

Optimal proof systems for Propositional Logic and complete sets

Jochen Meßner and Jacobo Torán
Universität Ulm
Theoretische Informatik
D-89069 Ulm, Germany
messner,toran@informatik.uni-ulm.de

June 18, 1997

Abstract

A polynomial time computable function $h : \Sigma^* \rightarrow \Sigma^*$ whose range is the set of tautologies in Propositional Logic (TAUT), is called a proof system. Cook and Reckhow defined this concept in [6] and in order to compare the relative strength of different proof systems, they considered the notion of p-simulation. Intuitively a proof system h p-simulates a second one h' if there is a polynomial time computable function γ translating proofs in h' into proofs in h . A proof system is called optimal if it p-simulates every other proof system. The question of whether p-optimal proof systems exist is an important one in the field. Krajíček and Pudlák [14, 13] have given a sufficient condition for the existence of such optimal systems, showing that if the deterministic and nondeterministic exponential time classes coincide, then p-optimal proof systems exist. They also give a condition implying the existence of optimal proof systems (a related concept to the one of p-optimal systems) exist. In this paper we improve this result giving a weaker sufficient condition for this fact. We show that if a particular class of sets with low information content in nondeterministic double exponential time is included in the corresponding deterministic class, then p-optimal proof systems exist. We also show some complexity theoretical consequences that follow from the assumption of the existence of p-optimal systems. We prove that if p-optimal systems exist the the class UP (an some other related complexity classes) have many-one complete languages, and that many-one complete sets for $NP \cap SPARSE$ follow from the existence of optimal proof systems.

1 Introduction

A systematic study of the complexity of proof systems for Propositional Logic, was started some time ago by Cook and Reckhow [6]. They were interested in studying the shortest proofs of propositional tautologies in different proof systems, and defined the abstract notion of proof system in the following way:

Definition 1.1 Let TAUT be the set of all Boolean tautologies (written in a fixed alphabet Σ). A *propositional proof system* (or just *proof system*) is a polynomial time computable function $h : \Sigma^* \rightarrow \Sigma^*$ whose range is TAUT¹.

For example the function h defined as

$$h(w) = \begin{cases} \varphi & \text{if } w = \langle \varphi, v \rangle \text{ and } v \text{ is a resolution proof of } \varphi, \\ x \vee \bar{x} & \text{otherwise.} \end{cases}$$

is a proof system.

If $h(w) = \varphi$ we say that w is a proof of φ in h . Observe that in the given definition a proof system h is not required to be polynomially honest. For a tautology φ , the shortest proof of φ in h , can be much longer than φ .

A *polynomially bounded proof system* h is a proof system in which every tautology has a short proof. More formally, there is a polynomial q such that for every $\varphi \in \text{TAUT}$, there is a string w of length bounded by $q(|\varphi|)$ with $h(w) = \varphi$.

It is not known whether polynomially bounded proof systems exist, on the other hand, many concrete proof systems have been shown not to be polynomially bounded (see for example [6],[16]). Besides the interest that concrete proof systems, like for example resolution or Frege systems have in their own, a main motivation for the study of proof systems comes in fact from the following relation between the NP versus *co*-NP question and the existence of polynomially bounded systems.

Theorem 1.2 [6] NP = *co*-NP if and only if polynomially bounded proof systems exist.

This result started the so called *Cook-Reckhow Program*: a way to prove that NP is different from *co*-NP might be to study more and more powerful concrete proof systems, showing that they are not polynomially bounded, until hopefully we have gained enough knowledge to be able to separate NP from *co*-NP (see [3]).

In order to compare the relative powers of two different proof systems, the notion of polynomial simulation (or p-simulation) was introduced in [6].

Definition 1.3 Let h and h' be two propositional proof system. We say that h simulates h' if there is a polynomially bounded function γ translating proofs in h' into proofs in h . In other words, for every tautology φ and every proof of φ in h' , $\gamma(w)$ is a proof of φ in h . If γ is computable in polynomial time, we say that h p-simulates h' .

Observe that p-simulation is a stronger notion than simulation. It is easy to see that simulation and p-simulation are reflexive and transitive relations. It is also clear that if a proof system h is not polynomially bounded, and h simulates another system h' , then h' cannot be polynomially bounded. Cook and Reckhow used p-simulation in order to classify proof systems in different classes with polynomially related derivation strength.

The notion of simulation between proofs systems is closely related to the notion of reducibility between problems. Continuing with this analogy, the notion of complete problem would correspond to the notion of optimal proof system.

¹The original definition allows in fact the use of different alphabets for the domain and range of h , but for the purposes of this paper the given definition suffices.

