Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 21 (1998)

On the Existence of Optimal Propositional Proof Systems

and Oracle-Relativized Propositional Logic

Shai Ben-David * Anna Gringauze†

Abstract

We investigate sufficient conditions for the existence of optimal propositional proof systems. We concentrate on conditions of the form $CoN\mathcal{F} = N\mathcal{F}$. We introduce a purely combinatorial property of complexity classes - the notions of *slim* vs. *fat* classes. These notions partition the collection of all previously studied time-complexity classes into two complementary sets. We show that for every slim class an appropriate collapse entails the existence of an optimal propositional proof system. On the other hand, we introduce a notion of a propositional proof system *relative to an oracle* and show that for every fat class there exists an oracle relative to which such an entailment fails.

As the classes $\mathcal{P}$ (polynomial functions), $\mathcal{E}$ ($2^{O(n)}$ functions) and $\mathcal{EE}$ ($2^{O(2^n)}$ functions) are slim, this result includes all the previously known sufficiency conditions for the existence of optimal propositional proof systems.

On the other hand, the classes $\mathcal{NEXP}$, $\mathcal{QP}$ (the class of quasi-polynomial functions) and $\mathcal{EEE}$ ($2^{O(2^{2^n})}$ functions), as well as any other natural time-complexity class which is not covered by our sufficiency result, are fat classes.

As the proofs of all the known sufficiency conditions for the existence of optimal propositional proof systems carry over to the corresponding oracle-relativized notions, our oracle result shows that no extension of our sufficiency condition to non-slim classes can be obtained by the type of reasoning used so far in proofs on these issues.

*CS Dept., Technion, Haifa 32000, Israel, and SysEng Dept., RSISE, ANU, Canberra 0200, Australia, email: shai@syseng.anu.edu.au, shai@cs.technion.ac.il

†IBM Haifa Research Center, Haifa, Israel, email: vanna@vnet.ibm.com

ISSN 1433-8092

1 Introduction

The questions concerning the relationships between the classes $\mathcal{P}, \mathcal{NP}$ and $Co\mathcal{NP}$ are of the most important open problems in computational complexity. It turns out that the proof theory of propositional calculus provides a framework in which some of these questions can be reformulated and analyzed from a somewhat different perspective.

The first step in this direction is due to Cook and Reckhow [CR79] who have posed the following proof-theoretic question:

> "Does there exist a propositional proof system in which every tautology has a short proof (of size polynomial in the size of the tautology)?"

To formulate this question precisely Cook and Reckhow in [CR79] introduced a following abstract definition of a propositional proof system:

Definition 1 *A propositional proof system is a polynomial time computable function* $f : \{0,1\}^* \stackrel{\text{onto}}{\mapsto} Taut$.

A propositional proof system that allows short proofs to all tautologies is called a *super* propositional proof system. More formally,
Let the *size* of a formula or a proof be the total number of symbols in it.

Definition 2 *A propositional proof system* f *is called* **super** *if there exists a polynomial* $p : N \mapsto N$ *s.t. for every* $\alpha \in Taut$ *there exists* $w \in \{0,1\}^*$ *of size* $\leq p(|\alpha|)$ *s.t.* $f(w) = \alpha$.

In the same paper, Cook and Reckhow show that the existence of a super propositional proof system is equivalent to the assertion $\mathcal{NP} = Co\mathcal{NP}$.

A natural step in the analysis of such a problem is to introduce a relevant partial order, in our case, a partial order reflecting the relative strength of propositional proof systems.

Definition 3 *Let* f, g *be propositional proof systems.*

- f **simulates** g *if there exists a polynomial* p *s.t for every* $w \in \{0,1\}^*$, *there exists* $w' \in \{0,1\}^*$ *of length* $\leq p(|w|)$ *s.t.* $f(w') = g(w)$.
- *We say that* f **p-simulates** g *if such a* w' *can be found efficiently, that is, if there exists a poly-time computable function* $h : \{0,1\}^* \mapsto \{0,1\}^*$ *s.t. for all* w, $g(w) = f(h(w))$.

A "strongest" propositional proof system is called optimal.

- *A propositional proof system is called* **optimal** *if it simulates every other propositional proof system.*

- *A propositional proof system is called* **p-optimal** *if it p-simulates every other propositional proof system.*

Here again, the existence of a propositional proof system having such a property is unknown. Moreover, no natural open problems in computational complexity theory are known to be equivalent to the existence of such a propositional proof system. The strongest result on this topic is due to Köbler and Meßner [KM98]:

Theorem [Köbler and Meßner [KM98]]:

- *If* $TallyCo\mathcal{NEE} \subseteq \mathcal{NEE}$ *then there exists a optimal propositional proof system.*
- *If* $TallyCo\mathcal{NEE} \subseteq \mathcal{EE}$ *then there exists a p-optimal propositional proof system.*

This improves an earlier result of Pudlák [Pu84] and of Meßner and Torán [MT97] deriving the existence of an optimal propositional proof system from the assumption $\mathcal{NE} = Co\mathcal{NE}$, and $SparseCo\mathcal{NEE} \subseteq \mathcal{NEE}$, respectively.

