Electronic Colloquium on Computational Complexity, Report No. 77 (2005)

On a $D-N-$optimal acceptor for TAUT

Zenon Sadowski

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

Abstract. A deterministic algorithm recognizing $TAUT$ is a D-N-optimal acceptor for $TAUT$ if no other nondeterministic algorithm accepting $TAUT$ has more than a polynomial speed-up over its running time on instances from $TAUT$.
We prove that the existence of a D-N-optimal acceptor for $TAUT$ is equivalent to the existence of an optimal and automatizable propositional proof system and to the existence of a suitable recursive presentation of the class of all **NP**-easy (acceptable by nondeterministic polynomial time machines) subsets of $TAUT$.
Additionally we show that the question of whether every proof system is weakly automatizable is equivalent to the questions: of whether every disjoint **NP**-pair is **P**-separable and of whether every function in **NPSV** has a total extension in **FP** and prove that the existence of a D-N-optimal acceptor for $TAUT$ implies that every disjoint **NP**-pair is **P**-separable.

1 Introduction

A **deterministic** algorithm recognizing $TAUT$ (the set of all propositional tautologies) is optimal if no other **deterministic** algorithm recognizing $TAUT$ has more than a polynomial speed-up over its running time. If the optimality property is stated only for any input string x which belongs to $TAUT$ and nothing is claimed for other x's, we name such an algorithm as an optimal deterministic acceptor for $TAUT$ (a D-D-optimal acceptor for $TAUT$ in our nomenclature). An N-N-optimal acceptor for $TAUT$ is a **nondeterministic** algorithm accepting $TAUT$ and such that no other **nondeterministic** algorithm accepting $TAUT$ has more than a polynomial speed-up over its running time (obviously on instances from $TAUT$).

In the definition of an optimal deterministic acceptor for $TAUT$, the collection $\mathcal{A}$ of all deterministic algorithms recognizing $TAUT$ occurs and we demand that an optimal acceptor belongs to $\mathcal{A}$. We propose to loosen this restriction and allow the situation when an optimal acceptor computes in different mode than algorithms from collection $\mathcal{A}$. This leads to the following definition: a **deterministic** algorithm recognizing $TAUT$ is a D-N-optimal acceptor for $TAUT$ if no other **nondeterministic** algorithm accepting $TAUT$ has more than a polynomial speed up over its running time on instances from $TAUT$. An N-D-optimal acceptor for $TAUT$ can be defined analogously.

ISSN 1433-8092

These two new optimal acceptors are natural variants of the previously studied ones. Moreover, the importance of the notion of a D-N-optimal acceptor for $TAUT$ lies in the fact that the existence of a D-N-optimal acceptor for $TAUT$ implies that **NP=co-NP** is equivalent to **P=NP**.

Propositional proof systems (proof systems for $TAUT$) can be compared using the notion of simulation and the presumably stronger notion of p-simulation. An optimal (p-optimal) proof system would have proofs which are no more than polynomially longer than any other proof system. The existence of an optimal proof system for $TAUT$ and the existence of a p-optimal proof system are interesting open problems posed by J. Krajíček and P. Pudlák [7] in 1989.

The same authors [7] proved that the existence of a D-D-optimal acceptor for $TAUT$ is equivalent to the existence of a p-optimal proof system for $TAUT$. Likewise, it is easily seen that the existence of an N-N-optimal acceptor for $TAUT$ is equivalent to the existence of an optimal proof system for $TAUT$ (see [10], [8]). Thus, the existence of optimal proof systems and the existence of optimal acceptors are different facets of the same problems. In this paper we develop further this connection and prove that the existence of a D-N-optimal acceptor for $TAUT$ is equivalent to the existence of an optimal and automatizable proof system for $TAUT$. Automatizability is a crucial notion for automated theorem proving. A proof system is automatizable if there is a deterministic procedure to find proofs in this system, in polynomial time with respect to the smallest proof in this system (see [1]). It seems that the stronger proof system is, the more difficult it is to search proofs in it. The investigation of the problem of the existence of a D-N-optimal acceptor may confirm this hypothesis.

It is not currently known whether **UP** and other promise classes have complete languages. J. Hartmanis, and L. Hemachandra pointed out in [4] that **UP** possesses complete languages if and only if there is a recursive enumeration of polynomial time clocked Turing machines covering all languages from this class. Earlier, the question of whether **NP** $\cap$ **co-NP** possesses complete languages was related to an analogous statement (see [5]).

It was observed in [10] that the question of the existence of p-optimal (optimal) proof systems for $TAUT$ (and hence D-D and N-N-optimal acceptors) can be characterized in a similar manner. For example, there exists a p-optimal proof system for $TAUT$ if and only if there is a suitable recursive presentation of the class of all easy (recognizable by deterministic polynomial time machines) subsets of $TAUT$. In this paper we make this picture complete and prove that a D-N-optimal acceptor exists if and only if there is a recursive enumeration of polynomial time clocked deterministic Turing machines covering all **NP**-easy (acceptable by nondeterministic polynomial time machines) subsets of $TAUT$.

