


Upward Translation of Optimal and P-Optimal Proof Systems in the Boolean Hierarchy over NP

Fabian Egidy  

Julius-Maximilians-Universität Würzburg, Germany

Christian Glaßer 

Julius-Maximilians-Universität Würzburg, Germany

Martin Herold 

Max-Planck-Institut für Informatik, Saarbrücken, Germany

Abstract

We study the existence of optimal and p-optimal proof systems for classes in the Boolean hierarchy over NP. Our main results concern DP, i.e., the second level of this hierarchy:

- If all sets in DP have p-optimal proof systems, then all sets in coDP have p-optimal proof systems.
- The analogous implication for optimal proof systems fails relative to an oracle.

As a consequence, we clarify such implications for all classes \mathcal{C} and \mathcal{D} in the Boolean hierarchy over NP: either we can prove the implication or show that it fails relative to an oracle.

Furthermore, we show that the sets SAT and TAUT have p-optimal proof systems, if and only if all sets in the Boolean hierarchy over NP have p-optimal proof systems which is a new characterization of a conjecture studied by Pudlák.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Oracles and decision trees

Keywords and phrases Computational Complexity, Boolean Hierarchy, Proof Complexity, Proof Systems, Oracle Construction

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.44

Related Version *Full Version:* <https://arxiv.org/abs/2304.14702>

Funding *Fabian Egidy:* supported by the German Academic Scholarship Foundation.

Martin Herold: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 399223600.

1 Introduction

This paper contributes to the study of proof systems initiated by Cook and Reckhow [11]. A proof system for a set L is a polynomial-time computable function f whose range is L . Cook and Reckhow motivate the study of proof systems with the $\text{NP} = \text{coNP}$ question: they consider propositional proof systems (pps), i.e., proof systems for the set of propositional tautologies (TAUT). They show that there exists a pps with polynomially bounded proofs if and only if $\text{NP} = \text{coNP}$. This approach to the $\text{NP} = \text{coNP}$ question is called the Cook-Reckhow program [9]. To obtain $\text{NP} \neq \text{coNP}$ one can either show that optimal pps (i.e., pps with at most polynomially longer proofs than any other pps) do not exist or show that a specific pps is optimal and has a non-polynomial lower bound on the length of proofs. This connection led to the investigation of upper and lower bounds for different pps [20] as well as the existence of optimal and p-optimal¹ proof systems for general sets.

¹ A stronger notion of optimal. We write (p-)optimal when the statement holds using optimal as well as p-optimal.



© Fabian Egidy, Christian Glaßer, and Martin Herold;
licensed under Creative Commons License CC-BY 4.0

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).

Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 44; pp. 44:1–44:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The latter question was explicitly posed by Krajíček and Pudlák [21] in the context of finite consistency. They revealed the following connection between both concepts: optimal pps exist if and only if there is a finitely axiomatized theory S that proves for every finitely axiomatized theory T the statement “ T has no proof of contradiction of length n ” by a proof of polynomial length in n . If optimal pps exist, then a weak version of Hilbert’s program is possible, i.e., proving the “consistency up to some feasible length of proofs” of all mathematical theories [24]. We refer to Krajíček [19] and Pudlák [26] for details on the relationship between proof systems and bounded arithmetic. More recently, Pudlák [25] draws new connections of (p-)optimal proof systems and statements about incompleteness in the finite domain.

Furthermore, proof systems have shown to be tightly connected to promise classes, especially pps to the class of disjoint NP-pairs, called DisjNP. Initiated by Razborov [27], who showed that the existence of p-optimal pps implies the existence of complete sets in DisjNP, many further connections were investigated. More generally, Köbler, Messner and Torán [18] show that the existence of p-optimal proof systems for sets of the polynomial-time hierarchy imply complete sets for promise classes like UP, $\text{NP} \cap \text{coNP}$, and BPP. Beyersdorff, Köbler, and Messner [7] and Pudlák [25] connect proof systems to function classes by showing that p-optimal proof systems for SAT imply complete sets for TFNP. Questions regarding non-deterministic function classes can be characterized by questions about proof systems [7]. Beyersdorff [3, 4, 5, 6], Beyersdorff and Sadowski [8] and Glaßer, Selman, and Zhang [13, 14] show further connections between pps and disjoint NP-pairs.

The above connections to important questions of complexity theory, bounded arithmetic, and promise classes motivate the investigation of the question “which sets do have optimal proof systems” posed by Messner [22]. Krajíček and Pudlák [21] were the first to study sufficient conditions for pps by proving that $\text{NE} = \text{coNE}$ implies the existence of optimal pps and $\text{E} = \text{NE}$ implies the existence of p-optimal pps. Köbler, Messner, and Torán [18] improve this result to $\text{NEE} = \text{coNEE}$ for optimal pps and $\text{EE} = \text{NEE}$ for p-optimal pps. Sadowski [28] shows different characterizations for the existence of optimal pps, e.g., the uniform enumerability of the class of all easy subsets of TAUT. In certain settings one can prove the existence of optimal proof systems for different classes: e.g., by allowing one bit of advice [10], considering randomized proof systems [16, 15], or using a weak notion of simulation [29].

Messner [22] shows that all nonempty² sets in P but not all sets in E have p-optimal proof systems. Similarly, all sets in NP but not all sets in coNE have optimal proof systems. Therefore, when going from smaller to larger complexity classes, there has to be a tipping point such that all sets contained in classes below this point have (p-)optimal proof systems, but some set contained in all classes above this point has no (p-)optimal proof systems. Unfortunately, oracle constructions tell us that for many classes between P and E (resp., NP and coNE) the following holds: with relativizable proofs one can neither prove nor refute that p-optimal (resp., optimal) proof systems exist (e.g. coNP [1, 21] and PSPACE [1, 12]). Thus, with the currently available means it is not possible to precisely locate this tipping point, but we can rule out certain regions for its location. For this, we investigate how the existence of (p-)optimal proof system for all sets of the class \mathcal{C} “translate upwards” to all sets of a class \mathcal{D} with $\mathcal{C} \subseteq \mathcal{D}$. This rules out tipping points between \mathcal{C} and \mathcal{D} .

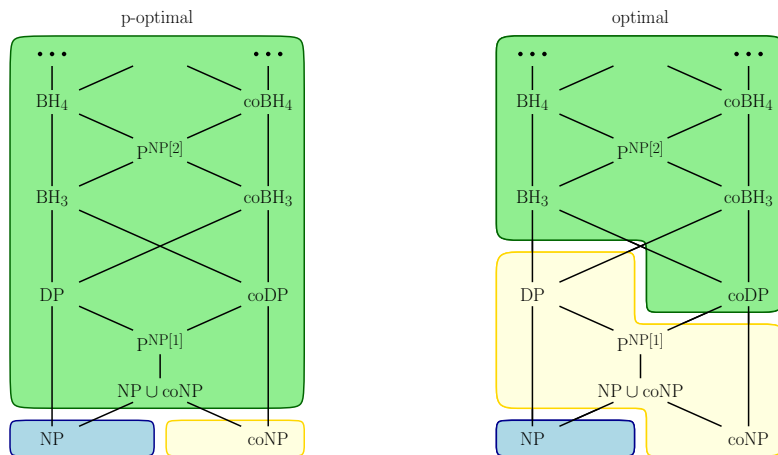
² By our definition, FP-functions are total, thus the empty set has no proof system. For the rest of this paper, we omit the word “nonempty” when referring to proof systems for all sets of a class, since this is only a technicality.

Our Contribution. Motivated by Messner’s general question, we study the existence of (p-)optimal proof systems for classes inside the Boolean hierarchy over NP. We use the expression “the class \mathcal{C} has (p-)optimal proof systems” for “all sets of a class \mathcal{C} have (p-)optimal proof systems”. We say that two classes \mathcal{C} and \mathcal{D} are equivalent with respect to (p-)optimal proof systems if \mathcal{C} has (p-)optimal proof systems if and only if \mathcal{D} has (p-)optimal proof systems.

