On Optimal Algorithms and Optimal Proof Systems

Jochen Messner
Abteilung Theoretische Informatik
Universität Ulm, 89069 Ulm, Germany
messner@informatik.uni-ulm.de

June 18, 1999

Abstract

A deterministic algorithm O accepting a language L is called (polynomially) optimal if for any algorithm A accepting L there is a polynomial p such that $\text{time}_O(x) \leq p(|x| + \text{time}_A(x))$ for every $x \in L$. It is shown that an optimal acceptor for a language L exists if there is a p-optimal proof system for L. If L is a p-cylinder also the inverse implication holds. This result widely generalizes work from Krajíček and Pudlák who showed the result for $L = \text{TAUT}$. It is further shown how to construct an optimal acceptor for a p-cylinder L, given an acceptor for L that runs fast on every easy subset of L. Then we investigate the relationship of this notion of an 'optimal acceptor' to a more general notion of optimality. Here, instead of considering time-complexity on each individual string x, worst-case time-bounds are considered. It is observed that every set complete for exponential time under linearly length-bounded polynomial-time many-one reducibility has an acceptor with an optimal time-bound whereas on the other hand no set hard for exponential time under polynomial-time many-one reducibility has a p-optimal proof system. Finally we show how these results can be translated to nondeterministic algorithms and optimal proof systems.

1 Introduction

The major aim in the development of algorithms for hard sets is to decrease the runtime. A related line of research is to design heuristics that have a good performance on important instances or to identify efficiently decidable subsets (see, e.g., [5]). It seems to be ambitious to ask for an algorithm that has in some sense the fastest possible runtime on every input, an algorithm that runs fast on easy instances, and in some sense includes all possible heuristics for the problem, even those that are not known yet. However, Levin [9] proved that such an optimal algorithm exists for the functional task to find

witnesses for elements of a given set in $\mathcal{NP}$ (cf., Theorem 1). For example, using the random access machine (RAM) model of computation, one can construct an algorithm O that finds satisfying assignments for formulas $\varphi \in$ SAT such that for any other algorithm A solving the same task, there is a constant c with $\text{time}_O(\varphi) \leq c(\text{time}_A(\varphi) + |\varphi|)$ for every $\varphi \in$ SAT (O may not halt on other inputs). One can rephrase Levin's result in terms of inverting polynomial time computable functions as follows (again we state the result using the RAM model).

Theorem 1 *For each (partial) function h computable by a RAM in polynomial time p there is a RAM M inverting h such that for every RAM M' inverting h there is a constant $c > 0$ with* $\text{time}_M(y) \leq c \cdot (\text{time}_{M'}(y) + p(|M'(y)|))$ *for every y in the range of h.*

Note that the brief note [9] contains no proof. Proofs of Theorem 1 can be found in [14, 6] (where the latter article uses the Kolmogorov-Uspensky machine model also used in [9]). It is noted in [14] that one can transfer the result to the Turing machine model if one replaces the term $c \cdot (\dots)$ in the above theorem by $c' \cdot ((\dots) \cdot \log(\dots))$.

To study the existence of optimal algorithms in a more machine independent fashion it is suitable to use the following definition of optimality. Intuitively, for some task to solve, let us call an algorithm O *optimal for this task on instances from $S \subseteq \Sigma^*$* if for any other algorithm A solving the same task there is a polynomial p such that $\text{time}_O(x) \leq p(\text{time}_A(x) + |x|)$ for every $x \in S$. Using the Turing machine model of computation (as we will do in the rest of the paper) one can formalize this intuition as follows.

Definition 1 *Let $\mathcal{A}$ be a collection of Turing machines, let $S \subseteq \Sigma^*$. A Turing machine $M \in \mathcal{A}$ is called* polynomially time-optimal *(short:* optimal*)* for $\mathcal{A}$ on S *if for any $M' \in \mathcal{A}$ there is a polynomial p such that $time_M(x) \leq p(time_{M'}(x) + |x|)$ for each $x \in S$.*

Intuitively in the definition a task is identified with the (usually non-recursive) collection of Turing machines performing this task. The definition also implies that any task that can be solved in polynomial time has an optimal algorithm.

