold, if and only if the i^{th} bit of the encoding of the lexically shortest maximal hard sequence for length n is 1. *HardSeqBits* can be defined by a bounded predicate.

By the assumption $VPV \vdash \mathsf{PH} \subseteq \mathsf{BH}_{2^k}$ and Lemma 2, there is a formula ψ as in (3), with appropriate polynomial-time computable functions $F_1, \ldots, F_{2^k}$ and open formulas $\varphi_1, \ldots, \varphi_{2^{2^k}}$, such that the predicate $\mathit{HardSeqBits}(X)$ is VPV-provably equivalent to ψ. Without loss of generality, we may assume, that $|F_i(1^n, a)| = |F_j(1^n, b)|$ for all i, j and a, b.

Using ψ we can prove $VPV \vdash \mathit{HardSeqBits} \in \mathsf{NP}/k$ (we omit the details due to space constraints). This means that we can construct Σ_1^B-formulas $\psi^z_{HSB}(X)$ of the form $(\exists Y \leq t)\varphi^z_{HSB}(X, Y)$ with open formulas φ^z_{HSB} for $z = 0, \ldots, 2^k - 1$ such that

$$VPV \vdash (\forall n) \bigvee_{0 \leq z < 2^k} (\forall X = n)\left(\mathit{HardSeqBits}(X) \leftrightarrow (\exists Y \leq t)\varphi^z_{HSB}(X, Y)\right) \ .$$

In this formula, z is the order of a maximal hard sequence for length n. Observe that z, acting as the advice, can be non-uniformly obtained from n.

Provided the right z, there is a Σ_1^B-formula $\mathit{EasyUnSat}_z(X)$ that, for every X of length n, is VPV-equivalent to $(\forall T \leq n)\neg Sat(T, X)$. This formula $\mathit{EasyUnSat}_z(X)$ is defined as

$$\begin{aligned}(\exists C \leq t')\ (\forall i \leq |C|)(\exists Y \leq t)[(\pi^*_{i+1}(C) = 1 \leftrightarrow \varphi^z_{HSB}(1^{|X|}, i, Y)) \\ \wedge (\exists T \leq |X|)(\exists \bar{U} \in (\Sigma^n)^{2^k - 1 - |enc(C)|}) \\ Sat(T, \pi_{2^k - |enc(C)|}(h(\bar{U}, X, enc(C))))]\end{aligned}$$

for an appropriate number term t'. Now, by line 1 of this formula, C is the encoding of some maximal hard sequence. As in Lemma 5, C is used to define $\neg Sat$ by a Σ_1^B-formula (lines 2 and 3). Thus, we have

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

This concludes the proof. □

With this result we can now prove the optimality of the following Karp-Lipton collapse result of Cook and Krajíček [9]:

Theorem 7 (Cook and Krajíček [9]). *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 8 (Cook and Krajíček [9]). $VPV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ *if and only if* $VPV \vdash \mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.

We remark that the proof of Theorem 8 uses strong witnessing arguments in form of the Herbrand Theorem and the KPT witnessing theorem [19]. 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 7 can be obtained as a consequence of Theorem 8 and a complexity-theoretic proof of $\mathsf{coNP} \subseteq \mathsf{NP}/O(1) \Rightarrow \mathsf{PH} \subseteq \mathsf{BH}$ (cf. [3, 9]).

Combining Theorems 6, 7, and 8 we can now state the optimality of the Karp-Lipton collapse $\mathsf{PH} \subseteq \mathsf{BH}$ in VPV.

Corollary 9. *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.*

The backward direction of this result can also be obtained in a less direct way using a recent result of Jeřábek [13]. The argument goes as follows:[3] by results of Zambella [21], $PV \vdash \mathsf{PH} = \mathsf{BH}$ implies $PV = S_2$. The latter, however, implies $PV \vdash \mathsf{NP} \subseteq \mathsf{P}/poly$ by a result of Jeřábek [13].

5 Propositional Proof Systems with Advice

Cook and Krajíček [9] defined propositional proof systems with advice, both in the functional and in the relational setting for 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.

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 in [9].