In the last section we investigate the question of whether every proof system for $TAUT$ is weakly automatizable. We show that this question is equivalent to the Complexity Theory questions studied before: of whether every disjoint **NP**-pair is **P**-separable and of whether every function in **NPSV** has a total extension in **FP**. Our characterization of the investigated question is analogous to the characterization of the problem of whether every proof system for $TAUT$

admits an effective interpolation obtained by J. Köbler and J. Messner (see [6]). As a corollary we obtained the result that the existence of a D-N-optimal acceptor for $TAUT$ implies that every disjoint **NP**-pair is **P**-separable.

2 Preliminaries

We assume some familiarity with basic complexity theory, see [2]. The symbol Σ denotes, throughout the paper, a certain fixed finite alphabet. The set of all strings over Σ is denoted by $\Sigma^\star$. For a string x, $|x|$ denotes the length of x.

We use Turing machines (acceptors and transducers) as our basic computational model. We will not distinguish between a machine and its code. For a deterministic Turing machine M and an input w, TIME(M; w) denotes the computing time of M on w. When M is a nondeterministic Turing machine TIME(M; w) is defined only for w's accepted by M and denotes the number of steps in the shortest accepting computation of M on w. For a Turing machine M the symbol L(M) denotes the language accepted by M.

A Turing transducer M computes a (not necessarily total) function $f : \Sigma^\star \longrightarrow \Sigma^\star$ defined by $f(x) = M(x)$. Here, $M(x)$ denotes the output of M on input $x \in \Sigma^\star$ ($M(x) = y$ means that M on input x reaches an accepting state and stops with y written on its output tape).

A function $f : \Sigma^\star \longrightarrow \Sigma^\star$ (not necessarily total) is in **NPSV** (the class of all single valued **NP** functions) if there is a nondeterministic Turing transducer N working in polynomial time and such that for every x, $N(x) = f(x)$ if and only if $f(x)$ is defined, and in case $f(x)$ is defined all accepting computations of N produce the same output. By **FP** we denote the class of all functions computed by deterministic Turing transducers working in polynomial time.

We consider deterministic and nondeterministic polynomial time clocked Turing machines with uniformly attached standard $n^k + k$ clocks which stop their computations in polynomial time (see [2]). We impose some restrictions on our encoding of these machines. From the code of any polynomial time clocked Turing machine we can detect easily (in polynomial time) the natural k such that $n^k + k$ is its polynomial time bound.

We consider only languages over the alphabet Σ (this means that, e. g. , 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 Optimal algorithms and optimal proof systems

The notion of an almost optimal deterministic algorithm for $TAUT$, with the optimality property stated only for input strings which belong to $TAUT$, was introduced by J. Krajíček and P. Pudlák [7] (see also [10]). J. Messner [8] named this algorithm as an optimal deterministic acceptor for $TAUT$. In order to unify our notation we propose to name this algorithm as a D-D-optimal acceptor for $TAUT$.

Additionally we propose to introduce the notions of a D-N-optimal acceptor for TAUT. The problem of the existence of a D-N-optimal acceptor for $TAUT$ will be studied in our paper.

Definition 1. *A D-D-optimal acceptor (D-N-optimal acceptor) for $TAUT$ is a deterministic Turing machine M which recognizes $TAUT$ and such that for every deterministic (nondeterministic) Turing machine M' which recognizes (accepts) $TAUT$ there exists a polynomial p such, that for every $\alpha \in TAUT$*

$$TIME(M;\alpha) \leq p(|\alpha|, TIME(M';\alpha))$$

The importance of the question of the existence of a D-N-optimal acceptor for $TAUT$ can be described by the following fact:

Fact 1. If a D-N-optimal acceptor for $TAUT$ exists then statements (i) – (ii) are equivalent:

(i) **P=NP**.
(ii) **NP=co-NP**.

The nondeterministic counterpart of a D-D-optimal acceptor for $TAUT$ is an N-N-optimal acceptor for $TAUT$. This algorithm was named in [8] as an optimal nondeterministic acceptor for $TAUT$ and in [10] as an optimal nondeterministic algorithm for $TAUT$.

We propose to introduce additionally the notion of an N-D-optimal acceptor for $TAUT$.

Definition 2. *An N-N-optimal acceptor (N-D-optimal acceptor) for $TAUT$ is a nondeterministic Turing machine N which accepts $TAUT$ and such that for every nondeterministic (deterministic) Turing machine N' which accepts (recognizes) $TAUT$ there exists a polynomial p such that for every $\alpha \in TAUT$*

$$TIME(N;\alpha) \leq p(|\alpha|, TIME(N';\alpha))$$

The systematic study of the efficiency of propositional proof systems (proof systems for $TAUT$) was started by S. Cook and R. Reckhow in [3]. They introduced the abstract notion of a proof system for $TAUT$ and proved that a polynomially bounded proof system for $TAUT$ exists if and only if **NP=co-NP** (see [3]).

Definition 3. *(see [3]) A proof system for $TAUT$ is a polynomial time computable function $h : \Sigma^\star \xrightarrow{onto} TAUT$.*

If $h(w) = x$ we say that w is a proof of x in h.