For the classes of the Boolean hierarchy over NP, denoted by BH, we identify three equivalence classes for p-optimal proof systems and three other classes for optimal proof systems. We also show that the classes of the bounded query hierarchy over NP are all equivalent for p-optimal proof systems and we identify two equivalence classes for optimal proof systems. Moreover, we show that relativizable techniques cannot prove all identified equivalence classes to coincide. These results follow from our main theorems:

- (i) If DP has p-optimal proof systems, then coDP has p-optimal proof systems.
- (ii) There exists an oracle relative to which coNP has p-optimal proof systems and coDP does not have optimal proof systems.

Using the result by Köbler, Messner, and Torán that (p-)optimality is closed under intersection [18] and two oracles by Khaniki [17], we obtain the equivalence classes visualized in Figure 1, which cannot be proved to coincide with relativizable proofs.



■ **Figure 1** Equivalence classes for p-optimal proof systems (left) and optimal proof systems (right) in the Boolean hierarchy over NP and the bounded query hierarchy over NP.

This clarifies all questions regarding relativizably provable translations of (p-)optimal proof systems for classes in the Boolean hierarchy over NP and the bounded query hierarchy over NP. We cannot expect to prove any further translations with the currently available means, because for every such translation there is an oracle against it. So we are dealing with an interesting situation: while p-optimal proof systems for DP *relativizably imply* p-optimal proof systems for coDP, this does not hold for optimal proof systems. Similarly, all classes of the bounded query hierarchy over NP are equivalent with respect to p-optimal proof systems, but $P^{NP[1]}$ and $P^{NP[2]}$ cannot be shown to be equivalent with respect to optimal proof systems by a relativizable proof. The result drastically limits the potential locations of a tipping point in the BH and the bounded query hierarchy over NP. They can only occur between two classes belonging to two different equivalence classes.

Furthermore, our results provide a new perspective on an hypothesis related to feasible versions of Gödel’s incompleteness theorem: Pudlák [25] studies several conjectures about incompleteness in the finite domain by investigating the (un)provability of sentences of some

specific form in weak theories. These conjectures can also be expressed as the non-existence of complete sets in promise classes or non-existence of (p-)optimal proof systems for sets. Pudlák considers the conjecture $\text{CON} \vee \text{SAT}$ stating that TAUT does not have p-optimal proof systems or SAT does not have p-optimal proof systems. Khaniki [17] proves this conjecture to be equivalent to RFN_1 , which is another conjecture considered by Pudlák. Our results show that both conjectures are equivalent to the statement that BH does not have p-optimal proof systems.

2 Preliminaries

Let $\Sigma = \{0, 1\}$ be the default alphabet and Σ^* be the set of finite words over Σ . We call subsets of Σ^* languages and sets of languages classes. We denote the length of a word $w \in \Sigma^*$ by $|w|$. The i -th letter of a word w for $0 \leq i < |w|$ is denoted as $w(i)$, i.e., $w = w(0)w(1) \cdots w(|w| - 1)$.

The set of all (positive) natural numbers is denoted as \mathbb{N} (\mathbb{N}^+). We write the empty set as \emptyset . We identify Σ^* with \mathbb{N} through the polynomial time computable and invertible bijection $\Sigma^* \rightarrow \mathbb{N}; w \mapsto \sum_{i < |w|} (1 + w(i))2^i$. This is a variant of the dyadic representation. Thus, we can treat words from Σ^* as numbers from \mathbb{N} and vice versa, which allows us to use notations, relations and operations of words for numbers and vice versa (e.g. we can define the length of a number by this). We resolve the ambiguity of 0^i and 1^i by always interpreting them as words from Σ^* . The cardinality of a set A is denoted as $|A|_c$. For $\circ \in \{<, \leq, =, \geq, >\}$, a set $A \subseteq \Sigma^*$ and a number $n \in \mathbb{N}$ we define $A^{\circ n} = \{w \in A \mid |w| \circ n\}$. For a clearer notation we use $\Sigma^{\circ n}$ as $\Sigma^{*\circ n}$ and Σ^n for $\Sigma^{=n}$. The operators \cup , \cap , and \setminus denote the union, intersection and set-difference. We denote the complement of a set A relative to Σ^* as $\bar{A} = \Sigma^* \setminus A$.

The image of a function f is denoted as $\text{img}(f)$. Let $\langle \cdot \rangle: \bigcup_{i \geq 0} \mathbb{N}^i \rightarrow \mathbb{N}$ be an injective polynomial time computable and invertible pairing function such that $|\langle u_1, \dots, u_n \rangle| = 2(|u_1| + \dots + |u_n| + n)$. The logarithm to the base 2 is denoted as \log . Furthermore, we define polynomial functions $p_i: \mathbb{N} \rightarrow \mathbb{N}$ for $i \in \mathbb{N}^+$ by $p_i(x) = x^i + i$.

We use the default model of a Turing machine in the deterministic as well as in the non-deterministic variation, abbreviated by DTM and NTM respectively. The language decided by a Turing machine M is denoted as $L(M)$. For a number $s \in \mathbb{N}$ the language of words that are accepted by a Turing machine M in s computation steps is denoted as $L^s(M)$. We use Turing transducer to compute functions. For a Turing transducer F we write $F(x) = y$ when on input x the transducer outputs y . A Turing transducer F computes a total function and we sometimes refer to the function computed by F as “the function F ”. Let $\{F_i\}_{i \in \mathbb{N}}$ and $\{G_i\}_{i \in \mathbb{N}}$ be standard enumerations of polynomial time Turing transducers. Let $\{N_i\}_{i \in \mathbb{N}}$ be a standard enumeration of non-deterministic polynomial time Turing machines with the special property that N_0 is the machine that always rejects and N_1 is the machine that always accepts, that is $L(N_0) = \emptyset$ and $L(N_1) = \mathbb{N}$. The runtime of F_i , G_i and N_i is bounded by p_i .

► **Proposition 1.** *There is a Turing machine M and a Turing transducer F such that for all $i, s, x \in \mathbb{N}$ the following properties hold:*

- $\langle i, x, 0^s \rangle \in L(M) \Leftrightarrow x \in L^s(N_i)$
- $F(\langle i, x, 0^s \rangle) = \begin{cases} \langle 1, F_i(x) \rangle & \text{if } F_i(x) \text{ stops within } s \text{ steps} \\ \langle 0, 0 \rangle & \text{else} \end{cases}$
- Both machines run in time $O(|i|s \log s)$.

FP, P, and NP denote standard complexity classes [23]. For a class \mathcal{C} define $\text{co}\mathcal{C} = \{A \subseteq \Sigma^* \mid \overline{A} \in \mathcal{C}\}$. We define the Boolean hierarchy over NP inductively. Let \mathcal{C} and \mathcal{D} be arbitrary complexity classes. First, we define boolean operators on classes:

$$\begin{aligned}\mathcal{C} \wedge \mathcal{D} &= \{A \cap B \mid A \in \mathcal{C} \wedge B \in \mathcal{D}\} \\ \mathcal{C} \vee \mathcal{D} &= \{A \cup B \mid A \in \mathcal{C} \vee B \in \mathcal{D}\}\end{aligned}$$

Then $\text{BH}_1 = \text{NP}$, $\text{BH}_{2k} = \text{coNP} \wedge \text{BH}_{2k-1}$, $\text{BH}_{2k+1} = \text{NP} \vee \text{BH}_{2k}$, and $\text{BH} = \bigcup_{k \geq 1} \text{BH}_k$ where BH_2 is called DP and BH is called Boolean hierarchy over NP. We want to emphasize that $\text{DP} = \text{NP} \wedge \text{coNP}$ and $\text{coDP} = \text{NP} \vee \text{coNP}$. Wagner [30] showed that $\text{BH}_k \subseteq \text{BH}_{k+1}$ and $\text{BH}_k \subseteq \text{coBH}_{k+1}$. The classes $\text{P}^{\text{NP}[k]}$ for $k \in \mathbb{N}^+$ contain all sets that can be accepted by a polynomial time Turing machine that queries at most k elements from an NP-set. The resulting hierarchy $\text{P}^{\text{NP}[1]}, \text{P}^{\text{NP}[2]}, \dots$ is called bounded query hierarchy over NP. Beigel [2] shows that $\text{BH}_{2^k-1} \cup \text{coBH}_{2^k-1} \subseteq \text{P}^{\text{NP}[k]} \subseteq \text{BH}_{2^k} \cap \text{coBH}_{2^k}$.

