Electronic Colloquium on Computational Complexity, Report No. 107 (2008)

Optimal Proof Systems and Complete Languages

Zenon Sadowski

Institute of Mathematics,
University of Białystok
15-267 Białystok, ul. Akademicka 2, Poland
e-mail: sadowski@math.uwb.edu.pl

Abstract. We investigate the connection between optimal propositional proof systems and complete languages for promise classes. We prove that an optimal propositional proof system exists if and only if there exists a propositional proof system in which every promise class with the test set in co-NP is representable. Additionally, we prove that there exists a complete language for UP if and only if there exists a propositional proof system such that UP is representable in it. UP is the standard promise class with the test set in co-NP.
Key words: Optimal proof systems, promise classes, complete languages

1 Introduction

Although there are many different formal systems for proving propositional tautologies in logic textbooks, they all fall under the concept of an abstract propositional proof system (a proof system for $TAUT$) introduced by S. Cook and R. Reckhow [5]. In order to compare the relative strength of different proof systems for $TAUT$ we use the notion of simulation [10] and the notion of p-simulation [5]. A proof system for $TAUT$ is optimal (p-optimal) if and only if it simulates (p-simulates) any other proof system for $TAUT$. The still unresolved problem of the existence of an optimal (p-optimal) proof system for $TAUT$ was posed by J. Krajíček and P. Pudlák [10] in 1989.

The notion of p-simulation between proof systems for $TAUT$ is similar to the notion of reducibility between languages. Analogously, the notion of a p-optimal proof system for $TAUT$ should correspond to the notion of a complete language.

Informally, a class of languages is a promise class if the languages in this class are accepted by nondeterministic polynomial-time clocked Turing machines which obey special conditions (promises). Common promise classes are **UP**, $\mathbf{NP} \cap \mathbf{co\text{-}NP}$, and **BPP**. It is still open whether there exist complete languages for these classes. The reason lies in the undecidability of the problem of whether a given nondeterministic polynomial-time Turing machine indeed obeys the promise of any of these classes. Moreover, there exist relativizations for which these classes do not have complete languages (see [8]).

Recently, O. Beyersdorff [3] introduced the notion of a disjoint **NP**-pair representable in a given propositional proof system f. The disjointness of such a pair is expressible by a sequence of propositional tautologies with short f-proofs.

ISSN 1433-8092

He also considered the complexity class of all disjoint **NP**-pairs representable in a proof system f. In this paper we extend these notions to any promise class with a propositionally expressible promise. It results in the notion of a language representable in a given proof system f and in the notion of a promise class representable in f. O. Beyersdorff proved [2] that the class of all disjoint **NP**-pairs has a complete pair if and only if there exists a proof system for $TAUT$ in which every disjoint **NP**-pair is representable. We prove the analogous theorem for the class **UP**. Namely, **UP** has a complete language if and only if there exists a proof system for $TAUT$ such that **UP** is p-representable in it.

It turns out that there is a close connection between optimal proof systems and complete languages for promise classes, namely, the existence of optimal proof systems implies the existence of complete languages for various promise classes (see [9]). Let us mention two exemplary results of this type. A. Razborov [14] observed that the existence of an optimal proof system suffices to guarantee the existence of complete disjoint **NP**-pairs. J. Messner and J. Torán showed [12] that a complete language for **UP** exists in case there is a p-optimal proof system for $TAUT$. The converses of these implications probably do not hold [6] and in this paper we address the question of just why it is so.

It seems that the promise that a Turing machine computes a proof system for $TAUT$, or more precisely it produces only propositional tautologies, is the hardest one among those promises which are propositionally expressible. Therefore, the sufficient condition for the existence of an optimal proof system should be as strong as the existence of a complete language for every promise class with a propositionally expressible promise. At present, this intuition is only supported by the result of J. Messner [11] which states that a p-optimal proof system for $TAUT$ exists if and only if every promise function class with a test set polynomial-time reducible to $TAUT$ has a complete function (see also [9]). The analogous theorem in the setting of promise language classes instead of promise function classes is missing. The main result from this paper, that the existence of an optimal proof system for $TAUT$ is equivalent to the existence of a proof system for $TAUT$ in which any promise class with the test set in co-**NP** is representable, may be treated as the first step in this direction.

2 Preliminaries

We assume some familiarity with basic complexity theory and refer the reader to [1] and [13] for standard notions and for definitions of complexity classes appearing in the paper. The class of all disjoint pairs (A, B) of **NP**-languages is denoted by Dis**NP**.