Here $\mathcal{NE}$ and $\mathcal{NEE}$ are the complexity classes of all languages recognized by nondeterministic Turing machines running in $2^{O(n)}$ and $2^{O(2^n)}$ time respectively. (where n denotes the size of an input), and $\mathcal{EE}$ and $Co\mathcal{NEE}$ denote the corresponding deterministic time and co-nondeterministic time classes. *Tally* stands for the restriction of a class to its unary langages.

Krajíček and Pudlák, [KP89] present several proof theoretic statements which are equivalent to the existence of an optimal propositional proof system.

The only known necessary condition for existence of optimal propositional proof system (relating it to a natural computational complexity issue) was shown (implicitly) by Razborov [Ra94]:

Theorem [Razborov [Ra94]]: *If there exists a optimal propositional proof system, then there is a complete (under polynomial reduction) problem is* $Dis\mathcal{NP}$.

Where $Dis\mathcal{NP}$ is defined as follows:

Definition 4 *For every pair of languages* L_1, L_2, *such that* $L_1 \cap L_2 = \emptyset$, *define a problem* D_{L_1,L_2}: *given an input* x, *find an* $i \in \{1,2\}$ *s.t.* $x \notin L_i$.
$Dis\mathcal{NP} \stackrel{def}{=} \{D_{L_1,L_2} : L_1, L_2 \in \mathcal{NP}, L_1 \cap L_2 = \emptyset\}$.

It is an open question whether the existence of optimal propositional proof systems implies the existence of complete problems for either $\mathcal{NP} \cap Co\mathcal{NP}$ or $Dis\mathcal{NP}$ (or for any other complexity class for which the existence of a complete language is an open question).

In this paper we take one more step towards clarifying the relationship between the collapse of complexity classes and the existence of optimal propositional proof systems. We introduce a purely combinatorial property of complexity

classes - the notions of *slim* vs. *fat* classes. These notions partition the collection of all previously studied time-complexity classes into two complementary sets. We show that for every class in one of these sets an appropriate collapse entails the existence of an optimal propositional proof system, while, for every class in the other set there exists an oracle relative to which such an entailment fails. More precisely; On one hand we generalize the Köbler - Meßner sufficiency condition and show:

Theorem 1: *For every slim class $\mathcal{F}$, $TallyCoN\mathcal{F} \subseteq N\mathcal{F}$ implies the existence of optimal propositional proof systems, and $TallyCoN\mathcal{F} \subseteq D\mathcal{F}$ implies the existence of p-optimal propositional proof systems.*

($D\mathcal{F}$ stands for the class of languages computable in deterministic time belonging to $\mathcal{F}$, $N\mathcal{F}$ and $CoN\mathcal{F}$ stand for the corresponding non-deterministic and co-non-deterministic classes). As the classes $\mathcal{P}$ (polynomial functions), $\mathcal{E}$ ($2^{O(n)}$ functions) and $\mathcal{EE}$ ($2^{O(2^n)}$ functions) are slim, this result includes all the previously know sufficiency conditions for the existence of an optimal propositional proof system.

On the other hand, we introduce a notion of a propositional proof system *relative to an oracle*, and show:

Theorem 2: *For every* fat *class $\mathcal{F}$, there exists an oracle relative to which $CoN\mathcal{F} = D\mathcal{F}$ and yet there is no optimal propositional proof system.*

The classes $\mathcal{NEXP}$, $\mathcal{QP}$ (the class of quasi-polynomial functions) and $\mathcal{EEE}$ ($2^{O(2^{2^n})}$ functions),as well as any other natural time-complexity class which is not covered by Theorem 1, are fat classes.

As the proofs of all the known sufficiency conditions for the existence of optimal pps's (including Theorem 1 here) carry over to the corresponding oracle-relativized notions, this result shows that the reason that Theorem 1 does not extend to non-slim classes is deeper than just a technicality in the existing proof.

Of independent interest is the following result, showing that the common knowledge rule of thumb, stating that a collapse of low-level complexity classes can be shown, by padding tricks, to imply a collapse of any higher complexity class, is not always true.

Corollary 1: For every fat class of functions, $\mathcal{F}$, there exists an oracle relative to which $N\mathcal{F} = D\mathcal{F}$ and yet relative to the same oracle, for every slim class $\mathcal{F}'$, $TallyCoN\mathcal{F}' \not\subseteq \mathcal{NF}'$. (In particular, there is an oracle relative to which $\mathcal{NEEE} = \mathcal{DEEE}$ and yet there are Tally languages in $Co\mathcal{NEE}$ which are not in $\mathcal{NEE}$).

2 Definitions and Notation

We say that a function f is *super-polynomial* if $\forall k \forall^* n \, f(n) \geq n^k$ (where $\forall^* n$ means 'for all but finitely many n's'). We say that f is *non-polynomial* if $\forall k \neg(\forall^* n \, f(n) \leq n^k)$, that is, f exceeds every polynomial infinitely often.