We use the common polynomial time many-one reducibility for sets $A, B \subseteq \Sigma^*$, i.e., $A \leq_m^p B$ if there exists an $f \in \text{FP}$ such that $x \in A \Leftrightarrow f(x) \in B$. For a class \mathcal{C} and some problem A , we say that A is hard for \mathcal{C} if for all $B \in \mathcal{C}$ it holds $B \leq_m^p A$. The set A is called complete for \mathcal{C} if $A \in \mathcal{C}$ and A is hard for \mathcal{C} . We define the following complete problems for NP and DP.

$$\begin{aligned}\mathcal{C} &= \{\langle 0^i, x, 0^p \rangle \mid i \in \mathbb{N}, x \in \Sigma^* \text{ and } x \in L^p(N_i)\} \\ \mathcal{D} &= \{\langle 0^i, 0^j, x, 0^p \rangle \mid i, j \in \mathbb{N}, x \in \Sigma^* \text{ and } x \in L^p(N_i) \cap \overline{L^p(N_j)}\} \\ \mathcal{D}' &= \mathcal{D} \cup \{w \mid \nexists i, j \in \mathbb{N}, x \in \Sigma^* : \langle 0^i, 0^j, x, 0^p \rangle = w\}\end{aligned}$$

It is easy to see that \mathcal{C} is NP-complete and \mathcal{D} and \mathcal{D}' are DP-complete. Furthermore, their complements are complete for coNP and coDP respectively. The purpose of \mathcal{D}' is that $\overline{\mathcal{D}'}$ consists only of words of the form $\langle 0^i, 0^j, x, 0^p \rangle$, which simplifies some arguments in section 3. Let $N_{\mathcal{C}}$ denote the polynomial time machine with $L(N_{\mathcal{C}}) = \mathcal{C}$.

We use proof systems for sets defined by Cook and Reckhow [11]. They define a function $f \in \text{FP}$ to be a proof system for $\text{img}(f)$. Furthermore:

- A proof system g is (p-)simulated by a proof system f , denoted by $g \leq f$ (resp., $g \leq^p f$), if there exists a total function π (resp., $\pi \in \text{FP}$) and a polynomial p such that $|\pi(x)| \leq p(|x|)$ and $f(\pi(x)) = g(x)$ for all $x \in \Sigma^*$. In this context the function π is called simulation function. Note that $g \leq^p f$ implies $g \leq f$.
- A proof system f is (p-)optimal for $\text{img}(f)$, if $g \leq f$ (resp., $g \leq^p f$) for all $g \in \text{FP}$ with $\text{img}(g) = \text{img}(f)$.
- A complexity class \mathcal{C} has (p-)optimal proof systems, if every $A \in \mathcal{C}$ with $A \neq \emptyset$ has a (p-)optimal proof system.
- We say that (p-)optimal proof systems translate from a class \mathcal{C} to \mathcal{D} if the existence of (p-)optimal proof systems for \mathcal{C} implies their existence for \mathcal{D} .

By the following result of Köbler, Messner and Torán [18], we can prove or refute the existence of (p-)optimal proof systems for a class \mathcal{C} by proving or refuting the existence of such proof systems for a complete set of \mathcal{C} .

► **Proposition 2** ([18]). *If $A \subseteq \Sigma^*$ has a (p-)optimal proof system and $\emptyset \neq B \leq_m^p A$, then B has a (p-)optimal proof system.*

► **Corollary 3.** *If $A \subseteq \Sigma^*$ is a hard set for some class \mathcal{C} and A has a (p-)optimal proof system, then \mathcal{C} has (p-)optimal proof systems.*

Furthermore, it was shown by Köbler, Messner, and Torán [18] that the class of sets having (p-)optimal proof systems is closed under intersection. This result can easily be extended to the operator \wedge for complexity classes.

► **Proposition 4** ([18]). *If $A, B \subseteq \Sigma^*$, $A \cap B \neq \emptyset$ and both sets have a (p-)optimal proof system, then $A \cap B$ has a (p-)optimal proof system.*

► **Corollary 5.** *If two classes \mathcal{C} and \mathcal{D} have (p-)optimal proof systems, then $\mathcal{C} \wedge \mathcal{D}$ has (p-)optimal proof systems.*

Finally, every (p-)optimal proof system can be transformed into a (p-)optimal proof system that runs in linear time by polynomially padding the proofs.

► **Proposition 6.** *If f is a (p-)optimal proof system for $A \subseteq \Sigma$, then there is a (p-)optimal proof system g for A that runs in linear time.*

3 Translation of P-Optimal Proof Systems from DP to coDP

In this chapter we show that p-optimal proof systems for DP imply p-optimal proof systems for coDP. This proof is based on machine simulation which is a relativizable proof technique. Thus, the following theorem also holds in the presence of an arbitrary oracle O .

► **Theorem 7.** *If there exists a p-optimal proof system for D , then there exists a p-optimal proof system for \overline{D} .*

We start by sketching the key idea used in the proof. Our approach needs some technique to verify that a given instance is in \overline{D} . There is no known way to decide \overline{D} in polynomial time, but we can use the p-optimal proof system for D for this verification. We define a function $f' : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that there is a polynomial-time-computable encoding $c : \mathbb{N} \rightarrow \mathbb{N}$ with $f'(a, c(x)) \in D$ if and only if $F_a(x) \in \overline{D}$ for all $a \in \mathbb{N}$ and $x \in \mathbb{N}$. Furthermore, $f'(a, x)$ can be computed in time $|x|^{O(a)}$. We derive a class of functions $\{f'_a\}_{a \in \mathbb{N}}$ from f' by fixing the first input to a . Note that f'_a runs in polynomial time for a fixed $a \in \mathbb{N}$ and that f'_a is a proof system for D if and only if F_a is a proof system for \overline{D} . Now, we define a machine that uses an additional input to verify $F_a(x) \in \overline{D}$. The inputs of the machine are a, x, b and it accepts if and only if $f(F_b(x)) = f'_a(x)$ for a p-optimal proof system f of D . So, if F_a is a proof system for \overline{D} , we know f'_a is a proof system for D . Thus, by the fact that f is p-optimal, there is a $b \in \mathbb{N}$ such that $f(F_b(x)) = f'_a(x)$ for all $x \in \mathbb{N}$. Thus, when knowing the value b , the machine can verify $F_a(x) \in \overline{D}$ for all $x \in \mathbb{N}$ for a proof system $F_a(x)$ for \overline{D} by accepting $f(F_b(c(x))) = f'_a(c(x))$. On the other hand if $F_a(x) \notin \overline{D}$ there is no b such that $f(F_b(c(x))) = f'_a(c(x))$ because $f'_a(c(x)) \notin D = \text{img}(f)$.

Proof. We start by defining an NTM A that checks for given a, y' whether $F_a(y') = \langle i, j, x', 0^p \rangle \in \overline{D}$. Since a deterministic polynomial-time computation cannot check every possible path of a coNP machine N_j , A gets a path y of N_j as an additional input and has the property that it accepts for all possible paths y if and only if $F_a(y') \in \overline{D}$. Arguing over all y' this means if F_a is a proof system for \overline{D} , then for all y' and all corresponding paths y the machine A accepts on input a, y', y .

Let f be a p-optimal proof system for D . Without loss of generality we assume $f(x)$ can be computed in $O(|x|)$ time by Proposition 6. We define A on input x as follows.

