Is the Standard Proof System for SAT P-optimal?
(Extended Abstract)

Johannes Köbler[1] and Jochen Messner[2]

[1] Institut für Informatik, Humboldt-Universität zu Berlin, 10099 Berlin, Germany.
koebler@informatik.hu-berlin.de
[2] Abteilung Theoretische Informatik, Universität Ulm, 89069 Ulm, Germany.
messner@informatik.uni-ulm.de

Abstract. We investigate the question whether there is a (p-)optimal proof system for SAT or for TAUT and its relation to completeness and collapse results for nondeterministic function classes. A p-optimal proof system for SAT is shown to imply (1) that there exists a complete function for the class of all total nondeterministic multi-valued functions and (2) that any set with an optimal proof system has a p-optimal proof system. By replacing the assumption of the mere existence of a (p-)optimal proof system by the assumption that certain proof systems are (p-)optimal we obtain stronger consequences, namely collapse results for various function classes. Especially we investigate the question whether the standard proof system for SAT is p-optimal. We show that this assumption is equivalent to a variety of complexity theoretical assertions studied before, and to the assumption that every optimal proof system is p-optimal. Finally, we investigate whether there is an optimal proof system for TAUT that admits an effective interpolation, and show some relations between various completeness assumptions.

1 Introduction and Overview

Following Cook and Reckhow [3] we define the notion of an abstract proof system for a set $L \subseteq \{0,1\}^*$ as follows. A (possibly partial) polynomial-time computable function $h : \{0,1\}^* \to \{0,1\}^*$ with range $L = \{h(x) \mid x \in \{0,1\}^*\}$ is called a *proof system for L*. In this setting, an *h-proof* for the membership of φ to L is given by a string w with $h(w) = \varphi$. In order to compare the relative strength of different proof systems for the set TAUT of all propositional tautologies, Cook and Reckhow introduced the notion of p-simulation. A proof system h *p-simulates* a proof system g if g-proofs can be translated into h-proofs in polynomial time, i.e., there is a polynomial-time computable function f such that for each v in the domain of g, $h(f(v)) = g(v)$. Similarly, h is said to *simulate* g if for each g-proof v there is an h-proof w of length polynomial in the length of v with $h(w) = g(v)$. A proof system for a set L is called *(p-)optimal* if it (p-)simulates every proof system for L (cf. [12]). It's a natural question whether a set L has a p-optimal (or at least an optimal) proof system. Note that a p-optimal proof system has the advantage that from any proof in an other proof system one can efficiently obtain a proof for the same instance in the p-optimal proof system.

Hence, any method that is used to compute proofs in some proof system can be reformulated to yield proofs in the p-optimal proof system with little overhead.

It is observed in [19, 16, 11] that (p-)optimal proof systems for certain languages can be used to define complete sets for certain promise classes. For example, if TAUT has an optimal proof system then $\mathcal{NP} \cap \mathit{Sparse}$ has a many-one complete set, and if TAUT as well as SAT have a p-optimal proof system then $\mathcal{NP} \cap \mathit{co}\text{-}\mathcal{NP}$ has a complete set. We complete this picture here by showing that already a p-optimal proof system for SAT can be used to derive completeness consequences.

These results are however unsatisfactory in so far as they provide only necessary conditions for the existence of (p-)optimal proof systems. It appears that a much stronger assumption like $\mathcal{NP} = \mathcal{P}$ is needed to derive a p-optimal proof system for TAUT (actually, a somewhat weaker collapse condition suffices, namely that all tally sets in nondeterministic double exponential time are contained in deterministic double exponential time; see [11]). If, however, we consider proof systems with certain additional properties then we can indeed derive collapse consequences from the assumption that these proof systems are (p-)optimal. We consider two examples:

The best known proof system is probably the standard proof system for SAT where proofs are given by a satisfying assignment for the formula in question. Adapting this proof method to the current setting one obtains the following natural proof system for SAT:

$$\mathit{sat}(x) = \begin{cases} \varphi & \text{if } x = \langle \alpha, \varphi \rangle \text{ and } \alpha \text{ is a satisfying assignment for } \varphi \\ \text{undef.} & \text{otherwise.} \end{cases}$$

We consider the question whether *sat* is p-optimal[1] in Sect. 3. It turns out that the assumption of *sat* being p-optimal is equivalent to a variety of well studied complexity theoretic assumptions (which have unlikely collapse consequences as, e.g., that $\mathcal{NP} \cap \mathit{co}\text{-}\mathcal{NP} = \mathcal{P}$). Most of these assumptions were listed in [5] under "Proposition Q" (see also [6]). Proposition Q states for example that any function in the class $\mathcal{NPMV}_t$ of total multi-valued functions computable in nondeterministic polynomial time has a refinement in $\mathcal{FP}$. We further add to this list the statement that every optimal proof system is p-optimal.