The symbol Σ denotes a certain fixed finite alphabet throughout the paper. The set of all strings over Σ is denoted by $\Sigma^\star$. For a string x, $|x|$ denotes the length of x.

Given two languages L_1 and L_2 ($L_1, L_2 \subseteq \Sigma^\star$), we say that L_1 is polynomial-time many-one reducible to L_2 if and only if there exists a polynomial-time

computable function $f : \Sigma^\star \longrightarrow \Sigma^\star$ such that $x \in L_1$ if and only if $f(x) \in L_2$ holds for any $x \in \Sigma^\star$.

We use Turing machines (acceptors and transducers) as our basic computational model. We will not distinguish between a machine and its code. For a Turing machine M the symbol $L(M)$ denotes the language accepted by M.

We consider deterministic and nondeterministic polynomial-time clocked Turing machines (PTM and $NPTM$ for short) with uniformly attached standard clocks that stop their computations in polynomial time (see [1]). We impose some restrictions on our encoding of these machines. From the code of any polynomial-time clocked Turing machine we can easily detect (in polynomial time) the polynomial p_N which is its polynomial-time bound.

Let D_1, D_2, D_3, ... and N_1, N_2, N_3, ... be respectively standard enumerations of all deterministic and nondeterministic polynomial-time clocked Turing machines. For any class of languages $\mathbf{C}$, we say that $\mathbf{C}$ has an uniform enumeration if and only if there exists a recursively enumerable list of nondeterministic polynomial-time clocked Turing machines $N_{i_1}, N_{i_2}, N_{i_3}, ...$ such that $\{L(N_{i_k})\colon k \geq 1\ \} = \mathbf{C}$.

We consider only languages over the alphabet Σ (this means that, for example, boolean formulas have to be suitably encoded). The symbol $TAUT$ denotes the set (of encodings) of all propositional tautologies over a fixed adequate set of connectives. Finally, $\langle ., \ldots, . \rangle$ denotes some standard polynomial-time computable tupling function.

3 Propositional proof systems

The concept of an abstract propositional proof system, subsuming all propositional proof systems used in practice, was introduced by S. Cook and R. Reckhow [5] in the following way:

Definition 1. *A proof system for $TAUT$ is a polynomial-time computable function $f : \Sigma^\star \stackrel{onto}{\longrightarrow} TAUT$.*

A string w such that $f(w) = \alpha$ we call an f-proof of a formula α. We write $f \vdash^* \alpha_n$ if and only if $\{\alpha_n\colon \mathrm{n} \geq 1\}$ is a sequence of tautologies with polynomial-size f-proofs. A polynomially bounded proof system for $TAUT$ (which allows short proofs to all tautologies) exists if and only if **NP=co-NP** (see [5]).

Proof systems are compared according to their strength using the notion of simulation and the presumably stronger notion of p-simulation.

Definition 2. *(Krajíček, Pudlák) Let h, h' be two proof systems for $TAUT$. We say that h simulates h' if there exists a polynomial p such that for any $x \in TAUT$, if x has a proof of length n in h', then x has a proof of length $\leq p(n)$ in h.*

Definition 3. *(Cook, Reckhow) Let h, h' be two proof systems for $TAUT$. We say that h p-simulates h' if there exists a polynomial-time computable function*

$\gamma : \Sigma^\star \longrightarrow \Sigma^\star$ such that for every $x \in TAUT$ and every $w \in \Sigma^\star$, if w is a proof of x in h', then $\gamma(w)$ is a proof of x in h.

In other words, γ translates h'-proofs into h-proofs of the same formula.

The notions of an optimal proof system for $TAUT$ and a p-optimal proof system for $TAUT$ were introduced by J. Krajíček and P. Pudlák [10].

Definition 4. *A proof system for $TAUT$ is* optimal (p-optimal) *if and only if it simulates (p-simulates) any proof system for $TAUT$.*

We will study the problem of the existence of an optimal proof system and the problem of the existence of a p-optimal proof system from computational-complexity perspective.

4 Promise classes representable in a proof system

A nondeterministic polynomial-time clocked Turing machine which is the computational model of a given promise (semantic) class should obey the special condition, called the promise of the class. It can be illustrated by an example of the class **UP**. We call a nondeterministic Turing machine categorical or unambiguous if it has the following property: for any input x there is at most one accepting computation. We define **UP**=$\{L(N_i)$: N_i is categorical$\}$.