- (i) Check whether $x = \langle a, y, y' \rangle$ for some $a \in \mathbb{N}$ and $y, y' \in \Sigma^*$, otherwise reject.
- (ii) Check whether $F_a(y') = \langle i, j, x', 0^p \rangle$ with $i, j, p \in \mathbb{N}$ and $x' \in \Sigma^*$ and whether $y \in \Sigma^p$, otherwise reject.

- (iii) Accept if $N_i(x')$ does not accept on path y within p steps.
- (iv) Simulate the first p steps of $N_j(x')$.
- (v) Accept if the simulation of $N_j(x')$ accepted within the first p steps, otherwise reject.

► **Observation 8.** $A(x)$ runs in time $O(|x|^{3a})$ for $x = \langle a, y, y' \rangle$.

Proof. Checking whether x has the required format is possible in linear time. By Proposition 1, $F_a(y')$ can be computed in time

$$\begin{aligned} O(|a| \cdot (|y'|^a + a) \log(|y'|^a + a)) &\subseteq O(|a| \cdot (|y'|^a + a)^2) \subseteq O((|x|^a + a)^3) \\ &\subseteq O(|x|^{3a} + a^3) \subseteq O(|x|^{3a}). \end{aligned}$$

The first p steps of the path y of $N_i(x')$ can be simulated in time

$$O(|i| \cdot p \log p) \subseteq O(|F_a(y')| \cdot |F_a(y')|^2) \subseteq O(|x|^{3a})$$

for all $a \in \mathbb{N}$. The computation $N_j(x')$ can be simulated in $O(|j| \cdot p \log p) \subseteq O(|x|^{3a})$ time for all $a \in \mathbb{N}$. ◀

► **Claim 9.** Let $F_a(y') = \langle i, j, x', 0^p \rangle$. Then $F_a(y') \in \overline{D'} \Leftrightarrow \forall y \in \Sigma^p : \langle a, y, y' \rangle \in L(A)$.

Proof. First we show " \Rightarrow ". We consider two cases:

- Suppose $x' \in L^p(N_i)$. By $\langle i, j, x', 0^p \rangle \in \overline{D'}$, it holds $x' \in L^p(N_j)$. The machine A on input $\langle a, y, y' \rangle$ with $F_a(y') = \langle i, j, x', 0^p \rangle \in \overline{D'}$ and $y \in \Sigma^p$ rejects only, if the non-deterministic check in step (iv) fails. But this is impossible since $x' \in L^p(N_j)$.
- Suppose $x' \notin L^p(N_i)$. In this case $N_i(x')$ does not accept within p steps for all paths $y \in \Sigma^p$. Thus, the machine accepts in step (iii).

Now, we show " \Leftarrow ". Again, we distinguish two cases:

- Suppose $A(\langle a, y, y' \rangle)$ accepts in step (iii) for all $y \in \Sigma^p$. This implies that $x' \notin L^p(N_i)$. Thus, $\langle i, j, x', 0^p \rangle \in \overline{D'}$.
- If A accepts but not in step (iii), we conclude it accepts in step (v). Hence, it holds $x' \in L^p(N_j)$ and $\langle i, j, x', 0^p \rangle \in \overline{D'}$. ◀

We want to define functions f_a in such a way that f_a is a proof system for D if F_a is a proof system for $\overline{D'}$. For this, we can exploit the relationship of A to proof systems of $\overline{D'}$ shown in Claim 9. Specifically, f_a trusts that $A(\langle a, y, y' \rangle)$ accepts for specific y and y' (note that the accepting behavior of A has influence on D), which is equivalent to F_a being a proof system for $\overline{D'}$.

Choose $a_1 \in \mathbb{N}$ such that $N_{a_1} = A$. Let $k_A \in \mathbb{N}^+$ be a constant such that $A(x)$ runs in time $k_A|x|^{3a} + k_A$. Recall that N_0 always rejects. We define a function $f^* : \mathbb{N} \rightarrow \mathbb{N}$:

$$f^*(x) = \begin{cases} \langle a_1, 0, \langle a, y, y' \rangle, 0^{k_A|x|^{3a}+k_A} \rangle & \text{if } x = \langle a, y, y' \rangle 0 \wedge F_a(y') = \langle i, j, x', 0^p \rangle \\ & \wedge y \in \Sigma^p \\ f(x') & \text{if } x = x' 1 \\ f(0) & \text{else} \end{cases}$$

► **Observation 10.** $f^*(x)$ runs in $O(|x|^{3a})$ time for $x = \langle a, y, y' \rangle 0$.

Proof. The case distinction for the first case is possible in $O(|x|^{3a})$ time because computing $F_a(y')$ is possible in that time. The output of the first case with exception of the unary runtime parameter $0^{k_A \cdot (|x|^{3a}) + k_A}$ is possible in linear time. The unary runtime parameter can be computed in $O(|x|^{3a})$ time. The output of the other cases is possible in linear time. ◀

We define a function f_a that is obtained from f^* by fixing an index $a \in \mathbb{N}$ of a polynomial-time function F_a .

$$f_a(x) = \begin{cases} f^*(\langle a, y, y' \rangle 0) & \text{if } x = \langle a, y, y' \rangle 0 \\ f(x') & \text{if } x = x' 1 \\ f(0) & \text{else} \end{cases}$$

► **Observation 11.** For $a \in \mathbb{N}$ and $y, y' \in \Sigma^*$ it holds that $f_a(\langle a, y, y' \rangle 0) = f^*(\langle a, y, y' \rangle 0)$.

► **Observation 12.** For a fixed $a \in \mathbb{N}$ the function $f_a(x)$ can be computed in polynomial time.

Proof. This follows from Observation 10 and the linear runtime of f . ◀

▷ **Claim 13.** Let $a \in \mathbb{N}$ and $y' \in \Sigma^*$ such that $F_a(y') = \langle i, j, x', 0^p \rangle \notin \overline{D'}$. Then there is a $y \in \Sigma^p$, such that $f_a(\langle a, y, y' \rangle 0) \notin D$.

Proof. By Claim 9 we conclude the existence of a $y \in \Sigma^p$ such that $A(\langle a, y, y' \rangle)$ rejects. This implies $f_a(\langle a, y, y' \rangle 0) = \langle a_1, 0, \langle a, y, y' \rangle, 0^{k_A|x|^{3a}+k_A} \rangle \notin D$. ◀

▷ **Claim 14.** Let $a \in \mathbb{N}$. Then $\text{img}(F_a) \subseteq \overline{D'} \Leftrightarrow \text{img}(f_a) = D$.

Proof. First we show " \Rightarrow ". $D \subseteq \text{img}(f_a)$ holds because $\text{img}(f) \subseteq \text{img}(f_a)$ and f is a proof system for D . Let $x \in \Sigma^*$. In the bottom two cases of f_a and f^* it is easy to see $f_a(x) \in D$ and $f^*(x) \in D$. So we can assume $x = \langle a, y, y' \rangle 0$ with $F_a(y') = \langle i, j, x', 0^p \rangle$ and $y \in \Sigma^p$. Since $\text{img}(F_a) \subseteq \overline{D'}$, it holds $\langle i, j, x', 0^p \rangle \in \overline{D'}$. By Claim 9 we obtain that $\langle a, y, y' \rangle \in L(A)$ for all $y \in \Sigma^p$. Thus, $f_a(x) = f^*(\langle a, y, y' \rangle 0) = \langle a_1, 0, \langle a, y, y' \rangle, 0^{k_A|x|^{3a}+k_A} \rangle \in D$ since for all $y \in \Sigma^p$, $N_{a_1}(\langle a, y, y' \rangle)$ accepts, $N_0(\langle a, y, y' \rangle)$ rejects and $|\langle a, y, y' \rangle| \leq |x|$.