Definition 4. *(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 5. *(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.*

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 in [7].

Definition 6. *A proof system for $TAUT$ is optimal (p-optimal) if it simulates (p-simulates) any proof system for $TAUT$.*

4 Optimal acceptors and the structure of easy subsets of TAUT

It was shown in [5] and [4] that the problem of the existence of a complete language for **NP** ∩ **co-NP** and the problem of the existence of a complete language for **UP** can be related to statements about the existence of recursive enumerations of polynomial time clocked Turing machines covering all languages from these classes. It turns out that the problems of the existence of optimal acceptors for TAUT can be also related to analogous statements. In this chapter we survey known results in this direction.

Definition 7. *By an easy subset of $TAUT$ we mean a set A such that $A \subset TAUT$ and $A \in \mathbf{P}$.*

Definition 8. *By an **NP**-easy subset of $TAUT$ we mean a set A such that $A \subset TAUT$ and $A \in \mathbf{NP}$.*

Let D_1, D_2, D_3, ... denote the standard enumeration of all deterministic polynomial time clocked Turing machines.

Theorem 1. *(see [10]) Statements (i) – (iii) are equivalent:*

(i) There exists a p-optimal proof system for $TAUT$.
(ii) There exists a D-D-optimal acceptor for $TAUT$.
*(iii) The class of all easy subsets of $TAUT$ possesses a recursive **P**-presentation.*

By statement (iii) we mean: there exists a recursively enumerable list of deterministic polynomial time clocked Turing machines $D_{i_1}, D_{i_2}, D_{i_3}, \ldots$ such that:

(1) $L(D_{i_j}) \subset TAUT$ for every j;
(2) For every $A \subset TAUT$ such that $A \in \mathbf{P}$ there exists j such that $A = L(D_{i_j})$.

Let N_1, N_2, N_3, ... denote the standard enumeration of all nondeterministic polynomial time clocked Turing machines.

Theorem 2. *(see [10]) Statements (i) – (iii) are equivalent:*

(i) *There exists an optimal proof system for $TAUT$.*
(ii) *There exists an N-N-optimal acceptor for $TAUT$.*
(iii) *The class of all* **NP***-easy subsets of $TAUT$ possesses a recursive* **NP***–presentation.*

By statement (iii) we mean: there exists a recursively enumerable list of nondeterministic polynomial time clocked Turing machines $N_{i_1}, N_{i_2}, N_{i_3}, \ldots$ such that:

(1) $L(N_{i_j}) \subset TAUT$ for every j;
(2) For every $A \subset TAUT$ such that $A \in \mathbf{NP}$ there exists j such that $A = L(N_{i_j})$.

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

(i) *There exists an optimal propositional proof system.*
(ii) *There exists an N-D-optimal acceptor for $TAUT$.*
(iii) *The class of all easy subsets of $TAUT$ possesses a recursive* **NP***–presentation.*

By statement (iii) we mean: there exists a recursively enumerable list of nondeterministic polynomial time clocked Turing machines $N_{i_1}, N_{i_2}, N_{i_3}, \ldots$ such that:

(1) $L(N_{i_j}) \subset TAUT$ for every j;
(2) For every $A \subset TAUT$ such that $A \in \mathbf{P}$ there exists j such that $A = L(N_{i_j})$.

Proof. $(i) \rightarrow (ii)$

With every proof system for $TAUT$ we can associate a nondeterministic "guess and verify" algorithm for $TAUT$. On an input α this algorithm guesses a string w and then checks in polynomial time whether w is a proof of α. If successful, the algorithm halts in an accepting state. Also any deterministic algorithm M for $TAUT$ can be transformed to a proof system for $TAUT$. The proof of a formula α in this system is a computation of M accepting α.

Let *Opt* denote an optimal proof system for $TAUT$ and let N denote a nondeterministic Turing machine associated with *Opt* (a "guess and verify" algorithm associated with *Opt*). It can be easily checked that N accepts $TAUT$ and for any deterministic Turing machine M recognizing $TAUT$ there exists a polynomial p such that for every tautology α it holds:

$$TIME(N;\alpha) \leq p(|\alpha|, TIME(M;\alpha))$$

$(ii) \rightarrow (iii)$ and $(iii) \rightarrow (i)$

This follows by the same arguments as in the proofs of $(ii) \rightarrow (iii)$ and $(iii) \rightarrow (i)$ from Theorem 6.3. in [10].

5 Result

For a given propositional proof system h there might not be an algorithm that would produce an h-proof of a tautology α in time polynomial in the size of α. Considering these limitations of propositional proof systems, the following definition was proposed in [1] (see also [9]).

Definition 9. *A propositional proof system h is automatizable provided there is an algorithm M and a polynomial p such that whenever a tautology α has an h-proof of length n, the algorithm M produces on input α some h-proof of α in time bounded by $p(n)$.*

Let L be any language. We say that L has the property $\mathbf{Q}$ if the following statement is true:

($\mathbf{Q}$) The class of all **NP**-easy subsets of L possesses a recursive **P**-cover.