Definition 5 *Let $\mathcal{F}$ be a class of functions (from the set of natural numbers to itself).*

- *$\mathcal{F}$ is* **slim** *if*

$$\exists f \in \mathcal{F} \ \forall g \in \mathcal{F} \ \exists c \ \forall^* n \ \ g(n+1) \leq (f(n))^c$$

- *For a function $H : \mathcal{N} \mapsto \mathcal{N}$, $\mathcal{F}$ is H-***fat** *if*

$$\forall f \in \mathcal{F} \ \exists g \in \mathcal{F} \ \forall^* n \ \ g(n+1) > H(f(n))$$

- *We say that $\mathcal{F}$ is* **fat** *if there exists some super-polynomial H such that $\mathcal{F}$ is H-fat.*

Note that the notions of slim and fat classes of functions are almost complementary; Every non-slim class is H-fat for some non-polynomial function H, and every class of functions used to bound some complexity measure in the definition of 'natural' complexity classes is either slim or fat.

Example 1:

- The classes $\mathcal{E} = \{f : \exists c \forall^* n f(n) \leq 2^{cn}\}$, $\mathcal{EE} = \{f : \exists c \forall^* n f(n) \leq 2^{c2^n}\}$ and $P = \{f : \exists c \forall^* n f(n) \leq n^c\}$ are slim classes.
- The classes $\mathcal{EXP} = \{f : \exists c \, s.t. \, \forall n \, f(n) \leq 2^{n^c}\}$, $\mathcal{QP} = \{f : \exists c \, s.t. \, \forall n \, f(n) \leq 2^{\log^c(n)}\}$, as well as $\mathcal{EEE} = \{f : \exists c \ \ s.t. \ \ \forall n \ \ f(n) \leq 2^{c2^{2^n}}\}$ are all fat classes.

Notation: For a class of functions $\mathcal{F}$,

- $D\mathcal{F}$ denotes the complexity class of all languages accepted by deterministic algorithms whose running time is a function in $\mathcal{F}$.
- $N\mathcal{F}$ denotes the complexity class of all languages accepted by non-deterministic algorithms whose running time is a function in $\mathcal{F}$.
- $CoN\mathcal{F}$ denotes the complexity class of all languages accepted by co-non-deterministic algorithms whose running time is a function in $\mathcal{F}$.
- A language $L \subseteq \{0,1\}^*$ is *Tally* if it is a subset of $\{0\}^*$. $TallyN\mathcal{F}$ is the class of all tally languages in $N\mathcal{F}$. We define the classes $TallyCoN\mathcal{F}$ and $TallyD\mathcal{F}$ similarly.

3 A General Sufficient Condition

The following theorem is a straightforward generalization of the Köbler and Meßner [KM98] sufficiency theorem.

Theorem 1 *Let $\mathcal{F}$ be a slim class of functions then*

1. *$TallyCo\mathcal{NF} \subseteq N\mathcal{F}$ implies the existence of an optimal pps.*
2. *$TallyCo\mathcal{NF} \subseteq D\mathcal{F}$ implies the existence of a p- optimal pps.*

Proof: We concentrate on the first claim of the theorem. The proof of the second claim is just a variation on this. We shall comment below on the changes needed to obtain that claim.

Let $(S_i : i \in N)$ be a recursive enumeration of all Turing machines mapping binary strings to binary strings and running in time $\leq n^2$ on input of size n. Assume further that, on input $(i, x, 0^n)$ the first k steps of the run of S_i on input x can be efficiently simulated. Say that a machine S is an **n-sound** pps if for every $\sigma \in \{0,1\}^{\leq n}$, $S(\sigma)$ is a propositional tautology.

Given a function $f : N \mapsto N$, define the language $Sound_f$ by

$$Sound_f = \{0^n : S_{i(n)} \text{ is } f(n)\text{-sound}\}$$

where $i(n)$ is $\max\{i : 2^i \textit{ divides } n\}$.
It is easy to see that given a class of functions $\mathcal{F}$, a function $f \in \mathcal{F}$ and an enumeration $(S_i : i \in N)$ as above, $Sound_f$ is in $TallyCo\mathcal{NF}$.

Assuming $TallyCo\mathcal{NF} \subseteq N\mathcal{F}$, there exists a polytime-computable relation R and a function $g \in \mathcal{F}$ so that

$$Sound_f = \{0^n : \exists y \leq g(n)\ \ R(0^n, y)\}$$

Let f be a function that demonstrates the slimness of $\mathcal{F}$, that is, for every $g \in \mathcal{F}$ there is some constant c such that for all n, $g(n+1) \leq f(n)^c$. Let us define a proof system S^* by having

$$S^*(0^s, w, y) = \begin{cases} S_i(w) & \text{if } |w| < s \text{ and, for some odd } k,\ s = f(k2^i),\ y \leq g(k2^i) \\ & \text{and } R(0^{k2^i}, y) \text{ holds.} \\ A \vee \neg A & \text{otherwise} \end{cases}$$

Let us show that S^* is indeed an optimal pps; Clearly S^* is a sound (that is, for every σ, $S^*(\sigma)$ is a tautology). Note also that, as $g(n) \leq f(n)^c$ for some constant c, S^* is efficiently computable.