Definition 1.4 A proof system is optimal (p-optimal) if it simulates (p-simulates) every other proof system.

An important open problem is whether optimal proof systems exist [3]. Observe that if this were the case, then in order to separate NP from *co*-NP it would suffice to prove that a concrete proof system is not polynomially bounded.

Krajíček and Pudlák have given sufficient conditions for the existence of p-optimal and optimal proof systems.

Theorem 1.5 [14, 13]

If $NE = co\text{-}NE$ then optimal proof systems exist.

If $E = NE$ then p-optimal proof systems exist.

On the other hand, to our knowledge, no complexity-theoretic consequences of the existence of optimal systems were known.

In the present paper we improve the mentioned result from [13] by weakening the conditions that are sufficient for the existence of optimal and p-optimal proof systems. We show in Section 3 that if the deterministic and nondeterministic double exponential time complexity classes coincide ($EE = NEE$) then p-optimal proof systems exist, and that $NEE = co\text{-}NEE$ is sufficient for the existence of optimal proof systems. In fact we give a probably weaker sufficient condition, using a special kind of sets with small information content. Let us say that a set is *almost tally*, if its words belong to the set 0^*10^* . We show that if the class of almost tally sets in NEE is included in EE , then p-optimal proof systems exist, and that optimal proof systems exist if almost tally sets in NEE belong also to $co\text{-}NEE$. Observe that if this result could also be proved for tally sets, instead of almost tally sets, then a sufficient condition for the existence of p-optimal proof systems would be $EEE = NEEE$, where EEE denotes the complexity class $\text{DTIME}(2^{O(2^{2^n})})$ (see [7]).

We also show some consequences from the existence of optimal and p-optimal proof systems, proving completeness results for the complexity classes UP and $\text{NP} \cap \text{SPARSE}$, that follow from the existence of such proof systems. Since complete problems for these classes have been unsuccessfully searched for in the past, the results give some evidence of the fact that optimal proof systems might not exist. At the same time they strengthen the connection between the notions of optimal proof systems and complete sets.

In Section 4 we show that if p-optimal proof systems exist, then the class UP (unambiguous NP) of problems in NP that can be accepted by nondeterministic polynomial time machines with at most one accepting path for every input [17], has complete problems under the logarithmic space many-one reductions. The existence of complete problems for UP has been studied in [9], where the authors show the existence of a relativization under which this is not possible. Considering that p-optimal proof systems exist, we also show the existence of complete sets for related *promise* classes like FewP . We also consider the weaker hypothesis of the existence of optimal proof systems, and show that the completeness results for the mentioned classes still hold for the nonuniform many-one reductions.

Finally in Section 5 we prove that optimal proof systems imply the existence of complete problems for the class $\text{NP} \cap \text{SPARSE}$ for many-one logarithmic space reductions. The question of the existence of such sets is subtle and has been intensively investigated. Although

many-one complete set in $\text{NP} \cap \text{SPARSE}$ are not known, Hartmanis and Yesha prove in [10] that there is a sparse set in NP that is Turing complete for $\text{NP} \cap \text{SPARSE}$, (in fact the given set is tally) and ask whether the result can be improved to the many-one case. Hartmanis has also shown that the set of satisfiable formulas with small Kolmogorov complexity $\text{SAT} \cap K[\log, n^2]$ is Turing complete for $\text{NP} \cap \text{SPARSE}$ [8], but the completeness of this set under many-one reductions would imply unexpected consequences in the exponential time hierarchy. More recently Schöningh has proven that there are sets that are complete for this class under many-one randomized reductions [15].

2 Basic Notions

We assume some familiarity with the standard results and notions about deterministic and nondeterministic complexity classes. For undefined complexity theory notions, and the definition of standard complexity classes, we refer the reader to the standard books in the area like [2]. We will use a pairing function $\langle \cdot, \cdot \rangle$ that is polynomial time computable and invertible. For a set A , $\mathfrak{P}(A) = \{L \mid L \subseteq A\}$ represents the power set of A .

E and EE , denote the time complexity classes $\text{DTIME}(2^{O(n)})$ and $\text{DTIME}(2^{O(2^n)})$, respectively, and NE , NEE their nondeterministic counterparts. For a language $L \subseteq \{0,1\}^*$ let $\text{Tally}(L) = \{0^{n(x)} \mid x \in L\}$ denote the tally version of L , where $n(x)$ for each $x \in \{0,1\}^*$ denotes the number with binary representation $1x$. The languages in E (NE) correspond to the tally languages in P (resp. NP), see [4]. Similarly the languages in EE (NEE) correspond to the tally languages in E (resp. NE).