By statement ($\mathbf{Q}$) we mean: there exists a recursively enumerable list of deterministic polynomial time clocked Turing machines $D_{i_1}, D_{i_2}, D_{i_3}, \ldots$ such that

(1) $L(D_{i_j}) \subset L$ for every j;
(2) For every $A \subset L$ such that $A \in \mathbf{NP}$ there exists j such that $A \subset L(D_{i_j})$.

Definition 10. *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 $x \in \Sigma^\star$.*

If f is a function from Definition 10. we will say that f is a polynomial time many-one reduction from L_1 to L_2 (L_1 is polynomial time many-one reducible to L_2 via f.

Definition 11. *A function $h : \Sigma^\star \longrightarrow \Sigma^\star$ is said to be length increasing if for all $w \in \Sigma^\star$ it holds that $|h(w)| \geq |w|$.*

Theorem 4. *If $TAUT$ has the property $\mathbf{Q}$ and L is a language that is polynomial time many-one reducible to $TAUT$ via a length increasing reduction then L has the property $\mathbf{Q}$.*

Proof. Let $D_{i_1}, D_{i_2}, D_{i_3}, \ldots$ be a recursive **P**-cover of the class of all **NP**-easy subsets of $TAUT$. Let M be a polynomial time transducer computing the reduction from L to $TAUT$. We define the new recursively enumerable list of deterministic polynomial time clocked Turing machines $D'_{i_1}, D'_{i_2}, D'_{i_3}, \ldots$ The machine D'_{i_j} on input x runs D_{i_j} on $M(x)$. It follows from the definition of a polynomial time many-one reduction from L to $TAUT$ that $L(D'_{i_j}) \subset L$ for any j.

Let A be any fixed **NP**-easy subset of L. Since M computes a length increasing function, the set B = $\{y$: there exists $x \in A$ such that $M(x) = y$ $\}$ is a **NP**-easy subset of $TAUT$. It follows from the definition of a **P**-cover of the class of all **NP**-easy subsets of $TAUT$, that there exists k such that $B \subset L(D_{i_k})$ hence $A \subset L(D'_{i_k})$.

Let $TAUT^\star$ be the following language:
$TAUT^\star = \{\langle M, 0^k, w\rangle$: where $w \in TAUT$ (w is a code of a propositional tautology), M is a nondeterministic Turing machine and k is a natural number$\}$.

We say that a string v is in good form if $v = \langle M, 0^k, w\rangle$ where w is a code of a propositional formula, M is a nondeterministic Turing machine and k is a natural number. Let α_0 be a certain fixed propositional formula such that $\alpha_0 \notin TAUT$. We define $f : \Sigma^\star \longrightarrow \Sigma^\star$ in the following way: $f(v) = w$ if v is in good form ($v = \langle M, 0^k, w\rangle$), otherwise $f(v) = \alpha_0$. It is easy to see that $TAUT^\star$ is polynomial time many-one reducible to $TAUT$ via f.

Now we will show how to design a length increasing reduction from $TAUT^\star$ to $TAUT$.

For any formula α by a padded version of α we mean a formula $\tilde{\alpha} = \alpha \wedge \beta$, where β is a sufficiently long conjunction of trivial propositional tautologies. It is clear that there exists a polynomial time computable function
$pad : \Sigma^\star \times \{0\}^\star \longrightarrow \Sigma^\star$ with the following properties:

(1) $\alpha \in TAUT$ if and only if $pad(\alpha, 0^n) \in TAUT$ for any n natural;
(2) $|pad(\alpha, 0^n)| \geq n + |\alpha|$

Lemma 1. *$TAUT^\star$ is polynomial time many-one reducible to $TAUT$ via a length increasing reduction.*

Proof. The following function $\tilde{f} : \Sigma^\star \longrightarrow \Sigma^\star$, $\tilde{f}(w) = pad(f(w), 0^{|w|})$ is a polynomial time length increasing reduction from $TAUT^\star$ to $TAUT$.

Definition 12. *A nondeterministic Turing machine M is called sound if M accepts only propositional tautologies (if M accepts w, then $w \in TAUT$).*

To any nondeterministic Turing machine M we will assign the following language L_M consisting of tuples $\langle M, 0^k, w\rangle$, where $w \in \Sigma^\star$ and k is a natural number:

$$L_M = \{\langle M, 0^k, w\rangle : M \text{ accepts } w \text{ in } k \text{ steps } \}$$

Since $L_M \subset TAUT^\star$ if and only if M is sound, the language L_M can be used to verify for a given nondeterministic Turing machine M, that M is sound. As $L_M \in \mathbf{NP}$ for any nondeterministic Turing machine M, we have the following fact:

Fact 2. For every sound nondeterministic Turing machine M, the set L_M is a $\mathbf{NP}$-easy subset of $TAUT^\star$.

Now we are ready to prove the main result of this section.