Now, given any proof system S_i, for every string w, let k_w be the minimal odd k s.t. $|w| \leq f(k2^i)$, and let $y \leq g(k_w 2^i)$ be an R-witness to the $f(k_w 2^i)$-soundness of S_i. We readily get $S^*(0^{f(k_w 2^i)}, w, y) = S_i(w)$. All that is needed to

complete the proof is to show that, for some polynomial p_i, for every w, k_w satisfies $g(k_w 2^i) + f(k_w 2^i) + |w| \leq p(|w|)$. By the minimallity of k_w, $f((k_w - 1)2^i) < |w|$. Let c be such that for all n, $f(n+1) + g(n+1) \leq f(n)^c$. It follows that for all n and i, $g(n+2^i) + f(n+2^i) \leq f(n)^{c^{2^i}}$. By substituting $n = (k_w - 1)2^i$ we conclude that, $g(k_w 2^i) + f(k_w 2^i) \leq |w|^{c^{2^i}}$.

To prove part 2 of the theorem, just note that under the assumption of that claim $Sound_f \in D\mathcal{F}$ so one can repeat the above construction without the 'witness' strings y and get an effective construction of the S^* proof from the given S_i proof, w. □

4 Oracle-Relativized Propositional Calculus

We wish to have a class of languages that are parametrized by oracles $\{Taut^{\mathcal{A}}\}$, so that for every oracle $\mathcal{A}$, the language $Taut^{\mathcal{A}}$ is $CoN\mathcal{P}^{\mathcal{A}}$ -complete. Furthermore, we wish to define a notion of a relativized proof system, such that all the previous results concerning proof systems lift up to these relativized notions.

It turns out that this can be achieved by extending propositional calculus to allow an extra connective, a connective whose semantics depends upon the oracle relative to which our complexity notions are defined.

Definition 6 *The set of X-well-formed formulas ($X - WFFs$) are defined inductively:*

a) Every propositional variable is an $X - WFF$.

b) If α and β are $X - WFFs$, then so are $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$.

c) For $n \in \mathcal{N}$, if $\alpha_1, \alpha_2, \ldots, \alpha_n$ are $X - WFFs$, then so is $X(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Definition 7 *Let $\mathcal{A}$ be an oracle (that is, $\mathcal{A} \subseteq \{0,1\}^*$) , v - a truth assignment to propositional variables. The truth value $\bar{v}$ of $X - WFFs$ is defined inductively:*

a) If A is a propositional variable then $\bar{v}(A) = v(A)$.

b) If α, β are $X - WFFs$, $\circ \in \{\wedge, \vee, \rightarrow\}$ then $\bar{v}(\alpha \circ \beta) = TT_\circ(\bar{v}(\alpha), \bar{v}(\beta))$; $\bar{v}(\neg\alpha) = TT_\neg(\bar{v}(\alpha))$. (Where, $TT_\circ$ stands for the usual truth-table of the connective $\circ$).

c) Let us interpret truth values as binary bits (so True *becomes 1 and* False *becomes 0). For $n \in \mathcal{N}$, if $\alpha_1, \alpha_2, \ldots, \alpha_n$ are $X-WFFs$, then $\bar{v}(X(\alpha_1, \alpha_2, \ldots, \alpha_n)) =$ 1 iff $((\bar{v}(\alpha_1)), (\bar{v}(\alpha_2)), \ldots (\bar{v}(\alpha_n))) \in \mathcal{A}$.*

Definition 8 *Let $\mathcal{A}$ be an oracle. $SAT_{\mathcal{A}}$ is the set of all $\mathcal{A}$-satisfiable $X-WFFs$. $Taut_{\mathcal{A}}$ is the set of all $X - WFFs$ which are tautologies with respect to $\mathcal{A}$.*

Definition 9 *Let $\mathcal{A}$ be an oracle. A propositional proof system w.r.t. $\mathcal{A}$ is a function $f : \{0,1\}^* \stackrel{onto}{\rightarrow} Taut_{\mathcal{A}}$ such that $f \in FP^{\mathcal{A}}$ (the class of functions computable by polynomial Turing machines with oracle $\mathcal{A}$).*

Lemma 1 *For every oracle $\mathcal{A}$, $SAT_{\mathcal{A}}$ is $\mathcal{NP}^{\mathcal{A}}$-complete for polynomial time reductions.*

Proof: Slightly modified Cook's proof of "SAT is $\mathcal{NP}$-complete". □

Corollary 2: [*A Relativised Cook-Reckhow Theorem*] For every oracle $\mathcal{A}$, there exists an $\mathcal{A}$-relativized super propositional proof system iff $N\mathcal{P}^{\mathcal{A}} = CoN\mathcal{P}^{\mathcal{A}}$.

It is also straight-forward to check the the sufficiency results for the existence of optimal propositional proof systems (in particular, Theorem 1 above) and Razborov's necessary condition carry over to the relativized notions.