Let T be any formal theory whose language contains the language of arithmetic. We say that T is "reasonable" if and only if T is sound (that is, in T we can prove only true theorems) and the set of all theorems of T is recursively enumerable. Let N be any $NPTM$. The notation $T \vdash$ "N is categorical" means that the first order formula expressing the categoricity of N is provable in T. We say that **UP** is representable in T if and only if for any $A \in$ **UP** there exists an $NPTM$ N such that $T \vdash$ "N is categorical". J. Hartmanis and L. Hemachandra [8] proved that **UP** has a complete language if and only if it has an uniform enumeration (see also [4]). It follows from Naming Lemma [7] that, the existence of a uniform enumeration of **UP** is equivalent to the existence of a "reasonable" theory T such that **UP** is representable in T (see also [8]). Therefore, the problem of the existence of a complete language for **UP** can be characterized in terms of a uniform representability of **UP** in a first order arithmetic theory T.

In this section we show that this problem can be also characterized in terms of a nonuniform representability of **UP** in a propositional proof system. In this case the promise of the class is expressed as the sequence of propositional tautologies with short proofs. We begin with the introduction of the necessary machinery.

Following J. Messner's approach [11], we define promise classes in a very general way. A promise R is described as a binary predicate on the Cartesian product of the set of all $NPTMs$ and the set of all strings, i. e. , $R(N, x)$ means that N obeys a promise R on input x. An $NPTM$ N is called an R-machine if and only if N obeys R on any input $x \in \Sigma^\star$. For a given promise predicate R we define the class of languages $C_R = \{L(N)$: N is an R-machine$\}$ and call it the promise class generated by R.

Definition 5. *(Messner) A class of languages C is called a promise class if and only if $C = C_R$ for some promise predicate R.*

The following notion of the test set for a promise class C_R serves as a tool for estimating the complexity of the promise R of this class.

Definition 6. *By the test set for a promise class C_R we mean the set $T_R = \{\langle N, 0^n, 0^{p_N(n)}\rangle$: n is a natural number, N is an $NPTM$ such that $R(N, x)$ holds for any x such that $|x| = n\}$*

The notion of the test set for C_R corresponds to the notion of the generic and length-only dependent test set from [9].

We are especially interested in the situation when the fact that a given $NPTM$ is an R-machine can be expressed propositionally, as a sequence of propositional tautologies. It can be done only when the promise R has an appropriate complexity.

To any $NPTM$ N we will assign the set $D_R^N = \{\ \alpha_1^N, \alpha_2^N, \alpha_3^N, ...\ \}$ of propositional formulas such that α_n^N is a propositional tautology if and only if $R(N, x)$ holds for any x such that $|x| = n$. So, for any $NPTM$ N it holds: N is an R-machine if and only if $D_R^N \subset TAUT$.

It should be possible to construct the formulas α_n^N for different sets D_R^N, corresponding to different $NPTM$s, easily and in an uniform manner. This leads to the following definitions:

Definition 7. *By a propositional description of a promise R we mean a set $D_R = \{\alpha_n^N$: N is an $NPTM$, n is a natural number$\}$ of propositional formulas fulfilling conditions (1) – (3):*

(1) Adequacy:
α_n^N is a propositional tautology if and only if $R(N, x)$ holds for any x such that $|x| = n$.

(2) Uniform constructibility:
There exists a polynomial time computable function f such that for any $NPTM$ N and for any n natural

$$f(\langle N, 0^n, 0^{p_N(n)}\rangle) = \alpha_n^N$$

(3) Local recognizability:
For any fixed $NPTM$ N, the set $D_R^N = \{\alpha_1^N, \alpha_2^N, \alpha_3^N, ...\}$ is in **P**.

Definition 8. *We say that a promise R is propositionally expressible if and only if there exists a propositional description of R.*

The next lemma will be needed in Section 5 in the proof of the main result of the paper.

Lemma 1. *Any promise R such that the class C_R possesses the test set T_R in co-***NP** *is propositionally expressible.*

Proof. (See also [12]) Since $TAUT$ is co-**NP** complete and paddable, there exists a polynomial-time and length-increasing function $f : \Sigma^\star \longrightarrow \Sigma^\star$, reducing T_R to $TAUT$.

Let $Form$ denotes the set of all propositional formulas. Since $Form \in \mathbf{P}$, there exists a polynomial-time and length-increasing function $\tilde{f} : \Sigma^\star \longrightarrow Form$ such that for any $x \in \Sigma^\star$ it holds: $x \in T_R$ if and only if $\tilde{f}(x) \in TAUT$. Using the function $\tilde{f}$ we can define α_n^N to be the formula $\tilde{f}(\langle N, 0^n, 0^{p_N(n)}\rangle)$, for any $NPTM$ N and any n natural. Clearly, the adequacy and uniform constructibility conditions are fulfilled.