Theorem 5. *Statements (i) - (iii) are equivalent:*

(i) There exists an optimal and automatizable proof system for TAUT.
(ii) There exists a D-N-optimal acceptor for $TAUT$.
(iii) The class of all $\mathbf{NP}$*-easy subsets of $TAUT$ possesses a recursive* $\mathbf{P}$*-cover.*

By statement (iii) we mean: there exists a recursively enumerable list of deterministic polynomial time clocked Turing machines $D_{i_1}, D_{i_2}, D_{i_3}, ...$ such that

(1) $L(D_{i_j}) \subset TAUT$ for every j;
(2) For every $A \subset TAUT$ such that $A \in \mathbf{NP}$ there exists j such that $A \subset L(D_{i_j})$.

Proof. *(i)* $\rightarrow$ *(iii)*

Let *Opt* be an optimal proof system for $TAUT$ which is automatizable using a deterministic Turing machine M. We define a recursively enumerable list of deterministic Turing machines F_1, F_2, F_3,... The machine F_k is obtained by attaching the shut-off clock $n^k + k$ to the machine M. On any input w, the machine F_k accepts w if and only if M on input w halts in an accepting state in no more than $n^k + k$ steps, where $n = |w|$, and F_k rejects w in the opposite case. The sequence F_1, F_2, F_3, F_4, ... of deterministic Turing machines possesses the properties (1) and (2):

(1) For every i it holds $L(F_i) \subset TAUT$;
(2) For every A which is an **NP**-easy subset of $TAUT$ there exists j such that $A \subset L(F_j)$.

To prove (2) let us consider A, an **NP**-easy subset of $TAUT$ accepted by a nondeterministic Turing machine N working in polynomial time. Combining this machine with the "brute force" algorithm for $TAUT$ we obtain the nondeterministic Turing machine N' accepting $TAUT$. From the definition of N' it follows that there exists a polynomial p such that $TIME(N'; \alpha) \leq p(|\alpha|)$ for any $\alpha \in A$.

Let us define a propositional proof system h such that if x codes a computation of N' which accepts α, then $h(x) = \alpha$. If x does not code an accepting computation of N', then $h(x) = \alpha_0$ for some fixed $\alpha_0 \in TAUT$. There exists a polynomial q such that for any $\alpha \in A$ there is a string w of length bounded by $q(|\alpha|)$ with $h(w) = \alpha$. This implies that there exists a polynomial r such that any $\alpha \in A$ possesses an *Opt*-proof w of length bounded by $r(|\alpha|)$. Since *Opt* is automatizable, there exists a polynomial s such that if $\alpha \in A$ then M produces an *Opt*-proof w of α in time bounded by $s(|\alpha|)$. From this we conclude that for a sufficiently large j, M produces an *Opt*-proof w of α in time bounded by $|\alpha|^j + j$ for any $\alpha \in A$. This gives $A \subset L(F_j)$.

(iii) $\rightarrow$ *(ii)*

It follows from Theorem 4. and Lemma 1. that the class of all **NP**-easy subsets of $TAUT^\star$ possesses a recursive **P**-cover. Let $D'_{i_1}, D'_{i_2}, D'_{i_3}, ...$ be a recursively enumerable list of polynomial time clocked deterministic Turing machines forming this cover. Let G be the machine generating the codes of the machines from this list.

Let M_1, M_2, M_3, ... be an enumeration of all nondeterministic Turing machines and let M_0 be the trivial "brute force" deterministic algorithm recognizing $TAUT$. The desired machine T, which is a D-N-optimal acceptor for $TAUT$, is constructed as follows:

On an input w, $|w| = n$, T spends its first n preliminary steps on computing, using the machine G, as many as possible codes of the machines from the sequence $D'_{i_1}, D'_{i_2}, D'_{i_3}, \ldots$ Let $D'_{i_1}, D'_{i_2}, D'_{i_3}, \ldots, D'_{i_m}$ be the result of these first n steps of T.

Then T simulates the work of M_1, M_2, M_3, ..., M_n and M_0 in several rounds. At the k-th round the machine T performs the following three groups of operations:

1. One additional computational step of M_0 on w.
2. One additional computational step of $D'_{i_1}, D'_{i_2}, D'_{i_3}, \ldots, D'_{i_m}$ on every input from the following list:
   $\langle M_1, 0^1, w\rangle$, $\langle M_2, 0^1, w\rangle$, ..., $\langle M_n, 0^1, w\rangle$,
   $\langle M_1, 0^2, w\rangle$, $\langle M_2, 0^2, w\rangle$, ..., $\langle M_n, 0^2, w\rangle$,
   $\langle M_1, 0^3, w\rangle$, $\langle M_2, 0^3, w\rangle$, ..., $\langle M_n, 0^3, w\rangle$,
   ...
   $\langle M_1, 0^k, w\rangle$, $\langle M_2, 0^k, w\rangle$, ..., $\langle M_n, 0^k, w\rangle$
3. T checks whether there are integers i, l ($i \le n$, $l \le k$) such that a certain machine D'_{i_j}, $1 \le j \le m$, has accepted $\langle M_i, 0^l, w\rangle$. If this is the case, T halts and accepts w, otherwise it continues operating till M_0 halts and delivers a YES or NO result.