As a second example, we consider in Sect. 4 proof systems that admit an effective interpolation. Due to Craig's Interpolation Theorem for Propositional Logic, for any tautology $\varphi \to \psi$ there is a formula ϕ that uses only common variables of φ and ψ such that $\varphi \to \phi$ and $\phi \to \psi$. A circuit C that computes the same function as ϕ is called an *interpolant* of $\varphi \to \psi$. Following [13] we say that a proof system h for TAUT *admits an effective interpolation* if there is a polynomial p such that for any h-proof w of a formula $h(w) = \varphi \to \psi$, the formula $\varphi \to \psi$ has an interpolant of size at most $p(|w|)$. We show that if TAUT has an optimal proof system with this property then any function in $\mathcal{NPSV}$ (the class of single valued functions computable in nondeterministic polynomial time)

[1] Pavel Pudlák posed this question during the discussion after Zenon Sadowski's talk at CSL'98 [20].

has a total extension in $\mathcal{FP}/poly$. The latter is equivalent to the statement that every disjoint pair of $\mathcal{NP}$-sets is $\mathcal{P}/poly$-separable which in turn implies that $\mathcal{NP} \cap co\text{-}\mathcal{NP} \subseteq \mathcal{P}/poly$ and that $\mathcal{UP} \subseteq \mathcal{P}/poly$.

The (likely) assumption that there are no p-optimal proof systems for SAT (as well as for TAUT) also has some practical implications due to its connection to the existence of optimal algorithms (see [12, 20, 15]). Note that usually a decision algorithm for SAT also provides a satisfying assignment for any positive instance. However, if *sat* is not p-optimal then there is a set $S \subseteq$ SAT of easy instances (i.e. $S \in \mathcal{P}$) for some of which it is hard to produce a satisfying assignment (i.e., there is no polynomial time algorithm that produces a satisfying assignment on all inputs from S, cf. Theorem 1). In fact, a stronger consequence can be derived: if *sat* is not p-optimal then there is a non-sparse set of easy instances from SAT for which it is hard to produce a satisfying assignment (see Theorem 5).

The observations from [19, 16, 11] that a p-optimal proof system for a set L implies the existence of a complete set for a certain promise class in fact shows a relationship between different completeness assumptions. Since the definition of p-simulation is equivalent to the definition of many-one reducibility between functions (in the sense of [11]), a proof system for L is p-optimal if and only if it is many-one complete for the (promise) function class $\mathcal{PS}_L$ that consists of all proof systems for L. Depending on the complexity of L, this completeness assumption can be used to derive the existence of complete sets for various other promise classes. This observation motivates us to further investigate whether there are relations between various completeness assumptions. Along this line of investigation we show in Sect. 5 that

- $\mathcal{NPSV}$ has a many-one complete function if and only if there is a strongly many-one complete disjoint $\mathcal{NP}$-pair,
- a complete function for $\mathcal{NPMV}_t$ implies a many-one complete pair for the class of disjoint $co\text{-}\mathcal{NP}$ pairs, and
- $\mathcal{NPSV}_t = \mathcal{NPMV}_t \cap \mathcal{NPSV}$ has a many-one complete function if and only if $\mathcal{NP} \cap co\text{-}\mathcal{NP}$ has a complete set.

The collapse consequences for the nondeterministic function classes $\mathcal{NPMV}_t$ and $\mathcal{NPSV}$ that are obtained in Sects 3, 4 from the assumption that *sat* is p-optimal, respectively that there is an optimal proof system for TAUT that admits an effective interpolation, are complemented by the following completeness consequences (presented in Sect. 6):

- If SAT has a p-optimal proof system then $\mathcal{NPMV}_t$ has a complete function.
- If TAUT has an optimal proof system then $\mathcal{NPSV}$ has a complete function.

Further we show that

- SAT has a p-optimal proof system if and only if any language with an optimal proof system also has a p-optimal proof system.

This result again complements the observation from Sect. 3 that *sat* is p-optimal if and only if every optimal proof system is p-optimal. As an application we can weaken the assumption used in [11] to show that $\mathcal{NP} \cap co\text{-}\mathcal{NP}$ has a complete set: it suffices to assume that SAT has a p-optimal proof system and TAUT has an optimal proof system.

Due to the limited space, several results are presented here without proof.

2 Preliminaries

Let $\Sigma = \{0, 1\}$. We denote the cardinality of a set A by $\|A\|$ and the length of a string $x \in \Sigma^*$ by $|x|$. A set S is called *sparse* if the cardinality of $S \cap \Sigma^n$ is bounded above by a polynomial in n. $\mathcal{FP}$ denotes the class of (partial) functions that can be computed in polynomial time. We use $\langle \cdot, \cdots, \cdot \rangle$ to denote a standard polynomial-time computable tupling function. The definitions of standard complexity classes like $\mathcal{P}$, $\mathcal{NP}$, etc. can be found in books like [1, 17]. For a class $\mathcal{C}$ of sets we call a pair (A, B) of disjoint sets $A, B \in \mathcal{C}$ a *disjoint* $\mathcal{C}$*-pair*. If for a class $\mathcal{D}$, and some $D \in \mathcal{D}$ it holds $A \subseteq D$, and $B \cap D = \emptyset$ we call the pair (A, B) $\mathcal{D}$*-separable*.