Due to the efficient simulation of RAMs by Turing machines (and vice versa) Theorem 1 immediately implies

Corollary 2 *For each (partial) function $f \in \mathcal{FP}$ there is a Turing transducer that is optimal on the range of f for the deterministic transducers inverting f.*

Contrasting with this functional task, in this paper we primarily investigate the existence of an algorithm that is optimal on L for the deterministic

algorithms accepting the language L. For short, such an algorithm is called an *optimal acceptor for* L. We show that the existence of an optimal acceptor for L is closely related to the existence of a p-optimal proof system for L. In [4] Cook and Reckhow considered a function $h \in \mathcal{FP}$ with range L as an *(abstract) proof system* for L. A proof system h for L is called *p-optimal* if every proof system f for L is p-simulated by h which means that there is a function $g \in \mathcal{FP}$ such that $h(g(x)) = f(x)$ for any x in the domain of f. Connections between the existence of p-optimal proof systems and other complexity theoretical notions have also been studied in [8, 13, 11, 7, 3, 12]. Krajíček and Pudlák showed in [8] that an optimal acceptor for TAUT exists if, and only if, there is a p-optimal proof system for TAUT. Recently, using an idea from [7], Sadowski [12] showed that the result holds also for SAT instead of TAUT. A main objective of this paper is to generalize the result to further languages L. We prove that for any language L, an optimal acceptor for L exists if there is a p-optimal proof system for L. The reverse implication is proved under the assumption that L is a p-cylinder.

Schnorr shows in [14] that for self-reducible sets L, the complexity of the functional problem which is the problem to find witnesses for membership in L is closely related to the complexity of the decision problem. Using the result of Levin he is able to construct 'optimal' acceptors for self-reducible problems like, e.g., SAT. However, the notion of optimality used here is a more general one than the notion of optimality considered above. Instead of considering the time-complexity of the algorithms on each individual string, worst-case time-bounds are considered. A function $t : \mathbb{N} \to \mathbb{N}$ is called a *time-bound for an algorithm* A *on* $S \subseteq \Sigma^*$ if $\text{time}_A(x) \leq t(n)$ for every $x \in S$ with $|x| \leq n$. Let us call t *a time-bound for the set* L if t is a time-bound on L for a deterministic Turing machine accepting L. If, in addition, for any time bound s for L there is a polynomial p such that $t(n) \leq p(s(n))$, t is called an *optimal time-bound for* L. Rephrasing the cited result, one can construct an acceptor for SAT with an optimal time-bound. Note however that such an acceptor may have exponential run-time on every instance from SAT even on those instances that can be solved efficiently by some known algorithm. On the other hand, under the assumption $\mathcal{P} = \mathcal{NP}$ this algorithm has a polynomial time-bound on instances from SAT (the algorithm may not halt on other inputs).

The relation between both notions of optimality is examined in Section 4. There we show that any deterministic time class $\text{DTIME}(t(n))$ determined by a time-constructible function t with $\mathcal{P} \subseteq \text{DTIME}(t(n))$ contains a set that has no optimal acceptor but an optimal time-bound. This implies for example that no set $\leq_m^p$-hard for exponential time has a p-optimal proof system. On the other hand it is shown that any set complete for exponential

time under linearly length-bounded many-one reducibility has an optimal time-bound. In Section 3 we prove the main theorem. We also show a relationship between the performance of acceptors or proof systems on easy instances from L, and the existence of optimal acceptors or p-optimal proof systems for L. Finally in Section 5 we briefly discuss how the results can be transferred to nondeterministic algorithms and optimal proof systems.

An excerpt of a preliminary version of this paper appears as [10]. In the meanwhile we were able to simplify the proof of the main theorem.