First we shall show that T recognizes $TAUT$. If T finishes the simulation of M_0, then this is clear. So suppose that T accepts w because the situation in 3. occurs. Since the string $\langle M_i, 0^l, w\rangle$ has been accepted by a certain machine from the **P**-cover of the class of all **NP**-easy subsets of $TAUT^\star$, w is a propositional tautology.

Let N be any nondeterministic Turing machine accepting $TAUT$ ($N = M_i$ for a certain i). It remains to be proved that there exists a polynomial p such that $TIME(T; w) \le p(|w|, TIME(N; w))$ for any $w \in TAUT$. By Fact 2., as N is sound, the set L_N is a **NP**-easy subset of $TAUT^\star$ and hence there exists a certain machine D'_{i_j} from the **P**-cover such that $L_N \subset L(D'_{i_j})$. Let r be a polynomial bounding the working time of D'_{i_j}.

Let w be any input such that $|w| \ge \max\{i,\ i_j\}$ and $w \in TAUT$. Let $k = TIME(M_i, w)$. The machine T accepts w in the $k + r(n)$th round or sooner. Since the mth round of T takes polynomially many steps in m and n, ($n = |w|$), there is a polynomial p such that $TIME(T, w) \le p(n, k)$. This proves that T is a D-N-optimal acceptor for $TAUT$.

$(ii) \to (i)$

Let D be a D-N-optimal acceptor for $TAUT$. The existence of D implies by Theorem 2, as D is also an N-N-optimal acceptor for $TAUT$, the existence of an optimal proof system $L : \Sigma^\star \xrightarrow{onto} TAUT$.

Define $REF(L) = \{\langle \alpha, 0^n\rangle$: there exists an L-proof of α of length $\le n\}$ and $REF^*(L) = \{pad(\alpha, 0^n)$: there exists an L-proof of α of length $\le n\}$. As L is a proof system for $TAUT$ and due to property (1) of pad, $REF^*(L) \subseteq TAUT$. As L and pad are polynomial time computable and due to property (2) of pad, $REF^*(L) \in$ **NP**.

Let N be a nondeterministic Turing machine working in polynomial time and accepting $REF^{*}(L)$. Let q be a polynomial bounding its working time. Combining the machine N with the "brute force" algorithm for $TAUT$ we obtain the nondeterministic Turing machine $\tilde{N}$ recognizing $TAUT$ and such that for any $\beta \in REF^{*}(L)$ it holds: $TIME(\tilde{N};\beta) \leq q(|\beta|)$. Let p be a polynomial connected with the pair: the machine D and the machine $\tilde{N}$. Since $TIME(D;\beta) \leq p(|\beta|, TIME(\tilde{N};\beta)) \leq p(|\beta|, q(|\beta|))$ for any $\beta \in REF^{*}(L)$, there exists a polynomial r such that $TIME(D;\beta) \leq r(|\beta|)$ for any $\beta \in REF^{*}(L)$.

Let D_r be a deterministic Turing machine obtained by attaching the shut-off clock r to the machine D. We say that a string $w \in \Sigma^{\star}$ is in good form if $w = \langle \alpha, 0^n \rangle$, where α is a propositional formula and n is a natural number. Let α_0 be a certain fixed propositional tautology. We define $\bar{L} : \Sigma^{\star} \longrightarrow \Sigma^{\star}$ in the following way: $\bar{L}(w) = \alpha$ if w is in good form ($w = \langle \alpha, 0^n \rangle$) and D_r accepts $pad(\alpha, 0^n)$, otherwise $\bar{L}(w) = \alpha_0$. Clearly, $\bar{L}$ works in polynomial time and $\bar{L} : \Sigma^{\star} \longrightarrow TAUT$. Since $L : \Sigma^{\star} \overset{onto}{\longrightarrow} TAUT$, we have $\bar{L} : \Sigma^{\star} \overset{onto}{\longrightarrow} TAUT$. The system $\bar{L}$ p-simulates the system L via the function $v \mapsto \langle L(v), 0^{|v|} \rangle$, so $\bar{L}$ is an optimal proof system for $TAUT$.

Let M be a deterministic Turing machine which on input α runs in turn D_r on the following inputs: $pad(\alpha, 0), pad(\alpha, 0^2), pad(\alpha, 0^3), pad(\alpha, 0^4)$, ... If for a certain n the machine D_r accepts $pad(\alpha, 0^n)$ then the machine M produces the string $\langle \alpha, 0^n \rangle$ and halts in an accepting state. As pad and D_r work in polynomial time, M produces an $\bar{L}$-proof of a tautology α in time polynomial in the size of the smallest $\bar{L}$-proof of this tautology, hence the system $\bar{L}$ is automatizable.

The proof above gives more: namely, the second version of Theorem 5 is valid. In this version condition (i) is replaced by the following one:

(i') There exists a p-optimal and automatizable proof system for TAUT.