Lemma 2.1 *Let $L \subseteq \{0,1\}^*$, $L_{(1)} = \text{Tally}(L)$, $L_{(2)} = \text{Tally}(L_{(1)})$. The following statements are equivalent*

1. $L \in \text{EE}$ (resp. $L \in \text{NEE}$, $L \in \text{co-NEE}$)
2. $L_{(1)} \in \text{E}$ (resp. $L_{(1)} \in \text{NE}$, $L_{(1)} \in \text{co-NE}$)
3. $L_{(2)} \in \text{P}$ (resp. $L_{(2)} \in \text{NP}$, $L_{(2)} \in \text{co-NP}$)

We have defined in the introduction the notion of proof system. The next lemma shows that for every set of tautologies that is polynomial time computable, there is a proof system in which the tautologies of the set have short proofs, and moreover, the proof can be found easily.

Lemma 2.2 *If $T \subseteq \text{TAUT}$ and $T \in \text{P}$, then there exists a proof system h and a function $t \in \text{FP}$ that produces proofs in h for every tautology in T . That is, for every $\varphi \in T$, $h(t(\varphi)) = \varphi$.*

Proof: Let $h' \in \text{FP}$ be a proof system. We can define a new proof system h as follows:

$$h(w) = \begin{cases} h'(v) & \text{if } w = 0v, \\ v & \text{if } w = 1v \text{ and } v \in T, \\ x \vee \bar{x} & \text{otherwise.} \end{cases}$$

Clearly, h is a proof system. The function t producing proofs in h for the elements of T is just $t(v) = 1v$. ■

In the definitions of proof systems and p-optimality we allowed both functions, the proof system h and the translation function, to be computable in polynomial time. The following lemma shows that the most part of the computational complexity of both functions can be concentrated in one of them, whereas the other function may be computed, for example, in logarithmic space. To formulate the lemma, let us say that a proof system h is *logspace-optimal* if it simulates in logarithmic space every other proof system h' , which means that there is logspace computable function γ such that $h(\gamma(w)) = h'(w)$ for every word w .

Lemma 2.3 *The following statements are equivalent*

1. *A p-optimal polynomial time computable proof system exists.*
2. *A logspace-optimal polynomial time computable proof system exists.*
3. *A p-optimal logspace computable proof system exists.*

Proof: Clearly, 2 and 3 imply 1.

To obtain 2 from 1 let h be a p-optimal polynomial time computable proof system, and let g be defined by

$$g(w) = \begin{cases} h(w') & \text{if } w = \langle M, 0^l, v \rangle, \text{ and } M \text{ is a deterministic Turing transducer} \\ & \text{which on input } v \text{ outputs } w' \text{ in at most } l \text{ steps,} \\ x \vee \neg x & \text{else.} \end{cases}$$

Clearly, g is polynomial time computable. We show that g is logspace-optimal. Let h' be a proof system. By assumption, there is a polynomial time computable translation function γ such that $h(\gamma(w)) = h'(w)$ for any w . Let M be a deterministic Turing transducer computing γ with time bounded by a polynomial $p(n)$. It's easy to see that the logspace computable function γ' with $\gamma'(w) = \langle M, 0^{p(|w|)}, w \rangle$ translates proofs in h' into proofs in g .

We now show that the existence of h implies a logspace computable p-optimal proof system f . Let M be a polynomial time machine computing h . Let $f(w) = \varphi$ if w encodes a complete computation of M (given by the sequence of configurations) with output φ , let $f(w) = x \vee \neg x$ otherwise. Choosing a suitable encoding f is logspace computable. Also f p-simulates h . Therefore f p-simulates every proof system. ■

3 Sufficient conditions

We give now a sufficient condition for the existence of optimal proof systems, based on almost tally sets in double exponential time. We will later see that these conditions can be translated into analogous ones for tally sets with very low density in the polynomial time setting.

Theorem 3.1

*If $\text{co-NEE} \cap \mathfrak{P}(0^*10^*) \subseteq \text{EE}$ then there exists a p-optimal proof system.*

*If $\text{co-NEE} \cap \mathfrak{P}(0^*10^*) \subseteq \text{NEE}$ then there exists an optimal proof system.*

Proof: Let M_1, M_2, M_3, \dots be some standard enumeration of deterministic Turing transducers with binary input alphabet such that there is an universal Turing machine which is able to simulate k steps of M_i in $(ik)^2$ time for any $k \geq 0$ (clearly, such enumerations exist). Now define the language

$$T = \{0^j10^i \mid \text{for any word } w \text{ of length at most } 2^{2^n}, \text{ where } n = i + j + 1: \\ \text{if } M_i \text{ stops on input } w \text{ in at most } 2^{2^n} \text{ steps, } M_i \text{ outputs a tautology}\}.$$

We show that $T \in \text{co-NEE}$:

On input 0^j10^i let $n = j + i + 1$, and check for all $w \in \{0,1\}^*$ of length at most 2^{2^n} whether M_i stops in at most 2^{2^n} steps. If this is not the case, accept; else, accept if the output is a tautology and reject otherwise.