5 An Oracle-Relativized Insufficiency Condition

Definition 10 *A class $\mathcal{F}$ is* reasonable *if*

- *For every $f \in \mathcal{F}$ and every $k \in N$, the function $f(n)^k$ is also in $\mathcal{F}$.*
- *For every pair of functions, f, g, if $f \in \mathcal{F}$ and $\{n : f(n) \neq g(n)\}$ is finite, then $g \in \mathcal{F}$.*
- *For every $f, g \in \mathcal{F}$ the function* $\max(f,g)$, *defined by* $\max(f,g)(n) = \max\{f(i), g(i) : i \leq n\}$, *is also in $\mathcal{F}$.*

Theorem 2 *Let $\mathcal{F}$ be a reasonable class of functions. If $\mathcal{F}$ is fat then there exists an oracle $\mathcal{A}$ relative to which $CoN\mathcal{F} = D\mathcal{F}$ and yet there is no optimal propositional proof system.*

Proof: We shall construct the oracle $\mathcal{A}$ inductively. At each stage i we handle some task by adding a finite number of strings of length $\leq i$ to $\mathcal{A}$. Let $\mathcal{A}_i$ denote the set of strings defined to be in $\mathcal{A}$ in stages $\leq i$, $\mathcal{A}$ will be defined as $\cup_i \mathcal{A}_i$. The tasks are of two types; Making $CoN\mathcal{F}^{\mathcal{A}} = D\mathcal{F}^{\mathcal{A}}$ and making sure that no optimal $\mathcal{A}$-propositional proof system exists. There will be a function Pl that will map tasks to numbers, determining which task will be taken care of at which stage of the construction.

Tasks of the first type. Let $\{M_i : i \in N\}$ enumerate all nondeterministic oracle machines, and let L_i denote the $CoN\mathcal{F}$ language defined by M_i. So, for some $g_i \in \mathcal{F}$ and some poly-time computable relation R_i, $L_i = \{x : \forall y \leq g_i(|x|) \;\; R_i(x,y)\}$. Let $r_i \in N$ be such that n^{r_i} dominates the running time of R_i.

For every pair $(k,n) \in \mathcal{N} \times \mathcal{N}$, the task $T_{(k,n)}$ is to guarantee that, relative to constructed oracle, some $D\mathcal{F}$ machine for the language L_k computes the correct outputs for input strings of length n. We carry out this task by defining, at stage $Pl(k,n)$, the strings of length $Pl(k,n)$ in $\mathcal{A}$ to be exactly $\{0x0^{Pl(k,n)-(n+1)} : |x| = n \text{ and } x \in L_k^{\mathcal{A}_{i-1}}\}$.

Claim 1 If *for every k, the function $Pl(k,\ )$ is a member of $\mathcal{F}$, and for all but finitely many of the pairs $\{(k,n) : n \in \mathcal{N}\}$, the membership of strings of length n in $L_k^{\mathcal{A}}$ is determined by the membership of strings of length $< Pl(k,n)$ in $\mathcal{A}$,* then, *carrying out $T_{(k,n)}$ for all but finitely many n's guarantees that $L_k^{\mathcal{A}} \in D\mathcal{F}^{\mathcal{A}}$.*

Proof: Immediate from the construction. □

Tasks of the second type. Let $(S_i : i \in N)$ be a recursive enumeration of all Turing machines mapping binary strings to binary strings and running in time $\leq n^2$ on input of size n (that is, an enumeration of all candidate propositional proof systems). Furthermore, we require that for every i, $\{j : S_i = S_j\}$ is an infinite set computable in polynomial time. For each number t the task R_t is to guarantee that, relative to constructed oracle, $S_t^{\mathcal{A}}$ is not an $\mathcal{A}$-optimal propositional proof system.

For each $n \in \mathcal{N}$, let α_n denote $\neg X(True, x_1, \ldots, x_n)$. α_n is an $X - WFF$ and, for every oracle $\mathcal{A}$, it is an $\mathcal{A}$-tautology iff $\mathcal{A}$ contains no strings of length $n+1$ that have 1 as their first bit. We shall define below a super-polynomial function h, satisfying $\forall n (h(n))^2 < 2^n$, and a function l. At stage $Pl(t) = i$ we check whether, relative to $\mathcal{A}_i$, there exists some string w of length $\leq h(l(t))$ such that $S_t^{\mathcal{A}_{i-1}}(w) = \alpha_{l(t)}$. That is, whether relative to the oracle at this stage, S_t has a 'short' proof for $\alpha_{l(t)}$. If no such w exist we do nothing, that is, define $\mathcal{A}_i = \mathcal{A}_{i-1}$. However, if such a w exists, we add to $\mathcal{A}$ some string of the form $1y$ of length $l(t)$ that was not queried by $S_t^{\mathcal{A}_{i-1}}$ on its run on input w. As S_t runs in time $\leq n^2$, for $w \leq h(l(t))$ its running time is $\leq (h(l(t))^2$ which, by the choice of h, is less than $2^{l(t)}$, so necessarily such a string y exists.