We claim that $D_R^N \in \mathbf{P}$. Indeed, in order to test whether a given formula α belongs to D_R^N we generate the strings from the set T_R up to a given length and check whether the image of one of these strings after applying $\tilde{f}$, coincides with α. Thus, the local recognizability condition is fulfilled.

□

Let R be a propositionally expressible promise and let $D_R = \{\alpha_n^N\colon N$ is an $NPTM$, n is a natural number$\}$ be its propositional description. Let h be a proof system for $TAUT$.

Definition 9. *A language A is weakly D_R-representable in h if and only if there exists an $NPTM$ K such that conditions (1) – (2) are fulfilled:*

(1) $L(K) = A$
(2) $h \vdash^* \alpha_n^K$

Definition 10. *A language A is strongly D_R-representable in h if and only if there exists an $NPTM$ K such that conditions (1) – (2) are fulfilled:*

(1) $L(K) = A$
(2) There exists a polynomial time algorithm that on input 0^n produces an h-proof of α_n^K, for any n natural.

Finally, we have the following definitions:

Definition 11. *A promise class C_R is representable in h if and only if there exists a propositional description D_R of R such that any language $A \in C_R$ is weakly D_R-representable in h.*

Definition 12. *A promise class C_R is p-representable in h if and only if there exists a propositional description D_R of R such that any language $A \in C_R$ is strongly D_R-representable in h.*

Now we present the above mentioned characterization of the problem of the existence of a complete language for **UP**.

Theorem 1. *There exists a complete language for* **UP** *if and only if there exists a propositional proof system h such that* **UP** *is p-representable in h.*

Proof. $(i) \rightarrow (ii)$ Let us notice that **UP** has the test set in co-**NP**. Therefore, the promise R of **UP** is propositionally expressible. Let $D_R = \{\alpha_n^N$: N is an $NPTM$, n is a natural number$\}$ be a propositional description of this promise. It follows from J. Hartmanis and L. Hemachandra's result [8], that **UP** has a uniform enumeration. Let G be a Turing machine generating the codes of the machines from the sequence N_{i_1}, N_{i_2}, N_{i_3},... forming a uniform enumeration of the class **UP**.

Now we can define a propositional proof system h in which **UP** is p-representable. We say that a string $v \in \Sigma^\star$ is in *good form* if

$$v = \langle Comp - G, 0^n, 0^{p_{N_{i_j}}(n)} \rangle$$

where n is a natural number, $Comp - G$ is a computation of the machine G. This computation produces a certain machine N_{i_j} from the uniform enumeration of the class **UP**. The polynomial $p_{N_{i_j}}$ bounds the running time of N_{i_j}.

Let α_0 be a certain fixed propositional tautology. Let f be the function existing on the ground of the uniform constructibility condition from the definition of D_R. We define $g : \Sigma^\star \longrightarrow \Sigma^\star$ in the following way: $g(v) = \alpha_n^{N_{i_j}}$ if v is in good form ($v = \langle Comp - G, 0^n, 0^{p_{N_{i_j}}(n)} \rangle$) N_{i_j} is the machine produced by the computation $Comp - G$, and $f(\langle N_{i_j}, 0^n, 0^{p_{N_{i_j}}} \rangle) = \alpha_n^{N_{i_j}}$; otherwise $g(v) = \alpha_0$.

It follows from the uniform constructibility condition from the definition of D_R that g is polynomial-time computable. It follows from the adequacy condition that $g : \Sigma^\star \longrightarrow TAUT$. Let f' be any standard Frege propositional proof system. Combining g with f' we obtain a propositional proof system $h : \Sigma^\star \stackrel{onto}{\longrightarrow} TAUT$.

$$h(y) = \begin{cases} g(v) & if \text{ y=0v and v is in good form} \\ f'(v) & if \text{ y=1v} \\ \alpha_0 & otherwise \end{cases}$$

Let A be any language from **UP**. There exists a machine N_{i_k} from the uniform enumeration of the class **UP** such that $L(N_{i_k}) = A$, and $p_{N_{i_k}}$ is its polynomial-time bound. Let $Comp - G$ be the computation of G producing the code of the machine N_{i_k}. The function $\lambda(0^n) = 0\langle Comp - G, 0^n, 0^{p_{N_{i_k}}(n)} \rangle$ produces an h-proof of $\alpha_n^{N_{i_k}}$ in polynomial time in n.