A *nondeterministic polynomial time Turing machine* (*NPTM*, for short) is a Turing machine N such that for some polynomial p, every accepting path of N on any input of length n is at most of length $p(n)$. A *nondeterministic transducer* is a nondeterministic Turing machine T with a write-only output tape. On input x, T outputs $y \in \Sigma^*$ (in symbols: $T(x) \mapsto y$) if there is an accepting path on input x along which y is written on the output tape. Hence, the function f computed by T on Σ^* could be multi-valued and partial. Using the notation of [2, 23] we denote the set $\{y \mid f(x) \mapsto y\}$ of all output values of T on input x by set-$f(x)$. $\mathcal{NPMV}$ denotes the class of all multi-valued, partial functions computable by some nondeterministic polynomial-time transducer. $\mathcal{NPSV}$ is the class of functions f in $\mathcal{NPMV}$ that are *single-valued*, i.e. $\|\text{set-}f(x)\| \leq 1$. (thus, a single-valued multi-valued function is a function in the usual sense, and we use $f(x)$ to denote the unique string in set-$f(x)$). The *domain* of a multi-valued function is the set of those inputs x where set-$f(x) \neq \emptyset$. A function is called total if its domain is Σ^*. For a function class $\mathcal{F}$ we denote by $\mathcal{F}_t$ the class of total functions in $\mathcal{F}$. We use $\mathcal{NPMV}_t \subseteq_c \mathcal{FP}$ to indicate that for any $g \in \mathcal{NPMV}_t$ there is a total function $f \in \mathcal{FP}$ that is a *refinement* of g, i.e. $f(x) \in$ set-$g(x)$ for all $x \in \Sigma^*$. We say that a multi-valued function h *many-one reduces* to a multi-valued function g if there is a function $f \in \mathcal{FP}$ such that for every $x \in \Sigma^*$ set-$g(f(x)) =$ set-$h(x)$.

For a function class $\mathcal{F}$ a function h is called $\mathcal{F}$*-invertible* if there is a function $f \in \mathcal{F}$ that *inverts* h, i.e. $h(f(y)) = y$ for each y in the range of h. A function h is *honest* if for some polynomial p, $p(|h(x)|) \geq |x|$ holds for all x in the domain of h. We call a function g an *extension* of a function f if $f(x) = g(x)$ for any x in the domain of f. A function $r : \mathbb{N} \to \mathbb{N}$ is called *super-polynomial* if for each polynomial p, $r(n) > p(n)$ for almost every $n \geq 0$. A set $B \in \mathcal{P}$ with $B \subseteq L$ is called a $\mathcal{P}$*-subset of* L.

3 Q and the p-optimality of *sat*

In [5] the following statements were all shown to be equivalent. There, Q is defined to be the proposition that one (and consequently each) of these statements is true. In this section we show that Q is also equivalent to the p-optimality of *sat*.

Theorem 1 ([5], cf. [6]). *The following statements are equivalent*

1. *For each NPTM N that accepts SAT there is a function $f \in \mathcal{FP}$ such that for each α encoding an accepting path of N on input φ, $f(\alpha)$ is a satisfying assignment of φ.*
2. *Each honest function $f \in \mathcal{FP}$ with range Σ^* is $\mathcal{FP}$-invertible.*
3. $\mathcal{NPMV}_t \subseteq_c \mathcal{FP}$.
4. *For all $\mathcal{P}$-subsets S of SAT there exists a function $g \in \mathcal{FP}$ such that for all $\varphi \in S$, $g(\varphi)$ is a satisfying assignment of φ.*

Clearly, each nondeterministic Turing machine N corresponds to a proof system h with $h(\alpha) = \varphi$ if α encodes an accepting path of N on input φ. Now h is honest if, and only if, N is a NPTM. This leads to the observation that Statement 1 in Theorem 1 is equivalent to the condition that *sat* p-simulates every honest proof system for SAT. Hence, we just need to delete the term 'polynomial-time' in the Statement 1 of Theorem 1 to obtain the desired result that Q is equivalent to the p-optimality of *sat*. That this is possible without changing the truth of the theorem can be shown by a padding argument.