" \Leftarrow " follows directly as the contraposition of Claim 13. ◀

We want to define another NTM B that checks for given a, y' whether $F_a(y') \in \overline{D'}$. To achieve this we use Claims 13 and 14. B checks for all $y \in \Sigma^p$ whether $f(F_b(\langle a, y, y' \rangle 0)) = f_a(\langle a, y, y' \rangle 0)$ on input a, b, y' . So if F_a is a proof system for $\overline{D'}$, then there is a b such that $B(a, b, y')$ rejects for all y' . Furthermore, if $F_a(y') \notin \overline{D'}$, then $B(a, b, y')$ accepts for all b . $B(x)$ operates as follows.

- (i) Check whether $x = \langle a, b, y' \rangle$ for some $a, b \in \mathbb{N}$ and $y' \in \Sigma^*$, otherwise reject.
- (ii) Check whether $F_a(y') = \langle i, j, x', 0^p \rangle$ for some $i, j, p \in \mathbb{N}$ and $x' \in \Sigma^*$, otherwise reject.
- (iii) Branch non-deterministically every $y \in \Sigma^p$.
- (iv) Accept if $f(F_b(\langle a, y, y' \rangle 0)) \neq f_a(\langle a, y, y' \rangle 0)$.
- (v) Reject.

► **Observation 15.** $B(x)$ runs in time $O(|x|^{9a^2b})$ for $x = \langle a, b, y' \rangle$.

Proof. Checking whether the input is formatted correctly is possible in linear time. $F_a(y')$ can be computed in $O(|a| \cdot (|y'|^a + a) \log(|y'|^a + a)) \subseteq O(|x|^{3a})$ time. We also observe that $|F_a(y')| \leq |y'|^a + a$. In line (iv) it holds that $|F_a(y')| \geq 2p \geq 2|y|$ and in line 1 it holds that $2|a| + 2|y'| + 8 \leq |x|$, and therefore,

$$|\langle a, y, y' \rangle 0| = 2(|a| + |y| + |y'| + 3) + 1 \leq |F_a(y')| + |x| \leq |x|^a + a + |x| \leq |x|^{a+1} + a.$$

Thus, by Observation 10, computing $f_a(\langle a, y, y' \rangle 0)$ is possible in time

$$O((|x|^{a+1} + a)^{3a}) \subseteq O(|x|^{a^2+3a}) \subseteq O(|x|^{3a^2+6a}) \subseteq O(|x|^{9a^2}).$$

The value $F_b(\langle a, y, y' \rangle 0)$ can be computed in time

$$\begin{aligned} O(|b| \cdot ((|x|^{a+1} + a)^b + b) \log((|x|^{a+1} + a)^b + b)) &\subseteq O(|b| \cdot (|x|^{a+2})^b + b) \\ &\quad \cdot \log((|x|^{a+2})^b + b) \\ &\subseteq O(|b| \cdot (|x|^{ab+2b} + b) \log(|x|^{ab+2b} + b)) \\ &\subseteq O(|b| \cdot (|x|^{ab+2b+1}) \log(|x|^{ab+2b+1})) \\ &\subseteq O(|b| \cdot |x|^{2ab+4b+1}) \subseteq O(|x|^{2ab+4b+2}). \end{aligned}$$

In particular $|F_b(\langle a, y, y' \rangle 0)| \in O(|x|^{2ab+4b+2})$ and hence the computation of $f(F_b(\langle a, y, y' \rangle 0))$ is possible in time $O(|x|^{2ab+4b+2})$. We simplify the sum of these runtimes.

$$O(|x|^{3a} + |x|^{9a^2} + |x|^{2ab+4b+2}) \subseteq O(|x|^{9a^2b}) \quad \blacktriangleleft$$

▷ **Claim 16.** Let $a \in \mathbb{N}$ and $y' \in \Sigma^*$, such that $F_a(y') = \langle i, j, x', 0^p \rangle \notin \overline{D'}$. Then for all $b \in \mathbb{N}$ it holds that $\langle a, b, y' \rangle \in L(B)$.

Proof. By Claim 13 there is a $y \in \Sigma^p$ such that $f_a(\langle a, y, y' \rangle 0) \notin D = \text{img}(f)$. Thus, B accepts in step (iv), because $\text{img}(f) = D$. \blacktriangleleft

▷ **Claim 17.** Let $a \in \mathbb{N}$, such that $\text{img}(F_a) \subseteq \overline{D'}$. Then, there is some $b \in \mathbb{N}$, such that for all $y' \in \Sigma^*$ it holds that $\langle a, b, y' \rangle \notin L(B)$.

Proof. By Claim 14 and Observation 12, we know that f_a is a proof system for D . Since f is a p-optimal proof system for D , there exists some $b \in \mathbb{N}$, such that for all $\hat{x} \in \Sigma^*$ it holds that $f_a(\hat{x}) = f(F_b(\hat{x}))$. Thus, the computation $B(\langle a, b, y' \rangle)$ cannot accept in step (iv) independent of y' . Hence, the machine rejects. \blacktriangleleft

Now, we define a function $g_{a,b}$ for every pair of possible proof system F_a and possible simulation function F_b . Similarly to f_a , the function $g_{a,b}$ trusts that $B(\langle a, b, y' \rangle)$ accepts for all y' (note that the accepting behavior of B has influence on D). If F_a is a proof system for $\overline{D'}$, then there is a $b \in \mathbb{N}$ such that $g_{a,b}$ is a proof system for D because the machine B accepts on input a, b, y' for all $y' \in \Sigma^*$. For $F_a(y') \notin \overline{D'}$, we know there is no $b \in \mathbb{N}$ such that B accepts on input a, b, y' . Hence, the corresponding output of $g_{a,b}$ is not in D .

Let b_1 be the index of the NTM B , that is $N_{b_1} = B$. Furthermore, let $k_B \in \mathbb{N}^+$ be a constant such that $B(x)$ runs in time $k_B|x|^{9a^2b} + k_B$ for all $x \in \Sigma^*$. Recall that N_1 always accepts. We define a function $g : \mathbb{N} \rightarrow \mathbb{N}$ whose input consists of two indices $a, b \in \mathbb{N}$ of polynomial-time functions F_a, F_b and a proof $y' \in \Sigma^*$.

$$g(x) = \begin{cases} \langle 1, b_1, \langle a, b, y' \rangle, 0^{k_B|x|^{9a^2b} + k_B} \rangle & \text{if } x = \langle a, b, y' \rangle 0 \\ f(x') & \text{if } x = x' 1 \\ f(0) & \text{else} \end{cases}$$

► **Observation 18.** $g(x)$ runs in $O(|x|^{9a^2b})$ time for $x = \langle a, b, y' \rangle 0$.

Proof. Checking whether the input is formatted correctly is possible in linear time. Furthermore, the output with exception of the last entry of the list can be computed in linear time. The string $0^{k_B|x|^{9a^2b} + k_B}$ can be computed in $O(|x|^{9a^2b})$ time. \blacktriangleleft

44:10 Upward Translation of (P-)Optimal Proof Systems in the Boolean Hierarchy over NP

We define a function $g_{a,b} : \mathbb{N} \rightarrow \mathbb{N}$ that is obtained from g by fixing two indices $a, b \in \mathbb{N}$ of polynomial-time functions F_a, F_b .

$$g_{a,b}(x) = \begin{cases} g(\langle a, b, y' \rangle 0) & \text{if } x = \langle a, b, y' \rangle 0 \\ f(x') & \text{if } x = x' 1 \\ f(0) & \text{else} \end{cases}$$

► **Observation 19.** For $x = \langle a, b, y' \rangle 0$ it holds that $g(x) = g_{a,b}(x)$.

► **Observation 20.** For fixed $a, b \in \mathbb{N}$ the function $g_{a,b}(x)$ can be computed in polynomial time.

Proof. This follows directly from Observation 18 and the linear runtime of f . ◀

▷ **Claim 21.** Let $a \in \mathbb{N}$ and $y' \in \Sigma^*$ such that $F_a(y') = \langle i, j, x', 0^p \rangle \notin \overline{D'}$. Then for all $b \in \mathbb{N}$ it holds that $g_{a,b}(\langle a, b, y' \rangle 0) \notin D$.