The simulation of 2^{2^n} steps of M_i can be performed in $i^2(2^{2^n})^2 \leq 2^{3 \cdot 2^n}$ steps. In $2^{2 \cdot 2^n}$ time it can be checked whether a formula of size at most 2^{2^n} is a tautology. For the guessing procedure one needs 2^{2^n} time. This is a total time of $2^{6 \cdot 2^n}$.

Assuming $\text{co-NEE} \cap \mathfrak{P}(0^*10^*) \subseteq \text{EE}$ we have $T \in \text{EE}$. Therefore there is a deterministic Turing machine M_T which decides T in time $2^{c \cdot 2^n}$ for some $c > 0$.

We describe a p-optimal proof system h :

On input $\langle 0^j10^i, 0^s, w \rangle$ examine if $s \geq 2^{2^l}$ and $|w| \leq 2^{2^l}$, where $l = i + j + 1$, and test whether M_T accepts 0^j10^i in at most s steps. If this is the case, output $M_i(w)$ if M_i stops after at most 2^{2^l} steps on input w . (If some other case applies, output some fixed tautology).

Clearly, $0^j10^i \in T$ implies that the Turing machine M_i on input w outputs a tautology if the computation needs at most 2^{2^l} steps. Therefore $h(\Sigma^*) \subseteq \text{TAUT}$. Also, h is computable in polynomial time. We have to show that h p-simulates every other proof system. Let g be a proof system computed by the deterministic Turing transducer M_i with time bound $n^k + k$. A proof w for g is translated into the proof $w' = \langle 0^j10^i, 0^s, w \rangle$ where $s = 2^{c \cdot 2^{i+j+1}}$, and $j = \max(0, \lceil \log \log |w|^k + k \rceil - i - 1)$. By the construction of h , we have $h(w') = g(w)$. We just have to show that the translation $w \mapsto w'$ is computable in polynomial time. Clearly, this is the case if the length of 0^s is polynomially bounded by $|w|$. Observe that $s = 2^{c \cdot 2^{i+j+1}} \leq 2^{c \cdot 2^{i+2} \cdot 2^{\log \log |w|^k + k}} = (|w|^k + k)^{c \cdot 2^{i+2}}$ which is clearly polynomial in $|w|$ (notice, i is fixed for g).

For the second part of the theorem, assume that $\text{co-NEE} \cap \mathfrak{P}(0^*10^*)$ is included in NEE which implies $T \in \text{NEE}$. Let N_T be a nondeterministic Turing machine accepting T with time bound $2^{c \cdot 2^n}$. An optimal proof system is given by the following algorithm:

On input $\langle 0^j10^i, 0^s, \alpha, w \rangle$ examine if $s \geq 2^{2^l}$ and $|w| \leq 2^{2^l}$, where $l = i + j + 1$, and test whether α encodes an accepting computation of N_T on input 0^j10^i . If this is the case, output $M_i(w)$ if M_i stops after at most 2^{2^l} steps on input w . (If some other case applies, output some fixed tautology).

A similar argument as the one above shows that this proof system simulates every other proof system. ■

Corollary 3.2

If $NEE \cap \mathfrak{P}(0^*10^*) \subseteq EE$ then there exists a p -optimal proof system.

If $NEE \cap \mathfrak{P}(0^*10^*) \subseteq co\text{-}NEE$ then there exists an optimal proof system.

Proof: The corollary follows from the observation that $co\text{-}NEE \cap \mathfrak{P}(0^*10^*) = \{0^*10^* \setminus L \mid L \in NEE \cap \mathfrak{P}(0^*10^*)\}$. ■

We translate now these results to the polynomial time setting. A tally language $L \subseteq \{0\}^*$ is called $f(n)$ -tally for some function $f : \mathbb{N} \rightarrow \mathbb{N}$ if $L \cap \{0\}^{\leq n} \leq f(n)$ for all n .

Theorem 3.3

P-optimal proof systems exist if every $(\log \log n)^2$ -tally language in NP is already in P.

Optimal proof systems exist if every $(\log \log n)^2$ -tally language in NP is in co-NP.