2 Preliminaries

We assume some familiarity with standard notions of computational complexity theory, and refer the reader to books like [2] for notions not defined in this paper. Let Σ be some fixed finite alphabet containing 0 and 1. The output of a Turing transducer M on input $x \in \Sigma^*$ is denoted by $M(x)$; we write $M(x) = \perp$ if M does not accept or runs forever on input x. Similarly for a partial function $f : \Sigma^* \to \Sigma^*$ we write $f(x) = \perp$ if f is undefined on x; a transducer M *computes* f if $f(x) = M(x)$ for every $x \in \Sigma^*$. Let $\mathcal{FP}$ denote the class of partial functions computed by transducers that have a polynomial time bound on Σ^*. Let h be a function with range $R \subseteq \Sigma^*$; For a function f (and also for a transducer M computing f) we say that f *(resp. M) inverts h on $S \subseteq R$* if $h(f(y)) = y$ for $y \in S$. If additionally $f \in \mathcal{FP}$ we say that h *is $\mathcal{FP}$-invertible on S*. Following [2] a function h is called *1-invertible* (in polynomial time) if there is a function, denoted h^{-1}, in $\mathcal{FP}$ such that $h^{-1}(h(x)) = x$ for $x \in \Sigma^*$; h is called *length-increasing* if $|h(x)| > |x|$ for any x; we call h *linearly length-bounded* if $|h(x)| \leq c \cdot |x|$ for some constant c and any x. Given a subset S of the domain of f, $f(S)$ denotes the set $\{f(x) \mid x \in S\}$. A set $A \subseteq \Sigma^*$ *many-one reduces to* B via a total function $f \in \mathcal{FP}$ (in symbols: $A \leq_m^p B$) if for all $x \in \Sigma^*$, $x \in A$ if, and only if, $f(x) \in B$. If, additionally, f is 1-invertible with range Σ^*, which means $B \leq_m^p A$ via f^{-1}, A and B are called *p-isomorphic*. A set A that is p-isomorphic to $A \times \Sigma^*$ is called *p-cylinder*. The following Lemma shows *the* property of p-cylinders that makes the notion useful for this paper.

Lemma 3 *L is a p-cylinder if, and only if, any set that is $\leq_m^p$-reducible to L, is $\leq_m^p$-reducible to A via a 1-invertible, length increasing function.*

A proof of the lemma is found in [2]. It is further shown there that L is a p-cylinder if, and only if, L is 1-invertible paddable (L is called *1-invertible paddable* if there is a 1-invertible function $g \in \mathcal{FP}$ such that $g(\langle x, y\rangle) \in L$ if, and only if, $x \in L$ for all $x, y \in \Sigma^*$).

Let $\mathcal{E}$ denote the class DTIME($2^{O(n)}$) and let $\mathcal{NE}$ be its nondeterministic counterpart.

3 Optimal acceptors and p-optimal proof systems

We now prove the main theorem. It generalizes the result of Krajíček and Pudlák [8] for TAUT to any p-cylinder. Besides the main equivalence between p-optimal proof system and optimal acceptors it also contains results that relate the performance of acceptors or proof systems on easy instances from L to the existence of optimal acceptors or p-optimal proof systems.

Theorem 4 *For a p-cylinder L, the following statements are equivalent.*

1. *There is a p-optimal proof system for L.*
2. *There is a proof system for L that is $\mathcal{FP}$-invertible on any $S \in \mathcal{P}$ with $S \subseteq L$.*
3. *There is an optimal acceptor for L.*
4. *There is an acceptor for L that has a polynomial time-bound on every $S \in \mathcal{P}$ with $S \subseteq L$.*

In addition, the implications 1 $\Rightarrow$ 2, 3 $\Rightarrow$ 4, 2 $\Leftrightarrow$ 4, and 1 $\Rightarrow$ 3 hold for any language L.

A proof of the theorem is obtained by several constructions. More detailed, the implications *1* $\Rightarrow$ *2*, *3* $\Rightarrow$ *4*, *2* $\Leftrightarrow$ *4*, *1* $\Rightarrow$ *3*, and *2* $\Rightarrow$ *1* are proved by the Propositions 6, 5, 9, 10, and 13, respectively.

The implication *3* $\Rightarrow$ *4* has the most simple proof. Speaking informally Proposition 5 states that an optimal acceptor runs fast on easy instances from L

Proposition 5 *Let M be an optimal deterministic acceptor for L, and let S be a subset of L with $S \in \mathcal{P}$. Then there is a polynomial p such that $time_M(x) \leq p(|x|)$ for $x \in S$.*

Proof. One can combine a polynomial-time machine accepting S with M to obtain an acceptor for L that has a polynomial time-bound on S. Due to optimality, M also has a polynomial time-bound on S. ∎

The dual result for proof systems is given by Proposition 6. Speaking informally the proposition states that in a p-optimal proof system proofs for easy instances are easy to compute.