Claim 2 *Let h be a super-polynomial function s.t. $h(n)^2 < H(n)$ for all n, and let $\mathcal{A}$ be constructed according to the above recipe. If for every t no strings of length $\leq (h(l(t))^2$ are added to $\mathcal{A}$ at stages later than stage $Pl(t)$, then there are no optimal $\mathcal{A}$-propositional proof system.*

Proof: As mentioned above, the output of each of the machines S_t on input strings of length $\leq h(l(t))$ is determined by $\mathcal{A}_{(h(l(t))^2}$. It follows that for w of length $\leq h(t)$, $S_t^{\mathcal{A}_{Pl(t)}}(w) = S_t^{\mathcal{A}}(w)$. Thus, for every propositional proof system S,

1. If there exists some t s.t. $S = S_t$ and there exists a short (that is, of length $\leq h(l(t))$) proof of $\alpha_{l(t)}$ in $S_t^{\mathcal{A}}$, then, by the construction, there exists a word y of length $l(t)$ s.t. $1y \in \mathcal{A}$. (Recall that $\alpha_{l(s)}$ is a tautology iff there is no word $1y$ in $\mathcal{A}$, where $|y| = l(s)$.) Thus, $\alpha_{l(t)}$ is not a tautology. Therefore, $S^{\mathcal{A}}$ is not sound.

2. Otherwise, for every k s.t. $S = S_k$ there exists no $S^{\mathcal{A}}$-proof of $\alpha_{l(k)}$ of size $\leq h(l(k))$. It follows that in $S^{\mathcal{A}}$ there are no polynomial proofs of the set of $\mathcal{A}$-tautologies

$$R_S = \{\alpha_{l(k)} : \quad S_k = S\}$$

$R_S \in P$, so there exists the propositional proof system U which has polynomial proofs for every $\alpha_{l(k)} \in R_S$. Thus $S^{\mathcal{A}}$ does not simulate U.

Therefore, $S^{\mathcal{A}}$ is not optimal. □

Lemma 2 *If*

1. *Pl is a one-to-one function mapping $(\mathcal{N} \times \mathcal{N}) \cup \mathcal{N}$ to $\mathcal{N}$, and it is increasing (that is, on the sub-domain $\mathcal{N} \times \mathcal{N}$ it is an increasing function whenever one coordinate is fixed, and it is also increasing on the sub-domain $\mathcal{N}$).*

2. *For all i, n, $Pl(i, n) > g_i(n)^{r_i}$.*

3. *h is super-polynomial and $h(n)^2 \leq H(n)$ for all n.*

4. *For all s, $l(s+1) > (h(l(s)))^2$.*

5. *For every i, $\{n : \exists s \ [g_i(n)^{r_i}, Pl(i, n)] \subseteq [l(s), h(l(s))^2]\}$ is finite.*

Then the oracle defined according to the above recipe witnesses the main theorem.

By Claims 1, 2 above, it suffices to show that for each of our tasks, once it is taken care of, no later addition to the oracle will interfere with it. We establish that by the following four simple claims.

Claim 3 *If $Pl(i, n) < Pl(r, s)$ then the machine M_i, while computing membership in L_i of strings of size n, cannot read the oracle values of strings of length $\geq Pl(r, s)$.*

Proof: On input of size n, M_i computes a poly-time relation R_i on strings of length $\leq g_i(n)$. The computation time of R_i on input of size m is bounded by m^{r_i}. So the requirement that $Pl(i, n) > g_i(n)^{r_i}$ takes care of the claim. □

Claim 4 *If $Pl(t) < Pl(s)$ then the machine S_t, while running on input of size $\leq h(l(t))$ (which bounds the length of the proofs of the tautology $\alpha_{l(t)}$ that we killed at stage $Pl(t)$) cannot read the oracle values for strings of length $\geq l(s)$ (which is where the task handled at stage $Pl(s)$ may write).*

Proof: As Pl is a monotone increasing function, $t \leq s-1$. Recalling that, for all n, $l(n+1) > h(l(n))^2$ and that $(h(l(t))^2$ is an upper bound on the range that S_t can access on $h(l(t))$-size proofs establishes the claim. □

Claim 5 *If $Pl(t) < Pl(i,n)$ then the machine S_t, while running on input of size $\leq h(l(t))$, cannot read the oracle values for strings of length $\geq Pl(i,n)$.*

Proof: On proofs of length $\leq h(l(t))$ the proof system S_t cannot read beyond length$Pl(t)$. □

Claim 6 *Under the assumption of Lemma 2, for every i, for all sufficiently large n and for all t, if $Pl(i,n) < Pl(t)$ then machine M_i, while running on input of size n, cannot read the oracle values of strings of length $\geq l(t)$ (which is where the task handles at stage $Pl(t)$ may write).*

Proof: Let n be large enough to make $\forall s\ [g_i(n)^{r_i}, Pl(k,n)] \not\subseteq [l(s), h(l(s))^2]$. For such n's, the assumption $Pl(i,n) < Pl(t)$ implies that $g_i(n)^{r_i} < l(t)$. and $g_i(n)^{r_i}$ is an upper bound on the length of strings that M_i can access on input of size $\leq n$. □

Now the proof of our theorem boils down to the following purely combinatorial problem of finding functions Pl, h and l so that the assumptions of Lemma 2 are met.

The existence of these functions follow easily from Lemma 3 and Lemma 5 of the next section.