6 D-N-optimal acceptors and P-separability of disjoint NP-pairs

For a disjoint pair (A, B) of languages we say that (A, B) is **C**-separable if there exists a language S $\in$ **C** that separates (A, B), i. e. $A \subseteq S$ and $B \cap S = \emptyset$.

Definition 13. *(cf. [9]) We say that a propositional proof system L is weakly automatizable if there exists a propositional proof system $\bar{L}$ that p-simulates L and is automatizable.*

J. Köbler and J. Messner [6] proved that the question of whether every proof system for $TAUT$ admits an effective interpolation is related to the question of whether every disjoint pair of **NP**-sets is **P/poly** separable (see also [11]). We show that analogously: the question of whether every proof system for $TAUT$ is weakly automatizable is related to the question of whether every disjoint pair of **NP**-sets is **P**-separable.

Theorem 6. *Statements (i) - (iv) are equivalent:*

(i) Every disjoint pair of **NP** *sets is* **P** *separable.*
(ii) Every function in **NPSV** *has a total extension in* **FP**.
(iii) Every proof system for $TAUT$ is weakly automatizable.
(iv) For any **NP***-easy subset A of $TAUT$ there exists a deterministic polynomial time clocked Turing machine D such that $L(D) \subset TAUT$ and $A \subset L(D)$.*

Proof. $(i) \to (ii)$ Our proof technique is adapted from [11].

Let f be a function in **NPSV** computed by a suitable nondeterministic transducer in time bounded by a polynomial p. We define $f'(x)$ as the string $f'(x) = 0^{p(|x|)-|f(x)|}1f(x)$. In case $f(x)$ is not defined, neither $f'(x)$ is defined. By the above, for every x of a given length, $f'(x)$ has the same length.

Define for $a \in \{0, 1\}$ the two disjoint **NP** sets
$A_a = \{\langle x, i\rangle$: $1 \leq i \leq p(|x|) + 1$ and the i-th bit of f'(x) is $a\}$. Since A_0 and A_1 are **P**-separable there exists a deterministic transducer M working in polynomial time which outputs 0 on A_0 and 1 on A_1. Consider a deterministic Turing machine M' which on input x computes $a_i = M(\langle x, i\rangle)$ for $1 \leq i \leq p(|x|) + 1$, and then outputs the string w obtained by erasing the prefix 000...01 from the string $a_1 a_2 a_3 ... a_{p(|x|)+1}$. It follows that, in case $f(x)$ is defined the machine M' produces $f'(x)$, and then extracts from this the correct value for $f(x)$. In case $f(x)$ is not defined the machine M' computes some value for $\tilde{f}$, the total extension of f, at x.

$(ii) \to (iii)$

Let L be a proof system for $TAUT$. Consider the following canonical **NP**-pair for the system L (see [9]): $REF(L) = \{\langle \alpha, 0^n\rangle$: there exists an L-proof of α of length $\leq n\}$, $FALS^* = \{\langle \alpha, 0^n\rangle$ α is a falsifiable propositional formula, n is a natural number $\}$. Define the function $h : \Sigma^\star \longrightarrow \Sigma^\star$ as follows: $h(x) = 1$ for any $x \in REF(L)$, $h(x) = 0$ for any $x \in FALS^*$ and $h(x)$ is undefined for any other x. As $REF(L)$, $FALS^*$ are in **NP** and $REF(L) \cap FALS^* = \emptyset$, the function h is in **NPSV**.

Let g be a total extension of h in **FP** and let M be a deterministic Turing transducer working in polynomial time and computing the function g. We say that a string w is in good form if $w = \langle \alpha, 0^n\rangle$, where α is a propositional formula and n is a natural number. Let α_0 be a certain fixed propositional tautology. We define $\bar{L} : \Sigma^\star \longrightarrow \Sigma^\star$ in the following way: $\bar{L}(w) = \alpha$ if w is in good form and $M(w) = 1$, otherwise $\bar{L}(w) = \alpha_0$.

The proof system $\bar{L}$ p-simulates the system L via the function
$w \mapsto \langle L(w), 0^{|w|}\rangle$. Consider the following deterministic Turing transducer K. On input α the machine K runs in turn the machine M on the following inputs: $\langle \alpha, 0\rangle, \langle \alpha, 0^2\rangle, \langle \alpha, 0^3\rangle, \langle \alpha, 0^4\rangle$, ... If for a certain n it holds: $M(\langle \alpha, 0^n\rangle) = 1$, the machine K produces the string $\langle \alpha, 0^n\rangle$ and halts in an accepting state. Since $\bar{L}$ is automatizable using the machine K, the system L is weakly automatizable.

$(iii) \to (iv)$

Let $A \subset TAUT$ and $A \in$ **NP**. There exists a nondeterministic Turing machine N working in time bounded by a polynomial p and accepting A. Combining this machine with the "brute force" algorithm for $TAUT$ we obtain the nondeterministic Turing machine N' accepting $TAUT$.Let h be a proof system for $TAUT$ associated with N'. It follows from the definition of h that any $\alpha \in A$ has an h-proof of length bounded by $p(|\alpha|)$.