Proposition 6 *A p-optimal proof system for L is $\mathcal{FP}$-invertible on any subset S of L with $S \in \mathcal{P}$.*

Proof. Let h be a p-optimal proof system for L and let $S \subseteq L$, $S \in \mathcal{P}$. Let g be a proof system defined as follows

$$g(x) = \begin{cases} h(z) & \text{if } x = 0z, \\ z & \text{if } x = 1z \text{ and } z \in S, \\ \bot & \text{otherwise}. \end{cases}$$

As h is p-optimal there is a function $f \in \mathcal{FP}$ such that $h(f(x)) = g(x)$ for any $x \in \Sigma^*$. Now, let r be a polynomial time computable function with $r(y) = f(1y)$ for $y \in S$. One easily checks that $h(r(y)) = h(f(1y)) = g(1y) = y$ for any $y \in S$. ∎

We now prove the implications *2* $\Leftrightarrow$ *4*, and *1* $\Rightarrow$ *3* of the main theorem. To do so we use the following constructions. For a function $h \in \mathcal{FP}$ let I_h denote an optimal transducer inverting h with $I_h(y) \neq \bot$ iff y is in the range of h (I_h is given by Corollary 2). Let A_h denote the acceptor version of the transducer I_h (i.e., A_h accepts y if I_h stops on y). Finally for a Turing machine M let h_M denote a proof system for $L(M)$ with $h_M(\langle y, 0^s \rangle) = y$ if M accepts y in not more than s steps.

The proof of the following lemma is obvious.

Lemma 7 *If M has a polynomial time-bound on $S \subseteq L(M)$ then h_M is $\mathcal{FP}$-invertible on S.*

Lemma 8 follows from the optimality of I_h.

Lemma 8 *If h is $\mathcal{FP}$-invertible on S then A_h has a polynomial time-bound on S.*

Combining both lemmata we obtain

Proposition 9 *There is a proof system for L that is $\mathcal{FP}$-invertible on any $S \in \mathcal{P}$ with $S \subseteq L$ if, and only if, there is an acceptor for L that has a polynomial time-bound on every $S \in \mathcal{P}$ with $S \subseteq L$.*

We now show that A_h is an optimal acceptor for L when h is a p-optimal proof system for L.

Proposition 10 *If L has a p-optimal proof system then an optimal acceptor for L exists.*

Proof. Let h be a p-optimal proof system for L. Let M be an acceptor for L. We show that A_h is at most polynomially slower than M on inputs from L. As h is p-optimal there is a function $g \in \mathcal{FP}$ such that $h_M(x) = h(g(x))$ for any x in the domain of h_M. Let f denote a function mapping any $y \in L$ to $\langle y, 0^s \rangle$ with $s = \text{time}_M(y)$. By definition of h_M, $h_M(f(y)) = y$ for $y \in L$,

and therefore $h(g(f(y)) = y$ for $y \in L$, i.e. $g \circ f$ inverts h. Notice that the function $g \circ f(y)$ is computable in time polynomial in $|y| + \text{time}_M(y)$. As I_h is optimal for the transducers inverting f there is a polynomial p such that $\text{time}_{A_h}(y) = \text{time}_{I_h}(y) \leq p(|y| + \text{time}_M(y))$ for $y \in L$. ∎

We will now see that an inverse version of Proposition 6 holds. In the proof of Proposition 13 we show how to construct a p-optimal proof system for a p-cylinder L given a proof system for L that is $\mathcal{FP}$-invertible on every easy subset of L. We need some definitions first. For any set L let $T(L)$ be defined by

$$T(L) = \{\langle M, x, 0^s\rangle \mid x \in \Sigma^*, s \geq 0,\ M \text{ is a det. Turing transducer and if } \text{time}_M(x) \leq s \text{ then } M(x) \in L\}.$$

Fixing some transducer M let

$$T(L)_M = \{\langle M, x, 0^s\rangle \in T(L) \mid x \in \Sigma^*, s \geq 0\}.$$

Notice, if L is the range of the function computed by the transducer M then $T(L)_M$ is the trivial language $\{\langle M, x, 0^s\rangle \mid x \in \Sigma^*, s \geq 0\}$. In [7] it had been observed that $T(L)$ is many-one equivalent to L for $L \neq \Sigma^*$. We state this observation in Lemma 11. It shows that in some sense $T(L)$ is the hardest set that is $\leq^p_m$-reducible to L, any set $\leq^p_m$-reducible to L is reducible to $T(L)$ via a very simple many-one reduction.