□

6 The Combinatorics

Lemma 3 *Let $\mathcal{F}$ be an H-fat and reasonable class of functions. Let $\{g_i : i \in \mathcal{N}\} \subseteq \mathcal{F}$, and $\{r_i : i \in \mathcal{N}\}$ a sequence of natural numbers. Then there exists a sequence of functions $\{f_k : k \in N\} \subseteq \mathcal{F}$ s.t .:*

1. *Every function in the sequence is monotone increasing.*
2. *For every $n \in N$, the sequence $(f_k(n) : k \in N)$ is increasing.*
3. *The ranges of the functions f_k are mutually disjoint and are all contained in the set of odd numbers.*
4. *For every k, for all n, $f_k(n) \geq g_k(n)^{r_k}$.*
5. *$f_{k+1}(n+1) > H(f_k(n))$, for every $k, n \in \mathcal{N}$.*

Proof: Define the sequence $\{f_k : k \in \mathcal{N}\}$ by induction on k. At stage $k+1$, apply the H-fatness of $\mathcal{F}$ to find a function f' so that $f'(n+1) > H(f_k(n))$ (for all n), and then apply the 'reasonability' property to find an increasing f_{k+1} above f_k, f' and $g_{k+1}(n)^{r_{k+1}}$. □

We shall also need the following lemma, that may be of independent interest.

Lemma 4 *For every suer-polynomial functions H, there exists a super-polynomial function h such that, for every $k \in \mathcal{N}$, for all sufficiently large n $H(n) > h^{(k)}(n)$. (where $h^{(k)}$ is the k'th iterate of h, that is, $h^{(1)}(n) = h(n)$ and $h^{(i+1)}(n) = h(h^{(i)}(n))$).*

Proof: For every n, let

$$r_n \stackrel{\text{def}}{=} \max\{k : \forall m \geq n,\ m^{(k^k)} \leq H(m)\}$$

Note that, for all n, $r_{n+1} \geq r_n$, and, as H eventually exceeds every polynomial, r_n goes to ∞ as n does.

Define a function g by (strong) induction on n, as follows:

$$g(n) = \min\{g(\lfloor \log(n) \rfloor) + 1, r_n\}$$

It is easy to see that,

1. $\lim_{n\to\infty} g(n) = \infty$. ($g$ is the minimum of two functions, both growing to infinity).
2. For every polynomial p, $\forall^* n\ \ g(n^{p(g(n))}) \leq g(n) + 1$. (This follows from the requirement $g(n) \leq g(\lfloor \log(n) \rfloor) + 1$).

We now define h by

$$h(n) = n^{g(n)}$$

As g increases to infinity, h is super-polynomial.

Let us now show that,

$$\forall k\, \forall^* n\ \ 2k \leq r_n \Longrightarrow h^{(k)}(n) \leq H(n)$$

As r_n grows unboundedly with n, this will establish the lemma.

Claim 7 *For all sufficiently large n, $h^{(k)}(n) \leq n^{(g(n)^{2k})}$*

Proof: [*of the claim*] By induction on k,

$$h^{(k)}(n) = h(h^{(k-1)}(n))$$

By the induction hypothesis, for all sufficiently large n,

$$h(h^{(k-1)}(n)) \leq h(n^{(g(n)^{2(k-1)})})$$

$$h(n^{(g(n)^{2(k-1)})}) = (n^{(g(n)^{2(k-1)})})^{g(n^{(g(n)^{2(k-1)})})}$$

By property 2 of g, for all sufficiently large n, this is

$$\leq (n^{(g(n)^{2(k-1)})})^{g(n)+1} = n^{(g(n)^{2(k-1)})(g(n)+1)}$$

Which, for sufficiently large n's is $\leq n^{(g(n)^{2k})}$. □

Having proved the claim, the proof of the lemma is now immediate; As $g(n) \leq r_n$, if $2k \leq r_n$ we get, for sufficiently large n's,

$$h^{(k)}(n) \leq n^{r_n^{r_n}}$$

Which, by the definition of r_n is $\leq H(n)$. □

Finally, the following Definition and Lemma will supply all the remaining ingredients in the construction of our oracle.

Definition 11 *A* **sequence of large intervals** *is a sequence of pairs of numbers* $\{(a_i, b_i) : i \in \mathcal{N}\}$ *such that, for some super-polynomial function* F*, for all* i $F(a_i) < b_i < a_{i+1}$.

Lemma 5 *Let* $\mathcal{Q} = \{q_{i,j}; i, j \in \mathcal{N}\}$ *be an infinite matrix of numbers.*

If
1. *For every* i*, the sequence* $(q_{i,j} : j \in \mathcal{N})$ *is increasing.*
2. *For every* j*, the sequence* $(q_{i,j} : i \in \mathcal{N})$ *is increasing.*
3. *There exists a super-polynomial function* H *such that* $q_{i+1,j+1} > H(q_{i,j})$*, for all* i*,* j*.*

Then *there exists a sequence of large intervals,* $\{(a_i, b_i) : i \in \mathcal{N}\}$ *such that for every* k*,*

$$\{n : \exists i \ [q_{k,n}, q_{k+1,n}] \subseteq [a_i, b_i]\} \text{ is finite.}$$

Proof: Applying Lemma 4, let F be a super-polynomial function such that for every $k \in \mathcal{N}$, for all sufficiently large n, $H(n) > F^{(k)}(n)$.