$(ii) \rightarrow (i)$ Assume that there exists a propositional proof system h such that **UP** is p-representable in it.

There exists a propositional description $D_R = \{\alpha_n^N$: N is an $NPTM$, n is a natural number$\}$ of the promise R of **UP** such that any language $A \in$ **UP** is strongly D_R-representable in h.

Let us consider the language

$$L = \{\langle N, x, Proof - \alpha_{|x|}^N, 0^{p_N(|x|)} \rangle : x \in L(N)\}$$

where N is an $NPTM$, p_N is the polynomial that bounds the running time of N, x is a string, $Proof - \alpha_{|x|}^N$ is an h-proof of $\alpha_{|x|}^N$, where $\alpha_{|x|}^N$ is a propositional

formula from the propositional description of the promise of **UP**, $0^{p_N(|x|)}$ is the sequence of zeros (padding).

We claim that L is the desired **UP** complete language.

(a) $L \in \mathbf{UP}$.
It is assured by the following nondeterministic Turing machine K working in polynomial time.
1. On input w the machine deterministically checks whether w is in the form $\langle N, x, Proof - \alpha^N_{|x|}, 0^{p_N(|x|)}\rangle$. If so then K comes to point 2, otherwise K rejects w.
2. K runs N on input x. If N accepts x then K accepts w.

Let us notice that if w is in the form $\langle N, x, Proof - \alpha^N_{|x|}, 0^{p_N(|x|)}\rangle$ then $\alpha^N_{|x|}$, as a formula possessing an h-proof is a propositional tautology. From this it follows that for any input of length $|x|$ (and in particular for x), N has at most one accepting computation. This clearly forces K to be a categorical Turing machine.

(b) L is a complete language for **UP**.
Let L' be any language in **UP**. Since L' is strongly D_R-representable in h, there exists an $NPTM$ K such that $L(K) = A$, and there exists a deterministic polynomial-time Turing machine M that on input 0^n produces an h-proof of α^K_n, for any n natural. The function $f : \Sigma^\star \longrightarrow \Sigma^\star$ defined by

$$f(x) = \langle K, x, Proof - \alpha^K_{|x|}, 0^{p_K(|x|)}\rangle$$

is the polynomial time many-one reduction of L' to L. The string $Proof - \alpha^K_{|x|}$ from the definition of f is the h-proof of the formula $\alpha^K_{|x|}$ produced by the deterministic Turing machine M.

□

5 Main results

J. Messner [11] considered the family of all proof systems for $TAUT$ as a promise function class. He proved that a p-optimal proof system exists if and only if any promise function class with a test set polynomial-time reducible to $TAUT$ possesses a complete function. Our intention was to find an analogous theorem in the setting of promise language classes. In our results from this section we characterize the existence of optimal proof systems in terms of a nonuniform presentability of promise classes in a proof system. For most promise classes, having a complete language and a uniform enumeration (a uniform representation in an arithmetic theory) are equivalent. Similarly, it seems that for promise classes a nonuniform p-presentability in a proof system is close to the possession of a complete language.

In our chracterization of the problem of the existence of optimal and p-optimal proof systems the families of all easy and all **NP**-easy subsets of $TAUT$ play a very important role.

Definition 13. *By an easy subset of $TAUT$ we mean a set A such that $A \subset TAUT$ and $A \in \mathbf{P}$.*

Definition 14. *By an* **NP***-easy subset of $TAUT$ we mean a set A such that $A \subset TAUT$ and $A \in \mathbf{NP}$.*

The next lemma shows that for every easy subset of $TAUT$ there exists a proof system in which tautologies from this set have short and easily constructible proofs.

Lemma 2. *(Messner, Torán [12]) If A is an easy subset of $TAUT$ then there exists a proof system $f : \Sigma^\star \stackrel{onto}{\longrightarrow} TAUT$ and a polynomial-time computable function t that on input α produces f-proof of α, for any tautology α in A. That is for every $\alpha \in A$, $f(t(\alpha)) = \alpha$.*

For any transducer N we will denote by f_N the function computed by N.

Definition 15. *(see [15]) A Turing transducer N is called sound if f_N maps $\Sigma^\star$ into $TAUT$ ($f_N : \Sigma^\star \longrightarrow TAUT$).*

To any polynomial-time clocked transducer M we will assign the set $A_M = \{Sound_M^1, Sound_M^2, Sound_M^3, ...\}$ of propositional formulas such that: $Sound_M^n$ is a propositional tautology if and only if for every input of length n, the machine M outputs a propositional tautology (see [15]).