Definition 10. *Let* $k : \mathbb{N} \to \mathbb{N}$ *be a function on natural numbers. A* general functional propositional proof system with k bits of advice*, abbreviated general fpps/*k*, consists of two functions* f *and* ℓ *such that*

1. $\ell : \Sigma^* \to \{1^n \mid n \geq 0\}$ *is computable in polynomial time.*

[3] We are grateful to an anonymous referee for supplying this alternative argument.

2. $f : \Sigma^* \to \text{TAUT}$ *is a surjective polynomial-time computable function which on input* π *uses* $k(|\ell(\pi)|)$ *bits of advice depending only on* $|\ell(\pi)|$.

Let us give some explanation for this definition. For each length n there is a unique advice string of length $k(n)$. Which of these strings is used at a particular computation of f is determined by the function ℓ which computes from the input π the relevant advice length. In the functional definition of propositional proof systems, there are two natural options for this function ℓ: the advice may depend on the length of the input (i.e. the proof) or the length of the output (i.e. the proven formula).

Definition 11. *Let* (f, ℓ) *be a general fpps/*k *using advice function* $k(n)$.

1. *We say that* f *has* input advice *if for all inputs* π *we have* $\ell(\pi) = 1^{|\pi|}$, *i.e., the proof system* f *uses* $k(|\pi|)$ *bits of advice.*
2. f *has* output advice *if for all inputs* π, *the length of the output* $f(\pi)$ *does not depend on the advice (i.e., the content of the advice tape) and we have* $\ell(\pi) = 1^{|f(\pi)|}$, *i.e., the proof system* f *uses* $k(|f(\pi)|)$ *bits of advice.*

We remark that Cook and Krajíček [9] 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 advice 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 10).

Proposition 12. *Let* $k : \mathbb{N} \to \mathbb{N}$ *be a monotone function and let* (f, ℓ) *be a general fpps/*k *with advice function* k. *Then there exists a functional proof system* f' *with* k *bits of input advice such that* f *and* f' *are p-equivalent.*

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

Definition 13 (Cook, Krajíček [9]). *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 *is can be decided by a polynomial-time (in* $|x| + |y|$*) algorithm which uses* $k(|x|)$ *bits of advice.*

It is easy to see that, as in the classical case without advice, relational proof systems with advice and functional proof systems with output advice are two formulations of the same concept:

Proposition 14. *Let* $k : \mathbb{N} \to \mathbb{N}$ *be a 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.*

As in the classical theorem of Cook and Reckhow [11], we get the following equivalence:

Theorem 15. *Let k be any function. Then there exists a polynomially bounded fpps/k with output advice if and only if* $\mathsf{coNP} \subseteq \mathsf{NP}/k$.

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. A strong positive result was shown by Cook and Krajíček [9].

Theorem 16 (Cook, Krajíček [9]). *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 17. *The class of all general fpps/$O(\log n)$ contains an optimal functional proof system with one bit of input advice.*

In the next definition we single out a large class of natural advice functions with at least logarithmic growth rate.

Definition 18. *A function k is* polynomially monotone *if k is computable in polynomial time and there exists a polynomial p, such that for each $x, y \in \Sigma^*$, $|y| \geq p(|x|)$ implies $|k(y)| > |k(x)|$.*

Polylogarithmic functions and polynomials are examples for polynomially monotone functions. If we consider proof systems with polynomially monotone advice functions, then we obtain p-optimal proof systems within each such class. This is the content of the next theorem which we prove by the same technique as was used for Theorem 16.

Theorem 19. *Let $k(n)$ be a polynomially monotone function. Then the class of all general fpps/k contains a p-optimal proof system.*

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 $k(|\ell(x)|) \geq k(|x|)+1$. Let $\|\cdot\|$ be an encoding of deterministic 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 define a functional proof system (P, ℓ) using advice function k, which is p-optimal for the class of all general *fpps*/k. Let Q be a system from the class of all general *fpps*/k. By Proposition 12 we may assume that Q has input advice.

First we will define a polynomial-time computable function f_Q translating Q-proofs into P-proofs and then we will describe how P works. We set $f_Q(\pi) = \pi 1^m$ where m is determined from the equation $m + |\pi| = \langle |\pi|, \|Q\|, |\pi|^{\|Q\|} \rangle$.