Theorem 2. *The following statements are equivalent.*

1. *For each NPTM N that accepts SAT there is a function $f \in \mathcal{FP}$ such that for each α encoding an accepting path of N on input φ, $f(\alpha)$ is a satisfying assignment of φ.*
2. *For each nondeterministic Turing machine N that accepts SAT there is a function $f \in \mathcal{FP}$ such that for each α encoding an accepting path of N on input φ, $f(\alpha)$ is a satisfying assignment of φ.*
3. *sat is a p-optimal proof system for SAT.*

It is known that the assumption $\mathcal{NP} = \mathcal{P}$ implies $\mathcal{NPMV}_t \subseteq_c \mathcal{FP}$ which in turn implies $\mathcal{NP} \cap co\text{-}\mathcal{NP} = \mathcal{P}$ (cf. [24]). Also, in [9] it has been shown that the converse of these implications is not true in suitable relativized worlds. The consequence $\mathcal{NP} \cap co\text{-}\mathcal{NP} = \mathcal{P}$ also shows that the assumption that *sat* is p-optimal is presumably stronger than the assumption that SAT has a p-optimal proof system. Namely the p-optimality of *sat* implies that $\mathcal{NP} \cap co\text{-}\mathcal{NP} = \mathcal{P}$, whereas the existence of a p-optimal proof system follows already (see [11]) if any super-tally set in Σ_2^P belongs to $\mathcal{P}$ (here, any set $L \subseteq \{0^{2^{2^n}} \mid n \geq 0\}$ is called super-tally).

The assumption that *sat* is a p-optimal proof system also has an effect on various reducibility degrees, as has been mentioned in [5] for Karp and Levin reducibility. Also in [14] it is shown that $\mathcal{NPMV}_t \subseteq_c \mathcal{FP}$ if and only if γ-reducibility equals polynomial time many-one reducibility. Furthermore it is shown in [4] that Statement 4 of Theorem 1 is equivalent to the assumption that the approximation class APX is closed under L-reducibility (see [4] for definitions).

The equivalence between the p-optimality of *sat* and $\mathcal{NPMV}_t \subseteq_c \mathcal{FP}$ directly leads to a proof of the following theorem.

Theorem 3. *The following statements are equivalent.*

1. *sat is p-optimal.*
2. *For any language L and all proof systems h and g for L*
   h p-simulates g if and only if h simulates g
   (i.e., the corresponding quasi-orders coincide).
3. *Every optimal proof system is p-optimal.*

In [15] it is observed that given a p-optimal proof system h for a language L the problem to find an h-proof for $y \in L$ is not much harder than deciding L, i.e. we can transform each deterministic Turing machine M with $L(M) = L$ to a deterministic Turing machine M' that on input $y \in L$ yields an h-proof of y in $\mathrm{time}_{M'}(y) \leq p(|y| + \mathrm{time}_M(y))$, for some polynomial p determined by M. Using this observation and the equivalences in Theorem 1 and 2 we obtain the following result: *sat* is p-optimal if and only if any deterministic Turing machine M that accepts SAT can be converted to a deterministic Turing machine that computes a satisfying assignment for any formula $\varphi \in$ SAT and runs not much longer than M on input φ.

Theorem 4. *The following statements are equivalent.*

1. *sat is p-optimal.*
2. *For any deterministic Turing machine M that accepts SAT, there is a deterministic Turing machine M' and a polynomial p such that for every $\varphi \in SAT$, M' produces a satisfying assignment of φ in $time_{M'}(\varphi) \leq p(|\varphi| + time_M(\varphi))$ steps.*

Under the assumption that *sat* is not p-optimal it follows from Theorem 4 that there is a Turing machine M that decides SAT such that any machine M' that on input $\varphi \in$ SAT has to produce a satisfying assignment for φ is much slower on some SAT instances. In some sense this appears counterintuitive as probably all SAT algorithms used in praxis produce a satisfying assignment in case the input belongs to SAT. Of course it follows from Theorem 4 that M is superior to any such M' on an infinite set of instances. As shown in the following theorem there is a deterministic Turing machine M accepting SAT that is more than polynomially faster than any deterministic transducer M' that produces satisfying assignments on a fixed non-sparse set of SAT instances. The result is due to the paddability of SAT, its proof uses ideas from the theory of complexity cores (see [21]).

Theorem 5. *The following statements are equivalent.*

1. *sat is not p-optimal.*
2. *There is a $\mathcal{P}$-subset S of SAT (i.e. there is a Turing machine M accepting SAT that has a polynomial time-bound on instances from S), a non-sparse subset L of S, and a super-polynomial function f such that for any deterministic Turing machine M' that on input of any $\varphi \in L$ produces a satisfying assignment of φ it holds $time_{M'}(\varphi) > f(|\varphi|)$ for almost every $\varphi \in L$.*

4 Collapse of $\mathcal{NPSV}$ and effective interpolation

In [22] the following hypothesis (called H2) has been examined.

> For every polynomial time uniform family of formulas $\{\varphi_n, \psi_n\}$ such that for every n, φ_n and ψ_n have n common variables and $\varphi_n \to \psi_n$ is a tautology, there is a polynomial p and a circuit family $\{C_n\}$ where for each n, C_n is of size at most $p(n)$ has n inputs and is an interpolant of $\varphi_n \to \psi_n$.