So, for any polynomial-time clocked transducer M, it holds: M is sound if and only if $A_M \subset TAUT$.

Moreover, the formulas describing the soundness of Turing machines possess the following *global uniformity property*: There exists a polynomial-time computable function f such that for any polynomial-time clocked transducer M with time bound p_M and for any $w \in \Sigma^\star$

$$f(\langle M, w, 0^{p_M(|w|)}\rangle) = Sound_M^{|w|}$$

Let us proceed to the main results of the paper.

Theorem 2. *Statements (i) - (iii) are equivalent:*

(i) There exists an optimal propositional proof system.
*(ii) There exists a propositional proof system in which any promise class with the test set in co-***NP** *is representable.*
(iii) There exists a propositional prof system in which the class of all **NP***-easy subsets of $TAUT$ is representable.*

Proof. $(i) \rightarrow (ii)$ Let *Opt* be an optimal propositional proof system and let **C** be any promise class defined by a promise R and possessing the test set T_R in co-**NP**. It follows from Lemma 1 that there exists a propositional description $D_R = \{\alpha_n^N : N \text{ is an } NPTM, n \text{ is a natural number}\}$ of R. Let A be any language from **C** and let K be an $NPTM$ such that $L(K) = A$. By the adequacy condition and by the local recognizability condition from Definition 7, the set $D_R^K = \{\alpha_1^K,$

α_2^K, α_3^K, ...} is an easy subset of $TAUT$. By Lemma 2 and by the definition of an optimal proof system we have $Opt \vdash^* \alpha_n^K$. As A was chosen arbitrarily, the class **C** is representable in *Opt*. The same reasoning applies to any other promise class with the test set in co-**NP**, so any such promise class is representable in *Opt*.

$(ii) \rightarrow (iii)$ Let h be a propositional proof system in which any promise class with the test set in co-**NP** is representable. Let R denote the promise of the class of all **NP**-easy subsets of $TAUT$. Thus, $R(N, x)$ holds if and only if on input x, if N accepts then x is in $TAUT$. The test set of the class of all **NP**-easy subsets of $TAUT$ is in co-**NP**. Hence, this class is representable in h.

$(iii) \rightarrow (i)$ Let h be a propositional proof system in which the class of all **NP**-easy subsets of $TAUT$ is representable. It follows from the existence of h that there exists the propositional description $D_R = \{\alpha_n^L$: L is an $NPTM$, n is a natural number$\}$ of the promise of this class.

We say that a string $v \in \Sigma^\star$ is in *good form* if

$$v = \langle M, w, N, Proof - \alpha_l^N, Comp - Sound_M^{|w|}, 0^{p_M(|w|)+p_N(l)} \rangle$$

where M is a polynomial-time clocked transducer with p_M time bound, $w \in \Sigma^\star$, N is an $NPTM$ with p_N time bound, $l = |Sound_M^{|w|}|$, $Proof - \alpha_l^N$ is an h-proof of the formula α_l^N from the propositional description of the promise of the class of all **NP**-easy subsets of $TAUT$, $Comp - Sound_M^{|w|}$ is a computation of the machine N accepting the formula $Sound_M^{|w|}$, $0^{p_M(|w|)+p_N(l)}$ is the sequence of zeros (padding).

We call a Turing transducer n-sound if and only if on any input of length n it produces a propositional tautology.

Let us notice that if v is in good form, then M on input w produces a propositional tautology. It can be proved in the following way. If v is in good form then α_l^N, as a formula possessing an h-proof, is a propositional tautology. By the adequacy condition from Definition 7, the machine N obeys, for any input of the length l, the promise of the class of all **NP**-easy subsets of $TAUT$. In consequence, $Sound_M^{|w|}$ is a propositional tautology. This clearly forces M to be n-sound, where $n = |w|$, so M on input w produces a propositional tautology.

Let α_0 be a certain fixed propositional tautology. We define $Opt : \Sigma^\star \longrightarrow \Sigma^\star$ in the following way: $Opt(v) = \alpha$ if v is in good form and α is a propositional tautology produced by M on input w, otherwise $h(v) = \alpha_0$. Clearly, $Opt : \Sigma^\star \stackrel{onto}{\longrightarrow} TAUT$.

Using the global uniformity property for the formulas expressing the soundness of Turing transducers, and using the uniform constructibility condition for the formulas from the propositional description of the promise under consideration, we can check in polynomial time whether v is in good form. From this it follows that *Opt* is polynomial time computable and, in consequence, *Opt* is a propositional proof system.