Let g be a proof system for $TAUT$ which p-simulates h and which is automatizable using the deterministic Turing transducer D. There exists a polynomial q such that any $\alpha \in A$ has a g-proof of length bounded by $q(|\alpha|)$ and, in consequence, there exists a polynomial r such that for any $\alpha \in A$ the machine D produces a g-proof of α in time bounded by $r(|\alpha|)$. Therefore for a sufficiently large j, D produces a g-proof of α in time bounded by $|\alpha|^j + j$.

Let D_j be a deterministic Turing machine obtained by attaching the shut-off clock $n^j + j$ to the transducer D. The machine D_j accepts a formula α if D on input α, halts in an accepting state after no more than $|\alpha|^j + j$ steps, and rejects α in the opposite case. It follows from the above that $L(D_j) \in \mathbf{P}$, $L(D_j) \subset TAUT$ and $A \subset L(D_j)$.

$(iv) \to (i)$

Let U, V be two disjoint **NP** sets. Take a polynomial time reduction g of U to the **NP**-complete set $\Sigma^\star\backslash$TAUT. Let $f : \Sigma^\star \longrightarrow \Sigma^\star$, $f(w) = pad(g(w), 0^{|w|})$ for any $w \in \Sigma^\star$. Since f is a length increasing reduction of U to $\Sigma^\star\backslash$TAUT we have: $f(U) \subset \Sigma^\star\backslash$TAUT, $f(U) \in \mathbf{NP}$, $f(V) \subset$ TAUT and $f(V) \in \mathbf{NP}$. It follows from (iv) that there exists a set $C \subset$ TAUT such that $C \in \mathbf{P}$ and $f(V) \subset C$.

Consider the set $W = f^{-1}(C) = \{w : f(w) \in C\}$. Clearly $W \in \mathbf{P}$ and $V \subset W$. In order to prove that $W \cap U = \emptyset$, suppose on the contrary that there exists x such that $x \in W$ and $x \in U$. Then by the definition of W, $f(x) \in C$ and, in consequence, $f(x) \in$TAUT. Conversely, since $x \in U$ and f reduces U to $\Sigma^\star\backslash$TAUT, we have $f(x) \in \Sigma^\star\backslash$TAUT, a contradiction. Thus $U \cap W = \emptyset$, which proves that the disjoint pair (U, V) is **P**-separable.

Clearly, statement (iii) from Theorem 5 (the class of all **NP**-easy subsets of TAUT possesses a recursive **P**-cover) implies (iv), so as a corollary we have obtained:

Corollary 1. *If there exists a D-N-optimal acceptor for TAUT then every disjoint pair of* **NP***-sets is* **P***-separable.*

References

1. Alekhnovich, M., Buss, S., Moran, S., Pitassi T.: Minimum propositional proof length is NP-hard to linearly approximate. In: Proc. 23rd Symposium Mathematical Foundations of Computer Science. Lecture Notes in Computer Science, Vol. 1450. Springer-Verlag, Berlin Heidelberg New York (1998) 176 – 184
2. Balcazar, J.L., Díaz, J., Gabarró, J.: Structural complexity I. 2nd edn. Springer-Verlag, Berlin Heidelberg New York (1995)
3. Cook, S.A., Reckhow R.A.: The relative efficiency of propositional proof systems. J. Symbolic Logic 44 (1979) 36–50
4. Hartmanis, J., Hemachandra, L.: Complexity classes without machines: On complete languages for UP. Theoret. Comput. Sci. 58 (1988) 129 – 142

5. Kowalczyk, W.: Some connections between presentability of complexity classes and the power of formal systems of reasoning. In: Proc. Mathematical Foundations of Computer Science. Lecture Notes in Computer Science, Vol. 176. Springer-Verlag, Berlin Heidelberg New York (1988) 364 – 369
6. Köbler, J., Messner, J.: Is the standard proof system for SAT p-optimal? In: Proc. 20th Annual Conference the Foundations of Software Technology and Theoretical Computer Science. Lecture Notes in Computer Science, Vol. 2136. Springer-Verlag, Berlin Heidelberg New York (2001) 361 – 372 Computational Complexity, (1998) 132–140
7. Krajíček, J., Pudlák, P.: Propositional proof systems, the consistency of first order theories and the complexity of computations. J. Symbolic Logic 54 (1989) 1063–1079
8. Messner,J.: On optimal algorithms and optimal proof systems. In: Proc. 16th Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science, Vol. 1563. Springer-Verlag, Berlin Heidelberg New York (1999) 541 –550
9. Pudlák, P.: On reducibility and symmetry of disjoint NP-pairs. In: Proc. 26th Symposium Mathematical Foundations of Computer Science. Lecture Notes in Computer Science, Vol. 2136. Springer-Verlag, Berlin Heidelberg New York (2001) 621 – 632
10. Sadowski, Z.: On an optimal propositional proof system and the structure of easy subsets of TAUT. Theoret. Comput. Sci. 288 (2002) 181 –193
11. Schöning, U.,Torán, J.: A note on the size of Craig's Interpolant. (1998) Unpublished manuscript