Theorem 6 ([22]). *The following statements are equivalent.*

1. *H2.*
2. *Every disjoint pair of $\mathcal{NP}$-sets is $\mathcal{P}/poly$-separable.*
3. *Every function in $\mathcal{NPSV}$ has a total extension in $\mathcal{FP}/poly$.*

As mentioned in [22] this in turn implies that H2 implies $\mathcal{NP} \cap co\text{-}\mathcal{NP} \subseteq \mathcal{P}/poly$ and $\mathcal{UP} \subseteq \mathcal{P}/poly$.

In [5] and [6] also a statement called Q$'$ that is implied by Q is examined. Q$'$ is equivalent to the statement that all disjoint *co*-$\mathcal{NP}$-pairs are $\mathcal{P}$-separable. Thus as proposed in [22] H2 is a nonuniform version of the dual condition to Q', namely that every disjoint pair of $\mathcal{NP}$-sets is $\mathcal{P}$-separable.

It is observed in [13] that extended Frege proof systems do not admit an effective interpolation if the RSA cryptosystem is secure. Partly generalizing this observation, one can state that the existence of an honest injective function in $\mathcal{FP}$ that is not $\mathcal{FP}/poly$-invertible (i.e., a one-way function that is secure against $\mathcal{FP}/poly$) implies the existence of a proof system for TAUT that does not admit an effective interpolation. Notice that each injective function in $\mathcal{FP}$ is invertible by a $\mathcal{NPSV}$-function. Thus the assumption that each $\mathcal{NPSV}$ function has a total extension in $\mathcal{FP}/poly$ implies that every injective function is $\mathcal{FP}/poly$-invertible. As the former assumption (that is equivalent to H2 by Theorem 6) implies $\mathcal{NP} \cap co\text{-}\mathcal{NP} \subseteq \mathcal{P}/poly$ and the latter is equivalent to $\mathcal{UP} \subseteq \mathcal{P}/poly$ (cf. [10, 7]) it is presumably stronger. In the following Theorem we observe that H2, respectively the statement that every function in $\mathcal{NPSV}$ has a total extension in $\mathcal{FP}/poly$ is true if, and only if, every proof system for TAUT admits an effective interpolation.

Theorem 7. *The following statements are equivalent.*

1. *Every function in $\mathcal{NPSV}$ has a total extension in $\mathcal{FP}/poly$.*

2. *Every proof system for TAUT admits an effective interpolation.*
3. *For any set $S \subseteq TAUT$, $S \in \mathcal{NP}$, there is a polynomial p, such that any formula $\varphi \to \psi \in S$ has an interpolant of size at most $p(|\varphi \to \psi|)$.*

Proof. The implication *2* $\Longrightarrow$ H2 is easy to see, as for every polynomial time uniform family of tautologies $\varphi_n \to \psi_n$ one may define a proof system h for TAUT that has a short proof for any tautology of this family. Thus *2* $\Longrightarrow$ *1* using Theorem 6.

The proof of the implication *1* $\Longrightarrow$ *3* is obtained by extending an idea from [22] that was used to prove the implication *3* $\Longrightarrow$ *1* of Theorem 6. Let $S \subseteq$ TAUT, $S \in \mathcal{NP}$. Let f be a function such that for any formula $\varphi \in S$, $\varphi = \varphi_0(x, y) \to \varphi_1(x, z)$ (where x, y, z denote vectors of variables), it holds

$$f(\langle \alpha, \varphi \rangle) = \begin{cases} 1 & \text{if for some } \beta, \quad \varphi_0(\alpha, \beta) \text{ holds} \\ 0 & \text{if for some } \gamma, \ \neg\varphi_1(\alpha, \gamma) \text{ holds.} \end{cases}$$

Otherwise, and for any other input let f be undefined. First observe that f is well defined, i.e. that f is single valued. This is due to the fact that $\varphi = \varphi_0(x, y) \to \varphi_1(x, z) \in$ TAUT. Further, f can be computed by a nondeterministic machine N that first (in deterministic polynomial time) validates that the input is of the appropriate form $\langle \alpha, \varphi \rangle$, $\varphi = \varphi_0(x, y) \to \varphi_1(x, z)$. Then N guesses a certificate for $\varphi \in S$ and, if successful, guesses some string w. Now if w is of an appropriate length and if $\varphi_0(\alpha, w)$ holds then N outputs 1, if $\varphi_1(\alpha, w)$ holds, N outputs 0. Hence $f \in \mathcal{NPSV}$. Assuming *1*, f has a total extension in $\mathcal{FP}/poly$. Thus there is a polynomial p and for any $n \geq 0$ a circuit C_n of size at most $p(n)$ such that for any tuple $v = \langle \alpha, \varphi \rangle$ of length n in the domain of f, $C_n(v) = f(v)$. Fixing the input bits of C_n that belong to the formula φ we obtain a circuit C_φ with $C_\varphi(\alpha) = C_n(\langle \alpha, \varphi \rangle) = f(\langle \alpha, \varphi \rangle)$ and thus C_φ is of size polynomial in $|\varphi|$. Now observe that C_φ is an interpolant for the formulas $\varphi_0(x, y)$ and $\varphi_1(x, z)$. If $\varphi_0(\alpha, y)$ is satisfiable then $C_\varphi(\alpha) = 1$, and if $C_\varphi(\alpha) = 1$ then for no γ it holds $\neg\varphi_1(\alpha, \gamma)$ and therefore $\varphi_1(\alpha, z)$ is a tautology.