It remains to prove that *Opt* simulates any propositional proof system. Let g be a propositional proof system computed by the polynomial time clocked

transducer K with time bound p_K. The set $A_K = \{Sound^1_K\ , Sound^2_K, Sound^3_K, ...\}$ is an **NP**-easy subset of $TAUT$. Since this set is weakly D_R-representable in h, there exists an $NPTM$ N such that $L(N) = A_K$ and $h \vdash^* \alpha^N_n$. Let α be any propositional tautology and let x be its g-proof. Then α possesses an Opt-proof of the form:

$$v = \langle K, x, N, Proof - \alpha^N_l, Comp - Sound^{|x|}_K, 0^{p_K(|x|)+p_N(l)} \rangle$$

The word $Comp - Sound^{|x|}_K$ is a computation of N accepting $Sound^{|x|}_K$, $l = |Sound^{|x|}_K|$, the word $Proof - \alpha^N_l$ is an h-proof of the formula α^N_l from the propositional description of the promise of the class of all **NP**-easy subsets of TAUT.

Let us notice that by the global uniformity property there exists a polynomial q_1, such that $l = |Sound^{|x|}_K| \leq q_1(|x|)$. Because N is polynomial-time clocked there exists a polynomial q_2 such that $Comp - Sound^{|x|}_K \leq q_2(|x|)$. By the uniform constructibility condition there exists a polynomial q_3 such that $|\alpha^N_l| \leq q_3(l)$ and because $h \vdash^* \alpha^N_l$ there exists a polynomial q_4 such that $|Proof - \alpha^N_l| \leq q_4(l)$. Finally, since l is polynomially related to $|x|$, there exists a polynomial q_5 such that $|Proof - \alpha^N_l| \leq q_5(|x|)$. This proves that Opt simulates g. □

The previous result can be translated to the deterministic case in the following way:

Theorem 3. *Statements (i) - (iii) are equivalent:*

(i) There exists a p-optimal propositional proof system.
*(ii) There exists a propositional proof system in which any promise class with the test set in co-***NP** *is p-representable.*
(iii) There exists a propositional prof system in which the class of all easy subsets of $TAUT$ is p-representable.

Proof. $(i) \rightarrow (ii)$ This follows by the same arguments as in the proof of $(i) \rightarrow (ii)$ from Theorem 2. Let Opt be a p-optimal propositional proof system. Let $\mathbf{C} = \mathbf{C}_R$ be any promise class with the test set T_R in co-**NP**. Let $D_R = \{\alpha^N_n$: N is an $NPTM$, n is a natural number$\}$ be a propositional description of R. Let A be any language such that $A \in \mathbf{C}$ and let K be an $NPTM$ such that $L(K) = A$. For the same reasons as before the set $\{\alpha^K_1, \alpha^K_2, \alpha^K_3, ...\}$ is an easy subset of $TAUT$. By the uniform constructibility condition, any formula α^K_n can be constructed in polynomial time in n. From Lemma 2 it follows that there exists a polynomial-time deterministic Turing machine M that on input 0^n produces an Opt-proof of α^K_n. As A was chosen arbitrarily, the class $\mathbf{C}$ is p-representable in Opt. The rest of the proof runs as before.

$(ii) \rightarrow (iii)$ Let h be a propositional proof system in which any promise class with the test set in co-**NP** is p-representable. Let R denotes the promise of the class of all easy subsets of TAUT. Thus $R(N, x)$ holds if and only if on input x, N only makes deterministic moves and if N accepts then x is in $TAUT$. Clearly,

the test set of the class of all easy subsets of $TAUT$ is in co-**NP**. Hence, this class is p-representable in h.

$(iii) \rightarrow (i)$ Let h be a propositional proof system in which the class of all easy subsets of $TAUT$ is p-representable. Let $D_R = \{\alpha_n^K$: K is any $NPTM$, n is a natural number $\}$ be the propositional description of the promise of this class. We say that a string $v \in \Sigma^\star$ is in *good form* if

$$v = \langle M, w, N, Proof - \alpha_l^N, Comp - Sound_M^{|w|}, 0^{p_M(|w|)+p_N(l)} \rangle$$

where the appropriate symbols mean the same as before. We define $Opt : \Sigma^\star \stackrel{onto}{\longrightarrow} TAUT$ analogously as in the proof of Theorem 2: $Opt(v) = \alpha$ if v is in good form and α is a propositional tautology produced by M on input w, otherwise $h(v) = \alpha_0$, where α_0 is a certain fixed propositional tautology.