Now we define the system 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 system P receives at least one more bit of advice than Q. We can therefore use the first advice bit of P to certify that Q is indeed a correct propositional proof system when it is supplied with the last $k(|\pi|)$ advice bits of P. 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 $\top$. Apparently, P is correct and p-simulates every *fpps*/k Q with input advice via the polynomial-time computable function f_Q. Thus, by Proposition 12, P also p-simulates every general *fpps*/k. □

In a similar way we get:

Proposition 20. *For each constant $k \geq 0$ there exists an fpps with $k+1$ bits of input advice that p-simulates every fpps with k bits of input advice.*

Proof. (Sketch) The proof uses the same construction as in the proof of Theorem 19 with the following difference in the usage of advice: the last k advice bits of the new *fpps*/$(k+1)$ 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 we also have a p-optimal system within the class of all general *fpps*/k for constant k. Going back to the proof of Proposition 20, we observe that the proof system with $k+1$ advice bits, which simulates each with k bits, 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 for SAT_2 (the canonical complete problem for Σ_2^{p}), we can manage to reduce the amount of the necessary advice to exactly k bits, thus obtaining a p-optimal system within the class of all general *fpps*/k.

Theorem 21. *Assume that there exists a p-optimal proof system for SAT_2. Then for each constant $k \geq 1$ the class of all general fpps/k contains a p-optimal proof system.*

Proof. Similarly as in Sadowski's characterization of the existence of p-optimal propositional proof systems [20], we can prove:

> *There exists a p-optimal proof system for SAT_2 if and only if there exists a recursive enumeration M_i, $i \in \mathbb{N}$, of deterministic polynomial-time Turing machines such that*
>
> 1. *for every $i \in \mathbb{N}$ we have $L(M_i) \subseteq \mathrm{SAT}_2$ and*

2. *for every polynomial-time decidable subset* $L \subseteq \mathrm{SAT}_2$ *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 SAT_2 as above. For every proof system Q with k bits of input advice we construct a sequence of propositional formulas

$$Prf^Q_{m,n,k}(\pi, \varphi, a) \ ,$$

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 $Taut_n(\varphi)$ stating that the formula encoded by φ is a propositional tautology. As Q is an *fpps*$/k$, the formulas

$$Correct^Q_{m,n,k} = (\exists a)(\forall \pi, \varphi) \left(Prf^Q_{m,n,k}(\pi, \varphi, a) \rightarrow Taut_n(\varphi) \right)$$

are quantified Boolean formulas from SAT_2 for every $n, m \geq 0$. Because these formulas can be constructed in polynomial time from Q, there exists an index $i \in \mathbb{N}$ such that M_i accepts the set $\{Correct^Q_{m,n,k} \mid m, n \geq 0\}$.

Now we construct a p-optimal system P with k bits of input advice as follows: at input x we compute the unique numbers $m_1, \ldots, m_4$ such that $|x| = \langle m_1, \ldots, m_4 \rangle$. As in the proof of Theorem 19, we set $\pi = x_1 \ldots x_{m_1}$ and $\|Q\| = m_2$. The system 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 $Correct^Q_{m_1,|\varphi|,k}$. 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. Therefore P is a correct *fpps*$/k$. To show the p-optimality of P, let Q be an *fpps*$/k$ with input advice and let M_i be the machine accepting $\{Correct^Q_{m,n,k} \mid m, n \geq 0\}$. Then the system Q is p-simulated by P via the mapping $\pi \mapsto \pi 1^m$ where $m = \langle |\pi|, \|Q\|, |\pi|^{\|Q\|}, i \rangle - |\pi|$. □

All the optimal and p-optimal proof systems 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. Our next result shows that this is rather unlikely, as otherwise collapse assumptions of presumably different strength would be equivalent. This result indicates that input advice for propositional proof systems is indeed a more powerful concept than output advice.

Theorem 22. *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 3), 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 3 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 15 this guarantees the existence of a polynomially bounded system P with $O(\log n)$ bits of output advice. By Theorem 16, 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 15 we obtain $\mathsf{coNP} \subseteq \mathsf{NP}/k$. □

With respect to the optimal proof system from Corollary 17 we obtain:

Corollary 23. *The optimal fpps*/1 *from Corollary 17 is not equivalent to an fpps*/1 *with output advice, unless* $\mathsf{PH} \subseteq \mathsf{BH}$ *implies* $\mathsf{PH} \subseteq \mathsf{D}^{\mathsf{p}}$.

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 [9] have defined the notion of extended Frege systems using advice. We give a more general definition.