Lemma 11 *For any set $L \subseteq \Sigma^*$ the following holds*

- *$T(L) \leq^p_m L$ if $\Sigma^* \neq L \neq \emptyset$.*
- *$A \leq^p_m L$ via f implies that A is reducible to $T(L)$ via $g : x \mapsto \langle M, x, 0^{p(|x|)}\rangle$ where M is a transducer computing f in polynomial time p.*

It is interesting to note that the Lemmas 3 and 11 imply that $T(L)$ is a p-cylinder.

Before proving Proposition 13. Let us first state the following lemma that holds for any language L.

Lemma 12 *Let g be a proof system for $T(L)$ such that for any transducer M computing a proof system for L, g is $\mathcal{FP}$-invertible on $T(L)_M$. Then there is a p-optimal proof system for L.*

Proof. We will observe that the following algorithm computes a p-optimal proof system h.

input $\langle M_1, M_2, x, 0^s, 0^n\rangle$
if $\text{time}_{M_1}(\langle M_2, x, 0^s\rangle) \leq n$ **then**
  let w be the output of M_1 on input $\langle M_2, x, 0^s\rangle$

if $g(w) = \langle M_2, x, 0^s \rangle$ and $\text{time}_{M_2}(x) \leq s$ **then**
output $M_2(x)$ and halt;
otherwise reject.

Notice that $g(w) = \langle M_2, x, 0^s \rangle$ implies $\langle M_2, x, 0^s \rangle \in T(L)$. Then, $M_2(x) \in L$ if $\text{time}_{M_2}(x) \leq s$. This shows that the range of h is a subset of L. We now show that h p-simulates any proof system f for L. Let M_2 be a transducer computing f in polynomial time p. By assumption there is a function $r \in \mathcal{FP}$ such that $g(r(\langle M_2, x, 0^s \rangle)) = \langle M_2, x, 0^s \rangle$ for all x and s. Let M_1 be a transducer computing r in polynomial time q. Now f is p-simulated by h via the translation $x \mapsto \langle M_1, M_2, x, 0^s, 0^n \rangle$ where $s = p(|x|)$ and $n = q(|\langle M_2, x, 0^s \rangle|)$. ∎

When L is a p-cylinder one can replace the set $T(L)$ in lemma above by L itself which yields the Proposition 13.

Proposition 13 *Let L be a p-cylinder, and let h be a proof system for L that is $\mathcal{FP}$-invertible on any $S \in \mathcal{P}$ with $S \subseteq L$. Then there is a p-optimal proof system for L.*

Proof. Let f be a 1-invertible reduction from $T(L)$ to L. We will show that a function $g \in \mathcal{FP}$ with $g(\langle z, x \rangle) = z$ if $f(z) = h(x)$ is a proof system for $T(L)$ that fulfills the conditions of Lemma 12. If a transducer M computes a proof system for L we have $T(L)_M \in \mathcal{P}$. Then $S = f(T(L)_M)$ is also in $\mathcal{P}$ as f is 1-invertible. By assumption h is $\mathcal{FP}$-invertible on S which means that there is a function $r \in \mathcal{FP}$ such that $h(r(y)) = y$ for $y \in S$. This implies $h(r(f(z))) = f(z)$ for $z \in T(L)_M$ and therefore $g(\langle z, r(f(z)) \rangle) = z$ for $z \in T(L)_M$. This means that g is $\mathcal{FP}$-invertible on $T(L)_M$. ∎

Notice that the implication in the Proposition 13 (and therefore also the other implications in Theorem 4) can be generalized to further languages L. For the proof of Proposition 13 to go through it suffices that there is a $\leq_m^p$-reduction from $T(L)$ to L such that $f(T(L)_M) \in \mathcal{P}$ for every machine M computing a proof system for L. On the other hand the author does not believe that any recursively enumerable language that has a maximal subset in $\mathcal{P}$, has an optimal acceptor. This would show that the implication *4* $\Rightarrow$ *3* of Theorem 4 does not hold for every non-p-levelable set (see [1] for definitions). It may be still possible though to prove the implication *3* $\Rightarrow$ *1* of Theorem 4 unconditionally using some other method.