Proof: Let $L \subseteq 0^*10^*$ be an arbitrary language in NEE, and let $L_{(2)} = \text{Tally}(\text{Tally}(L))$. By Lemma 2.1 $L_{(2)} \in NP$. Observe $L_{(2)} \subseteq \{0^{2^{2^l+2^j}} \mid 0 \leq j < l\}$ and

$$\begin{aligned} \|\{0^{2^{2^l+2^j}} \mid 0 \leq j < l\} \cap \Sigma^{\leq n}\| &\leq \|\{(l, j) \mid 2^l + 2^j \leq \log n\}\| \\ &\leq (\log \log n)^2. \end{aligned}$$

Therefore $L_{(2)}$ is a $(\log \log n)^2$ -tally language in NP. By Lemma 2.1, $L_{(2)} \in P$ implies $L \in EE$. Similarly $L_{(2)} \in co\text{-}NP$ implies $L \in co\text{-}NEE$. ■

4 Complete problems for UP

In this and the next sections we prove that the existence of optimal proof systems imply the existence of complete sets in certain *promise* complexity classes. The machines computing the sets in these classes can not be guaranteed to keep the condition of the class for all possible inputs, and because of this, complete problems for these classes are not known. We will first prove the existence of complete sets for UP under many-one polynomial time reductions, considering the existence of p -optimal proof systems. Later we will strengthen this result to logarithmic space reductions.

Define the set CAT containing descriptions of machines that are categorical, i.e. have at most one accepting path for all inputs up to a given length.

$$\text{CAT} = \{\langle M, 0^l, 0^n \rangle \mid M \text{ is a nondeterministic Turing machine and for every input } x, \\ |x| \leq n, M(x) \text{ has at most one accepting path of length } \leq l\}.$$

Clearly, $\text{CAT} \in co\text{-}NP$ and since TAUT is $co\text{-}NP$ complete for polynomial time many-one length-increasing reductions, there is such a function $f \in FP$ reducing CAT to TAUT.

For every fixed nondeterministic Turing machine M and every fixed monotone polynomial q , the set

$$\text{CAT}_{M,q} = \{\langle M, 0^{q(n)}, 0^n \rangle \mid n \geq 1\} \cap \text{CAT},$$

is in P. Also the image of $\text{CAT}_{M,q}$ under f , is in P; in order to test whether a given formula ϕ belongs to $f(\text{CAT}_{M,q})$ it suffices to generate the words of $\text{CAT}_{M,q}$ up to a given length, and check whether the image of one of these words after applying f coincides with ϕ .

Let $\varphi_{M,l,n}$ denote the formula $f(\langle M, 0^l, 0^n \rangle)$. Clearly, if machine M is categorical and its running time is bounded by q , then for every n , $\varphi_{M,q(n),n}$ is a tautology.

The following lemma intuitively says that under the hypothesis of the existence of p-optimal proof systems, for every categorical machine M there is a polynomial time computable function producing proofs of the categoricity of M .

Lemma 4.1 *Let $h \in \text{FP}$ be a p-optimal proof system. For every categorical machine M with running time bounded by a polynomial q , there is a function $g_{M,q} \in \text{FP}$ such that for every $l \in \mathbb{N}$, $g_{M,q}(0^l)$ produces an output w and $h(w) = \varphi_{M,q(l),l}$.*

Proof: Let M be a categorical machine polynomially time bounded by a polynomial q . Every formula in the set $f(\text{CAT}_{M,q})$ is a tautology. As we have seen this set is in P, and by Lemma 2.2, there is a proof system h' and a function $t \in \text{FP}$ that produces short proofs in h' for the tautologies in $f(\text{CAT}_{M,q})$. Formally, for every tautology $\varphi_{M,q(l),l} \in f(\text{CAT}_{M,q})$, $h'(t(\varphi_{M,q(l),l})) = \varphi_{M,q(l),l}$.

Since h is p-optimal, it p-simulates h' . This means that there is a function $\gamma \in \text{FP}$, translating proofs in h' into proofs in h , and for every tautology $\varphi_{M,q(l),l} \in f(\text{CAT}_{M,q})$ we have $h(\gamma(t(\varphi_{M,q(l),l}))) = \varphi_{M,q(l),l}$.

The claimed function $g_{M,q}$ on input 0^l computes the formula $\varphi_{M,q(l),l} = f(\langle M, 0^{q(l)}, 0^l \rangle)$, and applies functions t and γ to it. Clearly $g_{M,q} \in \text{FP}$. ■

We can now prove that p-optimal proof systems imply the existence of complete sets for UP.

Theorem 4.2 *If p-optimal proof systems exist then there are sets that are complete for UP under polynomial time many-one reductions.*

Proof: Let h be a p-optimal proof system and consider the set

$$A = \{\langle M, 0^l, w, x \rangle \mid M \text{ is the description of a NDTM and } h(w) = \varphi_{M,l,|x|} \\ \text{and } M \text{ accepts } x \text{ in } l \text{ steps or less}\}.$$

Set A is clearly in UP since $h(w) = \varphi_{M,l,|x|}$ means that this formula is a tautology, and therefore for every input of length smaller or equal than $|x|$ (and in particular for x), M has at most one accepting path of length l .