Proof. By Claim 16 we know that for all $b \in \mathbb{N}$ the computation $B(\langle a, b, y' \rangle)$ accepts. This implies $g_{a,b}(\langle a, b, y' \rangle 0) = \langle 1, b_1, \langle a, b, y' \rangle, 0^{k_B |x|^{9a^2b} + k_B} \rangle \notin D$. ◀

▷ **Claim 22.** Let $a \in \mathbb{N}$ such that $\text{img}(F_a) \subseteq \overline{D'}$. Then there is some $b \in \mathbb{N}$ with $\text{img}(g_{a,b}) = D$.

Proof. Choose $b \in \mathbb{N}$ according to Claim 17. Then $D = \text{img}(f) \subseteq \text{img}(g_{a,b})$ because f is a proof system for D . Let $x \in \Sigma^*$. In the bottom two cases of $g_{a,b}$ and g we have $g_{a,b}(x) \in D$ and $g(x) \in D$. So we can assume $x = \langle a, b, y' \rangle 0$ and $g_{a,b}(x) = \langle 1, b_1, \langle a, b, y' \rangle, 0^{k_B |x|^{9a^2b} + k_B} \rangle$. By the choice of b , it holds $\langle a, b, y' \rangle \notin L(B)$. By Observation 15 and the choice of k_B , $B(\langle a, b, y' \rangle)$ runs in time $k_B |\langle a, b, y' \rangle|^{9a^2b} + k_B \leq k_B |x|^{9a^2b} + k_B$. Therefore, $\langle 1, b_1, \langle a, b, y' \rangle, 0^{k_B |x|^{9a^2b} + k_B} \rangle \in D$ and hence $g_{a,b}(x) \in D$. This shows $\text{img}(g_{a,b}) \subseteq D$. ◀

Finally, we define the p-optimal proof system h for $\overline{D'}$. The key difficulty is that h wants to output $F_a(y')$ for all a and y' using a short proof only when F_a is a proof system for $\overline{D'}$. To do this h must be able to check this property efficiently. We can do this as follows: if $f(F_c(\langle a, b, y' \rangle)) = g_{a,b}(\langle a, b, y' \rangle)$, then we output $F_a(y')$ and otherwise some arbitrary word from $\overline{D'}$. If $F_a(y') \notin \overline{D'}$, we know that there is no $b \in \mathbb{N}$ such that the corresponding output of $g_{a,b}$ is in D and the check correctly fails and $F_a(y')$ is not outputted. By contraposition we observe that we output $F_a(y')$ only if it is in $\overline{D'}$. Hence, h is a proof system for $\overline{D'}$. Lastly, we show that h p-simulates all proof systems for $\overline{D'}$. Let F_a be an arbitrary proof system for $\overline{D'}$. Then there is a $b \in \mathbb{N}$ such that $g_{a,b}$ is a proof system for D . Let $c \in \mathbb{N}$ be such that f p-simulates $g_{a,b}$ with the function F_c . So, for all $y' \in \mathbb{N}$ the function h outputs $F_a(y')$ for the input to h corresponding to a, b, c, y' . Also this input is short in a, b, c, y' and can be computed in polynomial time in these parameters.

Let $h' : \mathbb{N} \rightarrow \mathbb{N}$ be a linear time proof system for $\overline{D'}$. We define a function $h : \mathbb{N} \rightarrow \mathbb{N}$.

$$h(x) = \begin{cases} \langle i, j, x', 0^p \rangle & \text{if } x = \langle a, b, c, \langle a, b, y' \rangle 0, 0^{k_B \cdot |\langle a, b, y' \rangle|^{9a^2b} + k_B}, 0^{|c| \cdot (|\langle a, b, y' \rangle|^{c+c^2})} 0 \wedge \\ & f(F_c(\langle a, b, y' \rangle 0)) = g_{a,b}(\langle a, b, y' \rangle 0) \wedge \\ & F_a(y') = \langle i, j, x', 0^p \rangle \\ h'(x') & \text{if } x = x' 1 \\ h'(0) & \text{else} \end{cases}$$

► **Observation 23.** $h(x)$ runs in time $O(|x|)$.

Proof. The bottom two cases are trivial. For the first case we observe that checking, whether the input is formatted correctly, can be done in linear time. The part $0^{k_B \cdot \langle a, b, y' \rangle 0^{9a^2b + k_B}}$ can be checked in linear time by iterated multiplication. The computation $F_a(y')$ can be simulated in $O(|a| \cdot (|y'|^a + a) \log(|y'|^a + a)) \subseteq O((|y'|^a + a)^3) \subseteq O(|y'|^{3a}) \subseteq O(|x|)$ time. The computation $f(F_c(\langle a, b, y' \rangle 0))$ can be simulated in $O(|c| \cdot (|\langle a, b, y' \rangle 0|^c + c) \log(|\langle a, b, y' \rangle 0|^c + c)) \subseteq O(|x|)$ time and $g_{a,b}(\langle a, b, y' \rangle 0)$ can be simulated in $O(|\langle a, b, y' \rangle 0|^{9a^2b}) \subseteq O(|x|)$ time by Observation 18. The output $\langle i, j, x', 0^p \rangle$ can be computed in $O(|x|)$ time because all of its elements have been computed in the steps analyzed above. ◀

▷ **Claim 24.** h is a proof system for $\overline{D'}$.

Proof. We have $h \in \text{FP}$ by Observation 23. $\overline{D'} \subseteq \text{img}(h)$, since $\text{img}(h') \subseteq \text{img}(h)$ and h' is a proof system for $\overline{D'}$. We show $\text{img}(h) \subseteq \overline{D'}$ by contradiction. Assume that there exists $x \in \Sigma^*$ such that $h(x) \notin \overline{D'}$. The last two cases in the definition of h give values obviously in $\overline{D'}$. Thus, we only look at the first case. In particular $F_a(y') = \langle i, j, x', 0^p \rangle$ and $g_{a,b}(\langle a, b, y' \rangle 0) = f(F_c(\langle a, b, y' \rangle 0))$. The second implies directly $g_{a,b}(\langle a, b, y' \rangle 0) \in \text{img}(f) = \overline{D'}$. Since $h(x) = F_a(y')$ in this case, by assumption $F_a(y') \notin \overline{D'}$. By Claim 21 we conclude the contradiction $g_{a,b}(\langle a, b, y' \rangle 0) \notin \overline{D'}$. ◀

▷ **Claim 25.** Let $a \in \mathbb{N}$ with $\text{img}(F_a) \subseteq \overline{D'}$. Then there exist $b, c \in \mathbb{N}$, such that

$$\forall y' \in \Sigma^* : F_a(y') = \langle i, j, x', 0^p \rangle = h(\langle a, b, c, \langle a, b, y' \rangle 0, 0^{k_B \cdot |y'|^{9a^2b + k_B}}, 0^{|c| \cdot (|\langle a, b, y' \rangle 0|^c + c)^2} \rangle 0)$$

Proof. Claim 22 shows that there is some $b \in \mathbb{N}$ such that $\text{img}(g_{a,b}) = D$. By Observation 20 this $g_{a,b}$ is a proof system for D . Since f is p-optimal, there exists $c \in \mathbb{N}$ such that $f(F_c(x)) = g_{a,b}(x)$ for all $x \in \Sigma^*$. Let $y' \in \Sigma^*$. From $\text{img}(F_a) \subseteq \overline{D'}$ it follows $F_a(y') = \langle i, j, x', 0^p \rangle$ for suitable i, j, x', p . Hence, in Claim 25 we are always in the first case of h . It follows

$$h(\langle a, b, c, \langle a, b, y' \rangle 0, 0^{k_B \cdot |y'|^{9a^2b + k_B}}, 0^{|c| \cdot (|\langle a, b, y' \rangle 0|^c + c)^2} \rangle 0) = \langle i, j, x', 0^p \rangle.$$

This shows Claim 25. ◀

Let $a \in \mathbb{N}$ be arbitrary such that F_a is a proof system for $\overline{D'}$. Choose b, c according to Claim 25. Then the following $z : \mathbb{N} \rightarrow \mathbb{N}$ shows $F_a \leq^p h$.