4 Sets without an optimal acceptor

In this section we show that one encounters sets that have no optimal acceptor immediately when one leaves $\mathcal{P}$ in the deterministic world. We construct a set that has an optimal time-bound but has no optimal acceptor. The result implies that no p-cylinder $\leq_m^p$-hard for (e.g.) exponential time has an optimal acceptor nor a p-optimal proof system. On the other hand we show that any set complete for $\mathcal{E}$ under linearly length-bounded many-one reducibility has an optimal time bound.

A function $t : \mathbb{N} \to \mathbb{N}$ is called *time-constructible* if there is Turing machine that on input 0^n stops after $t(n)$ steps.

Theorem 14 *Let $t : \mathbb{N} \to \mathbb{N}$ be a time-constructible function such that for every polynomial p there is a number n with $p(n) \leq t(n)$. Then there is a language $L \in DTIME(t(n))$ for which the following holds*

- *there is no optimal acceptor for L.*
- *$t(n)$ is an optimal time bound for L.*

Proof. Let $M_1, M_2, \ldots$ be a standard enumeration of deterministic Turing acceptors, and let U be a universal machine that on input of 0^i1x simulates s steps of M_i on input 0^i1x in time $\leq c_i \cdot s^2 + c_i$. For any $i > 0$ let L_i be the regular language described by the expression 0^i10^*. Define

$$L_i' = \{x \in L_i \mid U \text{ does not accept } x \text{ in less than } t(|x|) \text{ steps}\},$$

and let $L = \bigcup_{i>0} L_i'$. Clearly $L \in \mathrm{DTIME}(t(n))$.

We first observe that the construction guarantees that for any machine M_i accepting L it holds $L_i' = L_i$ and $c_i \mathrm{time}_{M_i}(x)^2 + c_i \geq t(|x|)$ for $x \in L_i$. Assume that there is an $x = 0^i1z \in L_i$ with $x \notin L_i'$. As M_i accepts L, $0^i1z \notin L(M_i)$ and therefore U does not accept $0^i1z = x$. However this implies $x \in L_i'$ which contradicts the assumption. Further notice that $x \in L$ implies $\mathrm{time}_U(x) \geq t(|x|)$.

Now, we obtain in a straightforward way that L has the stated properties. Let M_i be an arbitrary machine accepting L. Notice that $L_i \cap \Sigma^n \neq \emptyset$ for $n \geq i{+}1$. Therefore for any time-bound s of M_i we have $c_i{\cdot}s(n)^2{+}c_i \geq t(n)$ for $n \geq i{+}1$. This shows that t is an optimal time-bound for L. Assume now, for contradiction, that M_i is an optimal acceptor for L. Clearly, one can combine any algorithm for L with a finite automaton accepting $L_i = L_i'$ to obtain an acceptor M' for L with $\mathrm{time}_{M'}(x) \leq |x| + 1$ for $x \in L_i$. Assuming that M_i is optimal on L there is a polynomial q such that $\mathrm{time}_{M_i}(x) \leq q(|x|)$ for $x \in L_i$. This implies $t(n) \leq c_i q(n)^2 + c_i$ for all $n \geq i + 1$. Let $d_i = \max\{t(n) \mid n \leq i\}$, and $p(n) = c_i q(n)^2 + c_i + d_i$. One obtains $t(n) \leq p(n)$ for every n which

contradicts the assumption that for some n the polynomial $p(n)+1$ is at most $t(n)$. ■

Observe that any time-constructible function t with $\mathcal{P} \subseteq \mathrm{DTIME}(t(n))$ fulfills the condition of Theorem 14.

In [7] it was proved that if a set L has a p-optimal proof system then any set $\leq_m^p$-reducible to L has a p-optimal proof system, too. Together with Theorem 4 one obtains the following two corollaries.