Finally the proof of the implication *3* $\Longrightarrow$ *2* is obtained using padding techniques. We omit the details due to the limited space. $\square$

It is easy to see that a proof system g admits an effective interpolation if there is a proof system h that simulates g, and h admits an effective interpolation. Hence, any proof system for TAUT admits an effective interpolation if there is an optimal proof system for TAUT that admits an effective interpolation. As a corollary we obtain

Corollary 8. *If there is an optimal proof system for TAUT that admits an effective interpolation then H2 holds.*

5 Relations between completeness assumptions

Using the characterization $\mathcal{NPSV}_t = \mathcal{FP}_t^{\mathcal{NP} \cap co\text{-}\mathcal{NP}}$ [23, 8] we obtain the following result

Theorem 9. *$\mathcal{NP} \cap co\text{-}\mathcal{NP}$ has a many-one complete set iff $\mathcal{NPSV}_t$ has a many-one complete function.*

Now let us consider the function class $\mathcal{NPSV}$. In the same way as $\mathcal{NPSV}_t$ corresponds to the language class $\mathcal{NP} \cap co\text{-}\mathcal{NP}$, the function class $\mathcal{NPSV}$ corresponds to the class of all disjoint $\mathcal{NP}$-pairs. In fact, if we denote the class of all 0,1-valued functions in $\mathcal{NPSV}$ by $\mathcal{NPSV}_{\{0,1\}}$ then any function $h \in \mathcal{NPSV}_{\{0,1\}}$ can be identified with the $\mathcal{NP}$-pair (A_0, A_1) where $A_b = \{x \in \Sigma^* \mid h(x) \mapsto b\}$.

Razborov [18] introduced a notion of many-one reducibility between disjoint $\mathcal{NP}$-pairs a stronger version of which was studied in [11]. Let A_0, A_1, B_0, B_1 be $\mathcal{NP}$-sets with $A_0 \cap A_1 = B_0 \cap B_1 = \emptyset$. The pair (A_0, A_1) (strongly) many-one reduces to (B_0, B_1) if there is a function $f \in \mathcal{FP}$ such that $f(A_b) \subseteq B_b$ ($f^{-1}(B_b) = A_b$, respectively) for $b \in \{0, 1\}$. Actually, it is easy to see that f is a many-one reduction between two functions in $\mathcal{NPSV}_{\{0,1\}}$ if and only if f is a strong many-one reduction between the corresponding disjoint $\mathcal{NP}$-pairs. Thus, the class of disjoint $\mathcal{NP}$-pairs has a strongly many-one complete pair if and only if $\mathcal{NPSV}_{\{0,1\}}$ has a many-one complete function. As shown in the next theorem, this is even equivalent to the assumption that $\mathcal{NPSV}$ has a many-one complete function.

Theorem 10. *The following statements are equivalent.*

1. *$\mathcal{NPSV}$ has a many-one complete function.*
2. *$\mathcal{NPSV}_{\{0,1\}}$ has a many-one complete function.*
3. *There is a strongly many-one complete disjoint $\mathcal{NP}$-pair.*

Proof. (Sketch). Implication *1* $\Longrightarrow$ *2* is easy to prove, and the equivalence of *2* and *3* is clear by the preceding discussion. To see that *2* implies *1* we observe that $\mathcal{NPSV}$ can be characterized as $\mathcal{FP}^{\mathcal{NPSV}_{\{0,1\}}}$ where the value $M^f(x)$ computed by the deterministic oracle transducer M on input x is only defined if all oracle queries belong to the domain of the functional oracle f. □

We conclude this section by observing that the class of disjoint $co\text{-}\mathcal{NP}$ pairs corresponds to the class $\mathcal{NP}b\mathcal{V}_t$ of all 0,1-valued functions in $\mathcal{NPMV}_t$ studied in [5] (with the disjoint $co\text{-}\mathcal{NP}$-pair (A_0, A_1) associate the function $h \in \mathcal{NP}b\mathcal{V}_t$ defined by set-$h(x) = \{b \mid x \notin A_b\}$). Similar to the implication $1 \Longrightarrow 3$ in Theorem 10 the following theorem can be proved.