$$z(y') = \langle a, b, c, \langle a, b, y' \rangle 0, 0^{k_B \cdot |y'|^{9a^2b + k_B}}, 0^{|c| \cdot (|\langle a, b, y' \rangle 0|^c + c)^2} \rangle 0$$

By Claim 25 it holds $F_a(y') = h(z(y'))$. The function z can be computed in polynomial time, because $a, b, c \in \mathbb{N}$ and $k_B \in \mathbb{N}^+$ are constant values for a fixed F_a . This proves that h is a p-optimal proof system for $\overline{D'}$. ◀

► **Corollary 26.** *If DP has p-optimal proof systems, coDP has p-optimal proof systems.*

Proof. Since $D \in \text{DP}$, we obtain that there is a p-optimal proof system for D . Theorem 7 shows that it follows that there is a p-optimal proof system for $\overline{D'}$. The language $\overline{D'}$ is \leq_m^p -hard for coDP. By Corollary 3, there are p-optimal proof systems for coDP. ◀

4 Oracle Construction

Corollary 26 naturally leads to the question of whether optimal proof systems for DP translate to optimal proof systems for coDP. We show that a proof for this translation cannot be relativizable, i.e., we cannot expect to show this translation with the currently available means. This result is a consequence of the following theorem:

► **Theorem 27.** *There exists an oracle O with the following properties:*

1. $\overline{C^O}$ has p -optimal proof systems (implying p -optimal proof systems for coNP).
2. $\overline{D^O}$ has no optimal proof systems (ruling out optimal proof systems for coDP).

Sketch of the construction. Work towards 1 (coding): For all $a \in \mathbb{N}$, the construction tries to achieve that $\text{img}(F_a^O) \neq \overline{C^O}$ and thus, F_a^O is no proof system for $\overline{C^O}$. If this is not possible, we start to *encode* the mappings of F_a (i.e., on which input it gives which output) into the oracle. For the coding, we define code words of the form $c(a, x, y) := \langle 0^a, 0^{4(|x|^a + a + |y|)}, x, y \rangle$ for $a \in \mathbb{N}$, $x, y \in \Sigma^*$. The purpose of a code word $c(a, x, y)$ is to encode the computation $F_a(x) = y$. Thus, the final oracle O will contain the encoded mappings of all proof systems for $\overline{C^O}$. The crucial point is that such a code word lets us recognize that F_a is a proof system for $\overline{C^O}$ and $y \in \overline{C^O}$. This allows us to define a p -optimal proof system h which is able to simulate every proof system for $\overline{C^O}$ using oracle queries.

Work towards 2 (diagonalization): For all $b \in \mathbb{N}$, the construction tries to achieve that $\text{img}(G_b^O) \neq \overline{D^O}$ and thus, G_b^O is no proof system for $\overline{D^O}$. If this is not possible, we define some proof system z_b^O for $\overline{D^O}$ and show that z_b^O cannot be simulated by G_b^O . The latter is achieved by diagonalizing against every simulation function π , i.e., we make sure that G_b^O does not simulate z_b^O via π .

We call the functions z_b^O witness proof systems. The intuition behind their definition and behind the whole diagonalization is as follows: independent of the oracle each function z_b^O has short proofs for all elements of some polynomial-time-decidable set. But our construction offers the freedom to choose whether or not this set is a subset of $\overline{D^O}$. The latter depends on the following language

$$\begin{aligned} A^O &= \{x \in \Sigma^* \mid |O^{|x|}|_c \geq 2\} \cup \{x \in \Sigma^* \mid |O^{|x|}|_c = 0\} \\ &= \{x \in \Sigma^* \mid |O^{|x|}|_c \neq 1\}, \end{aligned}$$

which lies inside coDP^O and thus has influence on $\overline{D^O}$. Let y be a word whose membership to $\overline{D^O}$ is influenced by the question of whether $0^n \in A^O$. Observe that $0^n \in A^O$ if and only if $|O|_c^n \neq 1$. Thus, we can control the membership of y to $\overline{D^O}$ by adding none, one or more words of length n to O . There are 2^n such words. Let G_b^O be some proof system for $\overline{D^O}$. During the construction of O , we initially have no word of length n inside O and thus $y \in \overline{D^O}$ and G_b^O must have a proof for y . Case 1: All G_b^O -proofs for y are long. When G_b^O is given such a proof it can determine by exhaustive search the number of words of length n in O . However, G_b^O does not simulate z_b^O , because z_b^O has short proofs for y , but G_b^O has not. Case 2: G_b^O has a short proof x for y . In this case, $G_b^O(x)$ cannot query all 2^n words and hence cannot determine whether $y \in \overline{D^O}$. We can exploit this to create a situation where $G_b^O(x)$ outputs an element outside $\overline{D^O}$ and hence is no proof system for this set. So G_b^O can either simulate z_b^O or be a proof system for $\overline{D^O}$, but not both at once.

The main challenge of the oracle construction is to combine the work for 1 and 2, because the code words interact with the diagonalization. Indeed, in the example above G_b^O cannot query all 2^n words when having a short proof x for y , but there are many code words that can be queried by $G_b^O(x)$ whose memberships together can depend on all 2^n words of length n . We capture these dependencies in a graph data structure, where nodes are words from Σ^* and edges are oracle queries of underlying FP-computations of code words, i.e., for a code word $c(a, x, y)$ the computation $F_a^O(x)$. This helps to identify words of length n that are independent of the computation $G_b^O(x)$ and all queried code words.

5 Conclusion

We summarize all results to obtain the equivalence classes from Figure 1. First observe that (p-)optimal proof systems always translate from a class \mathcal{C} to \mathcal{D} when $\mathcal{C} \subseteq \mathcal{D}$ (respective solid arrows are omitted in Figure 2). We start with the equivalence classes for p-optimal proof systems (see Figure 2, left, solid arrows). P-optimal proof systems translate as follows:

- from $\text{NP} \cup \text{coNP}$ to DP by $\text{NP} \cup \text{coNP} \supseteq \text{NP}$, $\text{NP} \cup \text{coNP} \supseteq \text{coNP}$, $\text{NP} \wedge \text{coNP} = \text{DP}$, and Corollary 5 following from Köbler, Messner, and Torán [18].
- from DP to coDP by Corollary 26.
- from coBH_k to coBH_{k+1} for $k \geq 2$ by Corollary 5 following from Köbler, Messner, and Torán [18] and the following inclusions:

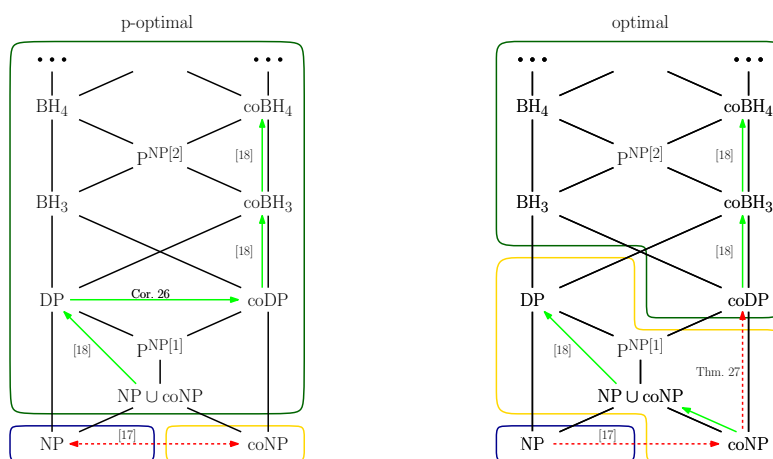
$$\begin{aligned} \text{coBH}_k \wedge \text{coBH}_k &\supseteq \text{coDP} \wedge \text{coBH}_k = (\text{NP} \vee \text{coNP}) \wedge \text{coBH}_k \\ &\supseteq \begin{cases} \text{coNP} \wedge \text{coBH}_k = \text{coBH}_{k+1} & \text{if } k \text{ is even} \\ \text{NP} \vee \text{coBH}_k = \text{coBH}_{k+1} & \text{else} \end{cases} \end{aligned}$$

Next, we derive the equivalence classes for optimal proof systems (see Figure 2, right, solid arrows). Optimal proof systems translate as follows:

- from coNP to $\text{NP} \cup \text{coNP}$ by the fact that NP has optimal proof systems.
- from $\text{NP} \cup \text{coNP}$ to DP by Corollary 5 following from Köbler, Messner, and Torán [18].
- from coBH_k to coBH_{k+1} for $k \geq 2$ by the same argument used for p-optimal proof systems.