Corollary 15 *No set $\leq_m^p$-hard for $\mathcal{E}$ has a p-optimal proof system.*

Corollary 16 *No p-cylinder $\leq_m^p$-hard for $\mathcal{E}$ has an optimal acceptor.*

Notice that the proof of Theorem 14 does not rely on the monotonicity of time-bounds. Due to the definition of the notion 'time-bound' in this article, a time-bound r' for a machine M on L can be replaced by the monotone function r defined by $r(n) = \min\{r'(m) \mid m \geq n\}$. Notice, r is a time-bound for M on L with $r(n) \leq r'(n)$ for each n. Using this monotonicity one obtains the following theorem. Let us call a set $L \in \mathcal{E}$ *complete for $\mathcal{E}$ under linearly length-bounded many-one reducibility* if for any set $B \in \mathcal{E}$ there is a linearly length-bounded function $f \in \mathcal{FP}$ that many-one reduces B to L.

Theorem 17 *Every set complete for $\mathcal{E}$ under linearly length-bounded many-one reducibility has an optimal time-bound.*

Proof. Let L be the set constructed in the proof of Theorem 14 with $t(n) = 2^n$. Let now $C \in \mathrm{DTIME}(2^{cn})$, let f be a many-one reduction of L to C with $|f(x)| \leq d|x|$ for any x, and let M_f be a transducer computing f in polynomial time $q(n)$. Combining M_f with a machine M_C accepting C one obtains a machine M_i accepting L with $\mathrm{time}_{M_i}(x) = \mathrm{time}_{M_f}(x) + \mathrm{time}_{M_C}(f(x))$ for any x. Therefore $s(n) = q(n) + r(dn)$ is a time-bound for M_i where r is a monotone time-bound for M_C. Remember that the construction of L implies that for any time-bound $s(n)$ for M_i on L we have $t(n) \leq c_i s(n)^2 + c_i$ for $n \geq i+1$. This implies $t(n) \leq p(r(dn))$ for some monotone polynomial p and any n. Due to the monotonicity of $p \circ r$ this implies $t(\lfloor n/d \rfloor) \leq p(r(d \cdot \lfloor n/d \rfloor)) \leq p(r(n))$. Now, $2^{cn} \leq 2^{cd(\lfloor n/d \rfloor + 1)} = 2^{cd} \cdot t(\lfloor n/d \rfloor)^{cd} \leq 2^{cd} p(r(n))^{cd}$. This proves that 2^{cn} is an optimal time-bound for C. ■

5 Optimal nondeterministic acceptors and optimal proof systems

In this section we briefly sketch how the results in the previous sections can be translated to the nondeterministic case. Clearly the notion of an optimal acceptor has a straightforward nondeterministic correspondence. A nondeterministic acceptor N for L is called optimal if for any nondeterministic acceptor N' for L there is a polynomial p such that $\text{time}_N(x) \leq p(|x| + \text{time}_{N'}(x))$ for $x \in L$, where $\text{time}_N(x)$ denotes the number of steps in the shortest accepting path of N on input x. A proof system h for L is called *optimal* if every proof system f for L is simulated by h which means that there is a polynomial p such that for any x in the domain of f there is a z, $|z| \leq p(|x|)$, with $h(z) = f(x)$. Basically, a proof system h can be associated with the nondeterministic acceptor N that on input y guesses x and accepts if $h(x) = y$. Symmetrically a nondeterministic acceptor N can be transformed to a proof system h with $h(\langle x, \alpha \rangle) = x$ if α denotes an accepting path of N on input x. Therefore it is a simple observation that there is an optimal nondeterministic acceptor for L if, and only if, there is an optimal proof system for L. Nonetheless, the result corresponding to the remaining equivalence of Theorem 4 is of some interest. Again this generalizes a result from [8] for TAUT.

Theorem 18 *For a p-cylinder L, the following statements are equivalent.*

1. *There is an optimal proof system for L.*
2. *There is an optimal nondeterministic acceptor for L.*
3. *There is a nondeterministic acceptor for L that has a polynomial time-bound on every $S \in \mathcal{NP}$ with $S \subseteq L$.*
4. *There is a nondeterministic acceptor for L that has a polynomial time-bound on every $S \in \mathcal{P}$ with $S \subseteq L$.*