For the hardness part, let B be a set in UP, accepted by a machine M in time bounded by a polynomial q . W.l.o.g. we can suppose that for every n , $q(n) \leq q(n+1)$, and that on any input x , every computation path of M halts after exactly $q(|x|)$ steps. Consider the

function $g_{M,q} \in \text{FP}$ whose existence was proved in the above lemma. The function $\lambda \in \text{FP}$ defined for every $x \in \Sigma^*$ as

$$\lambda(x) = \langle M, 0^{q(|x|)}, g_{M,q}(0^{|x|}), x \rangle$$

many-one reduces B to A since $h(g_{M,q}(0^{|x|})) = \varphi_{M,q(|x|),|x|}$. ■

To see that p-optimal proof systems also imply the existence of logspace many-one complete problem for UP, we show that the construction in the proof of Lemma 4.1 can be modified so that $g_{M,q}$ is logspace computable. In fact, by the same observations it can be seen that even weaker reductions are possible. Remember $g_{M,q}(0^l) = \gamma(t(f(\langle M, 0^{q(l)}, 0^l \rangle)))$ where f is a length-increasing reduction from CAT to TAUT, $t(x) = 1x$ is the function from the proof of Lemma 2.2, and γ is a function translating proofs in a proof system h' into proofs in h , whose existence is guaranteed by the p-optimality of system h . Clearly, $\langle M, 0^{q(l)}, 0^l \rangle$ can be computed in logarithmic space from 0^l , and the function t is logspace computable. Also f can be chosen to be a logspace computable (and length-increasing) many-one reduction from CAT to TAUT. By Lemma 2.3 we can assume h to be logspace-optimal, and therefore we can also assume γ to be logspace computable. As the composition of logspace computable functions is again logspace computable we obtain:

Theorem 4.3 *If p-optimal proof systems exist then there are sets that are complete for UP under logarithmic space many-one reductions.*

The completeness result for UP can be extended to the related complexity classes FewP and Few as stated in the next theorem. The class FewP was defined in [1] as a generalization of the class UP. The sets in FewP are accepted by nondeterministic polynomial time Turing machines in which the number of accepting paths is bounded by a polynomial on the input size, and the accepting criteria is defined as in standard Turing machines. In the definition of Few [5], the machines are also restricted to have at most a polynomial number of accepting paths on every input, but the accepting criteria is more flexible. More formally, for a nondeterministic Turing machine M , denote by $\text{acc}_M(x)$ the number of accepting paths of M on input x . A set $A \subseteq \Sigma^*$ belongs to the complexity class Few if there is a polynomial p and a polynomial time predicate Q , such that for every input x , $\text{acc}_M(x) \leq p(|x|)$, and $x \in A$ if and only if $Q(x, \text{acc}_M(x))$ holds.

Theorem 4.4 *If p-optimal proof systems exist then there are sets that are complete under polynomial logspace many-one reductions for the classes FewP and Few.*

Proof: The proof is very similar to the one for the complete problems for UP, and we just give a sketch of it. Supposing the existence of p-optimal proof systems, a complete set for FewP is the set of tuples $\langle M, 0^l, w, x \rangle$ so that M is the description of a nondeterministic Turing machine, M accepts input x in l steps, and w is a proof of the fact that for every string of length $\leq x$, M has at most l accepting paths.

For the case of Few, a complete problem is the set of tuples $\langle M, M_Q, 0^l, w, x \rangle$ were everything is as before, except that M_Q is a description of the deterministic machine computing predicate $Q(x, \text{acc}_M(x))$, (in time l). ■

Let us mention at this point that in [9] an oracle is constructed under which the class UP does not have many-one complete sets. In [11] this result is improved from many-one to Turing reducibility, and it is also shown that under certain relativizations the class FewP does not have Turing complete sets. Since the proofs in this paper relativize, we can state the following corollary:

Corollary 4.5 *There exists a relativization under which p -optimal proof systems do not exist.*

If we only consider the existence of optimal proof systems (instead of p -optimal systems), then we can prove a version of the above result for nonuniform reductions. Intuitively a set is polynomial time nonuniformly many-one reducible to a second one, if the reduction function is not necessarily in FP, as in the usual polynomial time reductions, but it is computed by a family of polynomial size circuits [12].