Theorem 11. *If $\mathcal{NPMV}_t$ has a many-one complete function then there exists a strongly many-one complete disjoint $co\text{-}\mathcal{NP}$-pair.*

We leave it open whether the reverse implication also holds.

6 *Existence* of (p-)optimal proof systems

In Theorem 3 it is observed that *sat* is p-optimal iff every optimal proof system is p-optimal. Although the assumption of the mere existence of a p-optimal proof system for SAT is presumably weaker than the assumption that *sat* is p-optimal, it is still equivalent to a quite similar statement, namely that any set with an optimal proof system has a p-optimal proof system. For the proof of this result we use the following observation from [11].

Lemma 12 ([11]). *If L has a (p-)optimal proof system, and $T \leq_m^p L$ then T has a (p-)optimal proof system (respectively).*

Theorem 13. *The following statements are equivalent.*

1. *SAT has a p-optimal proof system.*
2. *Any language L that has an optimal proof system also has a p-optimal proof system.*

Proof. Clearly *2* $\Longrightarrow$ *1*, as SAT has an optimal proof system. To see the inverse implication assume that SAT has a p-optimal proof system. Let T_L (cf. [11, 15]) be the following language consisting of tuples $\langle M, x, 0^s \rangle$ where M is a deterministic Turing transducer, $s \geq 0$ and $x \in \Sigma^*$.

$$T_L = \{\langle M, x, 0^s \rangle \mid \text{if } \mathrm{time}_M(x) \leq s \text{ then } M(x) \in L\}.$$

Notice that T_L is many-one reducible to L (without restriction assume $L \neq \emptyset$). Hence, the assumption that there is an optimal proof system for L implies that T_L has an optimal proof system, say h. Let

$$S = \{\langle\langle M, x, 0^s \rangle, 0^l \rangle \mid \exists w,\ |w| \leq l,\ h(w) = \langle M, x, 0^s \rangle\}.$$

Clearly $S \in \mathcal{NP}$. Therefore by assumption there is a p-optimal proof system system g for S. Let now f be the following proof system.

$$f(w) = \begin{cases} y & \text{if } g(w) = \langle\langle M, x, 0^s \rangle, 0^l \rangle,\ \mathrm{time}_M(x) \leq s, \text{ and } M(x) = y\,, \\ \text{undef.} & \text{otherwise.} \end{cases}$$

First notice that $y \in L$ if $f(w) = y$. This is due to the fact that $g(w) = \langle\langle M, x, 0^s \rangle, 0^l \rangle$ implies $\langle\langle M, x, 0^s \rangle, 0^l \rangle \in S$ which in turn implies $\langle M, x, 0^s \rangle \in T_L$. We now show that f p-simulates every proof system f' for L. Assume that f' is computed by the transducer $M_{f'}$ in polynomial time $p(n)$. Observe that $\langle M_{f'}, x, 0^{p(|x|)} \rangle \in T_L$ for any $x \in \Sigma^*$. Hence, one may define a proof system for T_L such that for any x the tuple $\langle M_{f'}, x, 0^{p(|x|)} \rangle$ has the short proof $1x$. Consequently, due to the optimality of h, there is a polynomial q such that $\langle M_{f'}, x, 0^{p(|x|)} \rangle$ has an h-proof of size $\leq q(|x|)$. Hence $\langle\langle M_{f'}, x, 0^{p(|x|)} \rangle, 0^{q(|x|)} \rangle \in S$ for any x, and one may define a proof system g' for S with $g'(1x) = \langle\langle M_{f'}, x, 0^{p(|x|)} \rangle, 0^{q(|x|)} \rangle$ for any x. As g is p-optimal, g p-simulates g', i.e. there is a function $t \in \mathcal{FP}$ such that $g(t(1x)) = g'(1x) = \langle\langle M_{f'}, x, 0^{p(|x|)} \rangle, 0^{q(|x|)} \rangle$. Observe now that $f(t(1x)) = f'(x)$ for any x. Hence f p-simulates f'. $\square$

As shown in [11], the assumption that SAT and TAUT both have p-optimal proof systems implies that $\mathcal{NP} \cap co\text{-}\mathcal{NP}$ has a many-one complete set. In fact, due to Theorem 13 it suffices to assume that SAT has a p-optimal proof system and TAUT only has an optimal proof system. Together with Theorem 9 we obtain

Corollary 14. *If SAT has a p-optimal and TAUT has an optimal proof system then $\mathcal{NP} \cap$ co-$\mathcal{NP}$ has a many-one complete set, and $\mathcal{NPSV}_t$ has a many-one complete function.*

It is observed in [18] that the existence of an optimal proof system for TAUT implies the existence of a many-one complete pair for the class of disjoint $\mathcal{NP}$-pairs. In [11] it is shown that the same assumption allows one to infer the existence of a strongly many-one complete disjoint $\mathcal{NP}$-pair which by Theorem 10 implies that $\mathcal{NPSV}$ has a many-one complete function.