We shall actually prove a slightly stronger result; For every k there exists some n_k s.t. for all $m \geq n_k$ the interval $[q_k(m), q_{k+1}(m+1)]$ contains an interval of the form $[n, F(n)]$ that contains no interval of the form $[q_i(r), q_{i+1}(r)]$ for any $i < k$.

Fix k and m. Let $I = [q_{i^*}(r^*), q_{i^*+1}(r^*+1)]$ be a containment- minimal interval in the (finite) set $\{(i, r) : [q_i(r), q_{i+1}(r+1)] \subseteq [q_k(m), q_{k+1}(m+1)]\}$. Note that for every i at most one interval of the form $[q_i(n), q_{i+1}(n)]$ is contained in I.

Otherwise, if both $[q_i(n), q_{i+1}(n)]$ and $[q_i(l), q_{i+1}(l+1)]$ are contained in I (for some $l > n$) then, as $q_{i+1}(n+1) \leq q_{i+1}(l+1)$, we'll have $[q_i(n), q_{i+1}(n+1)] \subseteq I$, contradicting the of I.

Consider the sequence of intervals

$$S = \Big([q_{i^*}(r^*), F(q_{i^*}(r^*))], \ldots\ldots [F^{(k)}(q_{i^*}(r^*)), F^{(k+1)}(q_{i^*}(r^*))]\Big)$$

There are k many disjoint (except for their endpoints) intervals in S. Recall that

$$q_{i^*+1}(r^*+1) > H(q_{i^*}(r^*)) > F^{(k+1)}(q_{i^*}(r^*))$$

(assuming r^* is sufficiently large). It follows that all of these k intervals are contained in I. As there are only $k-1$ many intervals in the set $\{[q_i(n), q_{i+1}(n)] \subseteq I : i < k\}$, one of the intervals in S contains no interval of the form $[q_i(r), q_{i+1}(r)]$ for any $i < k$. □

Completing the proof of Theorem 2 – The definitions of the functions Pl, h and l:

Apply Lemma 3 (for the given class $\mathcal{F}$ and function H) with the g_i's being the functions bounding the witness-length in the $CoN\mathcal{F}$ languages L_i and the r_i's being the exponents in the time bound for the computation of the relations R_i in the definitions of these languages (as detailed in the second paragraph of the Proof of Thm. 2). Let $\{f_k : k \in \mathcal{N}\}$ be the sequence of functions whose existence is guaranteed by the lemma.

Apply Lemma 5 to the matrix defined by $q_{k,n} = f_k(n)$, and let F be the 'largeness' function of the sequence of large intervals whose existence is concluded.

- We shall define Pl separately for the domain $\mathcal{N} \times \mathcal{N}$ (for tasks of the first type, indexed by a pair (k, n)) and for the domain $\mathcal{N}$ (for tasks of the second type, indexed by t).
  - For the sub-domain $\mathcal{N} \times \mathcal{N}$, define $Pl : \mathcal{N} \times \mathcal{N} \mapsto \mathcal{N}$, define $Pl(k.n) = f_{k+1}(n)$.
  - Let us define the missing part of Pl by $Pl(s) = F(l(s))$,
- Let l be any increasing function whose range is contained in $\{a_i : i \in \mathcal{N}\}$ where the a_i's are the left-most points in the intervals (a_i, b_i) whose existence is concluded by Lemma 5.
- Finally, define h by $h(n) = \min\{\sqrt{F(n)}, \sqrt{H(n)}\}$.

It is straightforward to verify that these functions, Pl, h and l satisfy the requirements of Lemma 2. This completes the proof of Theorem 2.

Acknowledgements

We wish to thanks Jochen Meßner for suggesting the use of techniques from [KM98] to obtain an improvement in our Theorem 1.

References

[CR79] Cook, S., A., and Reckhow, R., A., "The relative efficiency of Propositional Proof Systems" *Journal of Symbolic Logic* 44: 36-50, 1979.

[KM98] Köbler J., and Meßner J., "Complete Problems for Promise Classes by Optimal Proof Systems for Test Sets" To appear in Proceedings of the Computational Complexity Conference, 1998.

[KP89] Krajíček, J., and Pudlák, P., "Propositional Proof Systems, the Consistency of First Order Theories and the Complexity of Computation" *Journal of Symbolic Logic*, 54(3): 1063-1079, 1989.

[MT97] Meßner, J., and Torán, J., "Optimal Proof Systems for Propositional Logic and Complete Sets" *ECCC report series* Technical Report, June 1997.

[Pu84] Pudlák, P., "On the Length of Proofs of Finitistic Consistency Statements in First Order Logic" *Logic Colloquium '84*, pp 165-195.

[Ra94] Razborov, A., A., "On Provably Disjoint NP-pairs. Technical Report, *BRICS* november, 1994.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'