Definition 4.6 A function $f : \Sigma^* \rightarrow \Sigma^*$ is in the class FP/poly if there is a polynomial q and two functions $g : 0^* \rightarrow \Sigma^*$ and $h : \Sigma^* \rightarrow \Sigma^* \times \Sigma^*$, such that $h \in \text{FP}$, for every $n \in \mathbb{N}$, $|g(0^n)| \leq q(n)$ and for every $x \in \Sigma^*$, $f(x) = h(x, g(0^{|x|}))$.

A set A is non-uniformly many-one reducible to a set B in polynomial time, if A is many-one reducible to B via a function $f \in \text{FP/poly}$.

To prove the existence of complete sets for UP under FP/poly reductions we need the following weak version of Lemma 4.1:

Lemma 4.7 *Let $h \in \text{FP}$ be an optimal proof system. For every categorical machine M with running time bounded by a polynomial p , there is a polynomial q such that for every $l \in \mathbb{N}$, there is a string w of length at most $q(l)$ with $h(w) = \varphi_{M,q(l),l}$.*

Proof: Let M be a categorical machine time bounded by a polynomial p . As in the proof of Lemma 4.1, there is a proof system h' and a function $t \in \text{FP}$ that produces short proofs in h' for the tautologies in $f(\text{CAT}_{M,p})$. Since h is optimal, it can simulate h' . Therefore, there is a polynomial r such that for every tautology φ and every proof w' of φ in h' , there is a proof w of φ in h of size $|w| \leq r(|w'|)$. The claimed polynomial q is just the composition of r and the polynomial bounding the running time of t . ■

Theorem 4.8 *If optimal proof systems exist then there are sets that are complete for UP under nonuniform polynomial time many-one reductions.*

Proof: The complete set under FP/poly reductions is the same set A shown to be complete under uniform reductions in Theorem 4.2.

The hardness part is also the same, for a set B in UP, computed by machine M in time bounded by a polynomial p , the function λ reducing B to A is defined for every $x \in \Sigma^*$ as

$$\lambda(x) = \langle M, 0^{q(|x|)}, w_{|x|}, x \rangle,$$

where $w_{|x|}$ is for example the smallest string w satisfying $h(w) = \varphi_{M,q(|x|),|x|}$. Lemma 4.7 guaranties that $w_{|x|}$ is polynomially bounded in $|x|$, and this implies that $\lambda \in \text{FP/poly}$. ■

As expected, the many-one completeness results for Few and FewP from Theorem 4.4 become completeness results for nonuniform many-one reductions if only the existence of optimal proof systems is considered.

5 Complete sets for $\text{NP} \cap \text{SPARSE}$

We prove now that there are many-one complete sets for $\text{NP} \cap \text{SPARSE}$ under the hypothesis of the existence of optimal proof systems. The proof follows the same lines as the previous one for complete sets in UP, but in this case we do not need p-optimality, and the existence of optimal proof systems suffices.

Let us define the set SP containing descriptions of nondeterministic machines that do not accept too many strings up to a given length:

$$\text{SP} = \{ \langle M, 0^l, 0^n \rangle \mid M \text{ is a nondeterministic Turing machine and there are at most } l \text{ pairs } (x_i, y_i), |x_i| \leq n, |y_i| \leq l, \text{ such that } x_i \neq x_j \text{ for } i \neq j, \text{ and } y_i \text{ is an accepting path of } M \text{ on input } x_i \}.$$

It is not hard to see that $\text{SP} \in \text{co-NP}$, and therefore SP is polynomial-time many-one reducible to TAUT. Let $f \in \text{FP}$ be a length increasing function that reduces SP to TAUT.

Let M be a fixed nondeterministic Turing machine with running time bounded by a polynomial q , that for every length l accepts at most $q(l)$ words of length l . The set

$$\text{SP}_{M,q} = \{ \langle M, 0^{q(n)}, 0^n \rangle \mid n \geq 1 \} \cap \text{SP},$$

is in P, and the image of $\text{SP}_{M,q}$ under f , is also in P;

Let $\zeta_{M,l,n}$ denote the formula $f(\langle M, 0^l, 0^n \rangle)$. Clearly if machine M runs in time bounded by q and accepts a q -sparse set of inputs, then for every n , $\varphi_{M,q(n),n}$ is a tautology.

The following lemma is analogous to Lemma 4.7, and says that in an optimal proof system, a proof of the fact that a machine accepts a sparse language up to a given length, can be polynomially bounded. The proof follows the same lines as the one for Lemma 4.7 and it is omitted.