Corollary 15. *If TAUT has an optimal proof system then $\mathcal{NPSV}$ has a many-one complete function.*

Next we observe that a p-optimal proof system for SAT implies a complete function for the class $\mathcal{NPMV}_t$. The proof uses ideas from [11] (in fact, the only extension is that we are dealing with multi-valued functions here).

Theorem 16. *If SAT has a p-optimal proof system then $\mathcal{NPMV}_t$ has a many-one complete function.*

By Theorem 11 we obtain

Corollary 17. *If SAT has a p-optimal proof system then there exists a strongly many-one complete disjoint co-$\mathcal{NP}$-pair.*

7 Conclusion

We showed that the assumption that certain proof systems are (p-)optimal can be used to derive collapse results. Also we presented some relations between completeness assumptions for different classes. It would be interesting to know whether these observations could be extended to further proof systems and promise classes.

References

1. José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity I.* Springer-Verlag, 2nd edition, 1995.
2. Ronald V. Book, Timothy J. Long, and Alan L. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.
3. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.
4. Pierluigi Crescenzi, Viggo Kann, Riccardo Silvestri, and Luca Trevisan. Structure in approximation classes. *SIAM Journal on Computing*, 28(5):1759–1782, 1999.
5. Stephen A. Fenner, Lance Fortnow, Ashish V. Naik, and John D. Rogers. Inverting onto functions. In *Proceedings of the 11th Conference on Computational Complexity*, pages 213–222. IEEE, 1996.
6. Lance Fortnow and John D. Rogers. Separability and one-way functions. In *Proceedings of the 5th International Symposium on Algorithms and Computation, LNCS* #834, pages 396–404. Springer-Verlag, 1994.

7. Joachim Grollmann and Alan L. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.
8. Lane A. Hemaspaandra, Ashish V. Naik, Mitsunori Ogihara, and Alan L. Selman. Computing solutions uniquely collapses the polynomial hierarchy. In *Proceedings of the 5th International Symposium on Algorithms and Computation*, *LNCS* #834, pages 56–64. Springer-Verlag, 1994.
9. Russel Impagliazzo and Moni Naor. Decision trees and downward closures (extended abstract). In *Proceedings of the Third Conference on Structure in Complexity Theory*, pages 29–38. IEEE, 1988.
10. Ker-I Ko. On some natural complete operators. *Theoretical Computer Science*, 37:1–30, 1985.
11. Johannes Köbler and Jochen Messner. Complete problems for promise classes by optimal proof systems for test sets. In *Proceedings of the 13th Conference on Computational Complexity*, pages 132–140. IEEE, 1998.
12. Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, 1989.
13. Jan Krajíček and Pavel Pudlák. Some consequences of cryptographical conjectures for S_2^1 and EF. In D. Leivant, editor, *Logic and Computational Complexity*, *LNCS* #960, pages 210–220. Springer-Verlag, 1995.
14. Timothy J. Long. On γ-reducibility versus polynomial time many-one reducibility. *Theoretical Computer Science*, 14:91–101, 1981.
15. Jochen Messner. On optimal algorithms and optimal proof system. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science*, *LNCS* #1563. Springer-Verlag, 1999.
16. Jochen Messner and Jacobo Torán. Optimal proof systems for propositional logic and complete sets. In *Proceedings of the 15th Symposium on Theoretical Aspects of Computer Science*, *LNCS* #1373, pages 477–487. Springer-Verlag, 1998.
17. Christos H. Papadimitriou. *Computatational Complexity.* Addison-Wesley, 1994.
18. Alexander A. Razborov. On provably disjoint $\mathcal{NP}$-pairs. Technical Report RS-94-36, Basic Research in Computer Science Center, Aarhus, 1994.
19. Zenon Sadowski. On an optimal quantified propositional proof system and a complete language for $\mathcal{NP} \cap co\text{-}\mathcal{NP}$. In *Proceedings of the 11th International Symposium on Fundamentals of Computing Theory*, *LNCS* #1279, pages 423–428. Springer-Verlag, 1997.
20. Zenon Sadowski. On an optimal deterministic algorithm for SAT. In *Proceedings of the 12th Annual Conference of the European Association for Computer Science Logic, CSL '98*, *LNCS* #1584, pages 179–187. Springer-Verlag, 1999.
21. Uwe Schöning. *Complexity and Structure*, *LNCS* #211. Springer-Verlag, 1985.
22. Uwe Schöning and Jacobo Torán. A note on the size of craig interpolants, 1996. Unpublished manuscript.
23. Alan L. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*, 48(2):357–381, 1994.
24. Leslie G. Valiant. Relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.