It remains to prove that Opt p-simulates any propositional proof system. Let g be a propositional proof system computed by the polynomial-time clocked transducer K with time bound p_K. The set $A_K = \{Sound_K^1, Sound_K^2, Sound_K^3, ...\}$ is an easy subset of $TAUT$. This set is strongly D_R-representable in h, so there exists an $NPTM$ N such that $L(N) = A_K$ and there exists a deterministic polynomial-time Turing machine M that on input 0^n produces an h-proof of α_n^N, for any n natural. Since the formulas from the set $\{\alpha_1^N, \alpha_2^N, \alpha_3^N, ...\}$ are propositional tautologies and since these formulas fulfill the adequacy condition, N is a deterministic Turing machine and N accepts only propositional tautologies. The function $t : \Sigma^\star \longrightarrow \Sigma^\star$ defined by

$$t(x) = \langle K, x, N, Proof - \alpha_l^N, Comp - Sound_K^{|x|}, 0^{p_K(|x|)+p_N(l)} \rangle$$

translates g-proofs into Opt-proofs.

The word $Comp - Sound_K^{|x|}$ in the definition of t is the computation of the machine N accepting the formula $Sound_K^{|x|}$, $l = |Sound_K^{|x|}|$, $Proof - \alpha_l^N$ is an h-proof of the formula α_l^N.

From the global uniformity property and from the fact that N is deterministic and polynomial-time clocked it follows that $Comp - Sound_K^{|x|}$ can be constructed in polynomial time in $|x|$. Similarly, from the global uniformity property and from the fact that the set A_K is strongly D_R-representable in h, it follows that $Proof - \alpha_l^N$ can be constructed in polynomial time in $|x|$. This proves that t is polynomial time computable. □

We proved [15] that there exists a p-optimal proof system for $TAUT$ if and only if the class of all easy subsets of $TAUT$ is uniformly enumerable. It can be otherwise stated thus: there exists a p-optimal proof system for $TAUT$ if and only if there exists a "reasonable" arithmetic theory T such that the class of all easy subsets of $TAUT$ is representable in it (see [7]). In this paper we replaced representability in an arithmetic theory by representability in a proof system, in the characterization of the existence of a p-optimal proof system in terms of easy subsets of $TAUT$. It is typical for proof complexity that an arithmetic theory coincides with a proof system for $TAUT$ and the latter is a nonuniform version of the former.

References

1. J.Balcazar,J.Díaz and J.Gabarró, Structural Complexity I (Springer-Verlag, Berlin, 1995).
2. O.Beyersdorff, Tuples of disjoint NP-sets, Thechnical Report TR 05-123, Electronic Colloquium on Computational Complexity, 2005.
3. O.Beyersdorff, Disjoint NP-Pairs and Propositional Proof systems, PhD Thesis, Humbold-Universität zu Berlin, July 2006.
4. H. Buhrman, S. Fenner, L. Fortnow, D. van Melkebeek, Optimal proof systems and sparse sets, Proc. Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 1770, Springer - Verlag, Berlin, 2000.
5. S.Cook and R.Reckhow, The relative efficiency of propositional proof systems, J.Symb.Log.44(1979) 36-50
6. C. Glasser, A. L. Selman, S. Sengupta and L. Zhang, Disjoint NP-pairs, SIAM Journal on Computing, 33(6), (2004) 1369 –1416.
7. J.Hartmanis, Independence Results about Context-Free Languages and Lower Bounds, Technical Report, TR 84-606, Department of Computer Science Cornell University, May 1984.
8. J.Hartmanis, L.Hemachandra, Complexity classes without machines: On complete languages for UP, Theoretical Computer Science 58(1988) 129-142.
9. J. Köbler, J. Messner, J. Torán, Optimal proof systems imply complete sets for promise classes, Information and Computation 184 (2003) 71 – 92.
10. J.Krajíček and P.Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations, Journal of Symbolic Logic 54 (1989) 1063-1079.
11. J.Messner, On the simulation order of proof systems, PhD Thesis, Universität Ulm, December 2000.
12. J.Messner, J.Torán, Optimal proof systems for Propositional Logic and complete sets, in: Proc. 15th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 1373 (Springer,Berlin,1998) 477-487.
13. C.Papadimitriou, Computational Complexity, (Addison - Wesley, 1994).
14. A. Razborov, On provably disjoint NP-pairs, Technical Report 94–006, Electronic Colloquium on Computational Complexity, 1994.
15. Z. Sadowski, On an optimal propositional proof system and the structure of easy subsets of TAUT, Theoretical Computer Science, 288 (1), 2002, 181 – 193.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/