Lemma 5.1 *Let $h \in \text{FP}$ be a p -optimal proof system. For every nondeterministic Turing machine M with running time bounded by a polynomial q , and such that for every $n \in \mathbb{N}$, M accepts at most $q(n)$ words of length n , there is a polynomial r such that for every $l \in \mathbb{N}$, there is a string $w \in \Sigma^*$, with $|w| \leq r(l)$ and $h(w) = \zeta_{M,q(l),l}$.*

We can now prove that optimal proof systems imply the existence of complete sets for $\text{NP} \cap \text{SPARSE}$.

Theorem 5.2 *If optimal proof systems exist then there are sets that are complete for $\text{NP} \cap \text{SPARSE}$ under logarithmic space many-one reductions.*

Proof: Let h be an optimal proof system, and let S be the set

$$S = \{ \langle 0^M, 0^l, 0^j, x \rangle \mid M \text{ is the description of a NDTM and there is a string } w \text{ of length } |w| \leq j \text{ and } h(w) = \zeta_{M,l,|x|} \text{ and } M \text{ accepts } x \text{ in } l \text{ steps or less} \}.$$

S belongs clearly to NP. Also, the number of string x such that $\langle 0^M, 0^l, 0^j, x \rangle \in S$ is bounded by l , since $\langle 0^M, 0^l, 0^j, x \rangle \in S$ implies that $\zeta_{M,l,|x|}$ is a tautology. Therefore for every length n there are at most n words of this length in S . This proves that S is sparse.

In order to see that S is hard for the class, let S' be a set in $\text{NP} \cap \text{SPARSE}$, accepted by a nondeterministic Turing machine M with time bounded by a polynomial q , and with density also bounded by q . By Lemma 5.1 there is a polynomial r such that for every $l \in \mathbb{N}$, there is a string w with $|w| \leq r(l)$ and $h(w) = \zeta_{M,q(l),l}$. The reduction from S' to S is given by the function

$$\lambda(x) = \langle 0^M, 0^{q(|x|)}, 0^{r(|x|)}, x \rangle.$$

Observe that this function is computable in logarithmic space, one-to-one, length increasing and also invertible in logarithmic space. ■

Let us mention at this point that contrary to the UP case, there is no known relativization under which the class $\text{NP} \cap \text{SPARSE}$ does not have many-one complete sets. For this reason, and considering the existing results on sparse sets mentioned in the introduction, we feel that Theorem 5.2 only provides a weak consequence of the existence of optimal proof systems.

6 Conclusions and open problems

We have given sufficient conditions for the existence of optimal proof systems, as well as some complexity theoretic consequences that follow from such systems. Since the possible existence of optimal systems is an important open question, it would be interesting to improve these results so that the question can be completely characterized, for example, in terms of a collapse of complexity classes. It would also be interesting to know whether the existence of complete sets for other promise complexity classes, like $\text{NP} \cap \text{co-NP}$ or BPP, is also related to the existence of optimal proof systems for Propositional Logic.

References

- [1] E. Allender. Invertible functions. Ph.D. dissertation, Georgia Institute of Technology, 1985.
- [2] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1988.
- [3] S. Buss. Lectures on Proof Theory. Tech Report No. SOCS-96.1, McGill University, 1996. (<http://www.cs.mcgill.ca/~denis/TR.96.1.ps.gz>)
- [4] R. V. Book. Tally languages and complexity classes. *Information and Control* **26**, pp. 186–193, 1974.
- [5] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory* **23**, pp. 95–106, 1990.
- [6] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic* **44**, pp. 36–50, 1979.
- [7] J. Hartmanis, N. Immerman and V. Sewelson. Sparse sets in $\text{NP} - \text{P}$: EXPTIME versus NEXPTIME. *Information and Control* **65**, pp. 158–181, 1985.

- [8] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science (FOCS'83)*, pp. 439–445, 1983
- [9] J. Hartmanis and L. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theoretical Computer Science*, **58**, pp. 129–142, 1988.
- [10] J. Hartmanis and J. Yesha. Computaton times of NP sets of different densities. *Theoretical Computer Science*, **34**, pp. 17–32, 1984.
- [11] L. Hemaspaandra, S. Jain and N. Vereshchagin. Banishing robust Turing completeness. *Int. Journal of Foundations of Computer Science*, **4**, pp. 245–265, 1993.
- [12] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pp. 302–309, 1980.
- [13] J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic* **54**, pp. 1063–1079, 1989.
- [14] P. Pudlák. On the length of proofs of finitistic consistency statements in first order theories. Logic Colloquium'84 (J. B. Paris et al., editors), North-Holland, Amsterdam, pp. 165–196, 1986
- [15] U. Schöning. On random reductions from sparse sets to tally sets. *Information Processing Letters*, **46**, pp. 239–241, 1993.
- [16] A. Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic* **1**, pp. 425–467, 1995.
- [17] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, **5**, pp. 20–23, 1976.