The resulting equivalence classes for (p-)optimal proof systems are different (see Figure 2, left and right, dashed arrows) relative to oracles A, B, O with the following properties:

- NP^A has p-optimal proof systems and coNP^A has no optimal proof systems [17].
- coNP^B has p-optimal proof systems and NP^B has no p-optimal proof systems [17].
- coNP^O has p-optimal proof systems and coDP^O has no optimal proof systems (Thm. 27).



■ **Figure 2** Equivalence classes for p-optimal proof systems (left) and optimal proof systems (right) in the Boolean hierarchy over NP and the bounded query hierarchy over NP . Green solid arrows from A to B mean that (p-)optimal proof systems for A imply (p-)optimal proof systems for B . Red dashed arrows from A to B mean that there exists an oracle Q relative to which A^Q has (p-)optimal proof systems and B^Q has no (p-)optimal proof systems. Note that green solid arrows pointing downwards are omitted, since those are trivial and only the minimum number of required red dashed arrows to separate all equivalence classes are drawn.

Oracle A rules out translations from NP to any other class in Figure 2 for optimal and p-optimal proof systems. Oracle B rules out translations from coNP to NP and thus also to $\text{NP} \cup \text{coNP}$ for p-optimal proof systems. Oracle O rules out translations from coNP to coDP for optimal proof systems.

We obtain the following connection to a conjecture studied by Pudlák [25].

► **Corollary 28.** *The following statements are equivalent:*

- BH has no p-optimal proof system.
- TAUT has no p-optimal proof systems or SAT has no p-optimal proof systems (i.e., $\text{CON} \vee \text{SAT}$ in Pudlák's notation).

Proof. Figure 2 shows that $\text{NP} \cup \text{coNP}$ and BH are equivalent with respect to p-optimal proof systems. Hence, BH has no p-optimal proof systems if and only if $\text{NP} \cup \text{coNP}$ has no p-optimal proof systems. The latter holds if and only if TAUT has no p-optimal proof systems or SAT has no p-optimal proof systems. ◀

References

- 1 T. P. Baker, J. Gill, and R. Solovay. Relativizations of the $\text{P} = ? \text{NP}$ question. *SIAM Journal on Computing*, 4(4):431–442, 1975. doi:10.1137/0204037.
- 2 Richard Beigel. Bounded queries to sat and the boolean hierarchy. *Theoretical Computer Science*, 84(2):199–223, 1991. doi:10.1016/0304-3975(91)90160-4.
- 3 O. Beyersdorff. Representable disjoint NP-pairs. In *Proceedings 24th International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2004. doi:10.1007/978-3-540-30538-5_11.
- 4 O. Beyersdorff. Disjoint NP-pairs from propositional proof systems. In *Proceedings of Third International Conference on Theory and Applications of Models of Computation*, volume 3959 of *Lecture Notes in Computer Science*, pages 236–247. Springer, 2006. doi:10.18452/15520.
- 5 O. Beyersdorff. Classes of representable disjoint NP-pairs. *Theoretical Computer Science*, 377(1-3):93–109, 2007. doi:10.1016/j.tcs.2007.02.005.
- 6 O. Beyersdorff. The deduction theorem for strong propositional proof systems. *Theory of Computing Systems*, 47(1):162–178, 2010. doi:10.1007/s00224-008-9146-6.
- 7 O. Beyersdorff, J. Köbler, and J. Messner. Nondeterministic functions and the existence of optimal proof systems. *Theoretical Computer Science*, 410(38-40):3839–3855, 2009. doi:10.1016/j.tcs.2009.05.021.
- 8 O. Beyersdorff and Z. Sadowski. Do there exist complete sets for promise classes? *Mathematical Logic Quarterly*, 57(6):535–550, 2011. doi:10.1002/malq.201010021.
- 9 S. R. Buss. Lectures on proof theory. Technical Report SOCS 96.1, McGill University, 1996.
- 10 S. Cook and J. Krajíček. Consequences of the provability of $\text{NP} \subseteq \text{P/poly}$. *Journal of Symbolic Logic*, 72(4):1353–1371, 2007. doi:10.2178/jsl/1203350791.
- 11 S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979. doi:10.2307/2273702.
- 12 M. I. Dekhtyar. On the relativization of deterministic and nondeterministic complexity classes. In *Mathematical Foundations of Computer Science*, volume 45 of *Lecture Notes in Computer Science*, pages 255–259, Berlin, 1976. Springer-Verlag.
- 13 C. Glaßer, A. L. Selman, and L. Zhang. Canonical disjoint NP-pairs of propositional proof systems. *Theoretical Computer Science*, 370:60–73, 2007. doi:10.1007/11549345_35.
- 14 C. Glaßer, A. L. Selman, and L. Zhang. The informational content of canonical disjoint NP-pairs. *International Journal of Foundations of Computer Science*, 20(3):501–522, 2009. doi:10.1007/978-3-540-73545-8_31.

- 15 E. A. Hirsch. Optimal acceptors and optimal proof system. In J. Kratochvíl, A. Li, J. Fiala, and P. Kolman, editors, *Theory and Applications of Models of Computation*, pages 28–39. Springer Berlin Heidelberg, 2010.
- 16 E. A. Hirsch and D. Itsykson. On optimal heuristic randomized semidecision procedures, with application to proof complexity. In J. Marion and T. Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science*, volume 5, pages 453–464. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010. doi:10.4230/LIPIcs.STACS.2010.2475.
- 17 E. Khaniki. New relations and separations of conjectures about incompleteness in the finite domain. *The Journal of Symbolic Logic*, 87(3):912–937, 2022. doi:10.1017/js1.2021.99.
- 18 J. Köbler, J. Messner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184(1):71–92, 2003. doi:10.1016/S0890-5401(03)00058-0.
- 19 J. Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995. doi:10.1017/CB09780511529948.
- 20 J. Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019. doi:10.1017/9781108242066.
- 21 J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic*, 54:1063–1079, 1989. doi:10.2307/2274765.
- 22 J. Messner. *On the Simulation Order of Proof Systems*. PhD thesis, Universität Ulm, 2000.
- 23 C. H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981. doi:10.1145/322276.322287.
- 24 P. Pudlák. On the lengths of proofs of consistency. In *Collegium Logicum*, pages 65–86. Springer Vienna, 1996. doi:10.1016/S0049-237X(08)70462-2.
- 25 P. Pudlák. Incompleteness in the finite domain. *The Bulletin of Symbolic Logic*, 23(4):405–441, 2017.
- 26 P. Pudlák. The lengths of proofs. In Samuel R. Buss, editor, *Handbook of Proof Theory*, pages 547–637. Elsevier, Amsterdam, 1998.
- 27 A. Razborov. On provably disjoint NP-pairs. *BRICS Report Series*, 36, 1994. doi:10.7146/brics.v1i36.21607.
- 28 Z. Sadowski. On an optimal propositional proof system and the structure of easy subsets of TAUT. *Theoretical Computer Science*, 288(1):181–193, 2002. doi:10.1016/S0304-3975(01)00155-4.
- 29 T. Pitassi and R. Santhanam. Effectively polynomial simulations. In *Innovations in Computer Science*, pages 370–381, 2010.
- 30 K. W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1):53–80, 1987. doi:10.1016/0304-3975(87)90049-1.