Proof. One just has to show the implication *4 ⇒1*. Let N be a nondeterministic acceptor for L with a polynomial time-bound on every $S \in \mathcal{P}$ with $S \subseteq L$. Let f be a 1-invertible $\leq_m^p$-reduction from $T(L)$ to L. Let h be a proof system for L with $h(\langle M, x, 0^s, \alpha \rangle) = M(x)$ if α denotes an accepting path of N on input $f(\langle M, x, 0^s \rangle)$, and $\text{time}_M(x) \leq s$. We show that h is optimal. Let g be a proof system computed by a machine M in polynomial time p. Observe that for any x in the domain of g, $g(x) = h(\langle M, x, 0^{p(|x|)}, \alpha \rangle)$ where α denotes an accepting path of N on input $f(\langle M, x, 0^{p(|x|)} \rangle)$. As $f(T(L)_M) \in \mathcal{P}$, the assumption about N implies that an accepting path α of polynomial length exists. This shows that h simulates g. ∎

Similar to the discussion at the end of Section 3 the proof of the implication *4* $\Rightarrow$*1* goes already through if there is a many-one reduction $f \in \mathcal{FP}$ of $T(L)$ to L such that $f(T(L)_M) \in \mathcal{P}$ for any transducer M computing a proof system of L. For the implication *3* $\Rightarrow$*1* it suffices that $f(T(L)_M) \in \mathcal{NP}$ for any transducer M computing a proof system of L. This is already guaranteed when there is a honest many-one reduction from $T(L)$ to L ($f \in \mathcal{FP}$ is called *honest* if there is a polynomial p such that $|x| \leq p(|f(x)|)$ for any x in the domain of f).

With very few modifications the construction in the proof of Theorem 14 can be adjusted to the nondeterministic case. This yields

Theorem 19 *Let $t : I\!N \to I\!N$ be a time-constructible function such that for every polynomial p there is a number n with $p(n) \leq t(n)$. Then there is a language $L \in$ co-NTIME$(t(n))$ for which no optimal proof system exists.*

It is shown in [7] that any set that is $\leq^p_m$-reducible to a set possessing an optimal proof system has an optimal proof system, too. Together with Theorem 19 this yields the following corollary.

Corollary 20 *No set $\leq^p_m$-hard for co-$\mathcal{NE}$ has an optimal proof system.*

6 Acknowledgments

I whish to thank Johannes Köbler for his permanent advice, and for the motivating discussions about the topic.

References

[1] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity II.* Springer-Verlag, 1990.

[2] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity I.* Springer-Verlag, 2 edition, 1995.

[3] Shai Ben-David and Anna Gringauze. On the existence of optimal propositional proof system and oracle-relativized propositional logic. Technical Report TR98-021, Electronic Colloquium on Computational Complexity, 1998.

[4] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[5] Jun Gu, Paul Purdom, John Franco, and Benjamin Wah. *Algorithms for the Satisfiability Problem.* Cambridge University Press, to appear.

[6] Yuri Gurevich. The logic in computer science column. *Bulletin of the European Association for Theoretical Computer Science*, (35):71–82, 1988.

[7] Johannes Köbler and Jochen Messner. Complete problems for promise classes by optimal proof systems for test sets. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, CC 98*, pages 132–140, 1998.

[8] 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.

[9] Leonid A. Levin. Universal search problems (in russian). *Problemy Peredachi Informatsii*, 9(3):115–116, 1973. English translation in *Problems of Information Transmission* 9(3):265–266, 1973. Revised translation as an appendix to [15].

[10] Jochen Messner. On optimal algorithms and optimal proof system. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science (STACS'99)*. Springer-Verlag, 1999.

[11] 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 (STACS'98)*, number 1373 in Lecture Notes in Computer Science, pages 477–487. Springer-Verlag, 1998.

[12] Zenon Sadowski. On an optimal deterministic algorithm for SAT. Presented on *the Annual Conference of the European Association for Computer Science Logic*, CSL '98. To be published in the LNCS-series of Springer-Verlag.

[13] Zenon Sadowski. On an optimal quantified propositional proof system and a complete language for $\mathcal{NP} \cap$ *co-*$\mathcal{NP}$. In *Proceedings of the 11-th International Symposium on Fundamentals of Computing Theory, FCT'97*, number 1279 in Lecture Notes in Computer Science, pages 423–428. Springer-Verlag, 1997.

[14] Claus-Peter Schnorr. Optimal algorithms for self-reducible problems. In *Proceedings of the third International Colloquium on Automata, Languages, and Programming (ICALP'76)*, pages 322–337. Edinburgh University Press, 1976.

[15] Boris A. Trakhtenbrot. A survey of russian approaches to perebor (brute-force search) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.