Definition 24. *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 an EF-proof of an implication $\psi \to \varphi$, where ψ is a simple substitution instance of a formula from Φ (where 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 10).

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 [9], which are motivated

by the proof of Theorem 8. Cook and Krajíček proved that these systems are polynomially bounded if VPV proves $\mathsf{coNP} \subseteq \mathsf{NP}/O(1)$.

Our next result shows that the optimal proof systems constructed in Sect. 6 are equivalent to natural extensions of extended Frege systems with advice.

Theorem 25. *1. Let $k(n)$ be a polynomially monotone function. Then there exists a set $\Phi \in \mathsf{P}/k(n)$ such that $EF + \Phi/k$ is p-optimal for the class of all general $fpps/k(n)$.*
2. For every constant $k \geq 1$ there exists a set $\Phi \in \mathsf{P}/k$ such that $EF + \Phi/k$ p-simulates every general $fpps/k - 1$.
3. In contrast, none of the extensions of EF as defined in [9] simulates every general $fpps/1$, unless items 1 to 4 from Theorem 22 are equivalent.

Comparing the definition of EF with advice from [9] with our Definition 24, 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 24 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 [17]. In contrast, Cook and Krajíček's extended Frege systems with advice lead exactly to EF, if no advice is used. On the other hand, these systems appear to be strictly weaker than the systems from Definition 24, as indicated by item 3 of Theorem 25.

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. [9]). 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 [3] conjecture $\mathsf{coNP} \subseteq \mathsf{NP}/O(\log n) \iff \mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ (cf. also [12]). This seems unlikely, as Cook and Krajíček [9] 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 [3] to prove the converse implication. Is even $\mathsf{coNP} \subseteq \mathsf{NP}/\mathit{poly} \iff \mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$ true, possibly with the stronger hypothesis that both inclusions are provable in S^2_2? Currently, $\mathsf{coNP} \subseteq \mathsf{NP}/\mathit{poly}$ is only known to imply $\mathsf{PH} \subseteq \mathsf{S}^{\mathsf{NP}}_2$ [5].

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.

Acknowledgements

We are grateful to Jan Krajíček and the anonymous referees for helpful comments and detailed suggestions on how to improve this paper.

References

1. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin Heidelberg, 1988.
2. R. Beigel. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science*, 84:199–223, 1991.
3. H. Buhrman, R. Chang, and L. Fortnow. One bit of advice. In *Proc. 20th Symposium on Theoretical Aspects of Computer Science*, pages 547–558, 2003.
4. J.-Y. Cai. $S_2^p \subseteq ZPP^{NP}$. *Journal of Computer and System Sciences*, 73(1):25–35, 2007.
5. J.-Y. Cai, V. T. Chakaravarthy, L. A. Hemaspaandra, and M. Ogihara. Competing provers yield improved Karp-Lipton collapse results. *Information and Computation*, 198(1):1–23, 2005.
6. R. Chang and J. Kadin. The Boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340–354, 1996.
7. S. A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proc. 7th Annual ACM Symposium on Theory of Computing*, pages 83–97, 1975.
8. S. A. Cook. Theories for complexity classes and their propositional translations. In J. Krajíček, editor, *Complexity of Computations and Proofs*, pages 175–227. Quaderni di Matematica, 2005.
9. 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.
10. S. A. Cook and P. Nguyen. Foundations of proof complexity: Bounded arithmetic and propositional translations. Book in progress, Available from http://www.cs.toronto.edu/~sacook.
11. S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44:36–50, 1979.
12. L. Fortnow and A. R. Klivans. NP with small advice. In *Proc. 20th Annual IEEE Conference on Computational Complexity*, pages 228–234, 2005.
13. E. Jeřábek. Approximate counting by hashing in bounded arithmetic. Preprint, 2007.
14. J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988.
15. R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symposium on Theory of Computing*, pages 302–309. ACM Press, 1980.
16. J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. *SIAM Journal on Computing*, 28(1):311–324, 1998.
17. J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.
18. J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54:1063–1079, 1989.
19. J. Krajíček, P. Pudlák, and G. Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.
20. Z. Sadowski. On an optimal propositional proof system and the structure of easy subsets of TAUT. *Theoretical Computer Science*, 288(1):181–193, 2002.
21. D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.