# A note on the size of Craig Interpolants *

Uwe Schöning and Jacobo Torán
Abteilung Theoretische Informatik
Universität Ulm
Oberer Eselsberg
D-89069 Ulm, Germany

January 23, 2007

**Abstract**

Mundici considered the question of whether the interpolant of two propositional formulas of the form $F \rightarrow G$ can always have a short circuit description, and showed that if this is the case then every problem in NP $\cap$ co-NP would have polynomial size circuits. In this note we observe further consequences of the interpolant having short circuit descriptions, namely that UP $\subseteq$ P/poly, and that every single valued NP function has a total extension in FP/poly. We also relate this question with other Complexity Theory assumptions.

## 1 Introduction

Craig's Interpolation Theorem for propositional Logic [6] states that for every pair of propositional formulas $F, G$ such that $F \rightarrow G$ is a tautology, there is a formula $H$ that uses only the common variables from $F$ and $G$ satisfying $F \rightarrow H$ and $H \rightarrow G$. Formula $H$ is called an interpolant for $F$ and $G$.

The estimatation of the size of $H$ with respect to the sizes of $F$ and $G$, is an interesting open question connected with the relationship between uniform and non-uniform complexity classes. Mundici considered this question in [14, 15] and tried to estimate the size of a circuit description for an interpolant of two formulas. He gave in [15] a lower bound for the depth of such an interpolant circuit showing that there are infinitely many formulas $F \rightarrow G$ for which a circuit description $C_H$ of any interpolant $H$ must satisfy

$$depth(C_H) > d + \frac{1}{3}\log(\frac{d}{2}),$$

where $d$ is the maximum of the circuit depths of $G$ and $H$.

*Partially supported by DAAD and Spanish Government (Acción Integrada 131–B, 313–AI–e–es/zk).

Recently, several results bounding the size of the interpolants for a pair of formulas $F, G$ by the size of a formal proof of $F \to G$ in certain proof systems like Resolution or Cutting Planes [13, 16, 4] have attracted again interest in this question.

For a given pair of formulas satisfying $F \to G$, and with the set $x = \{x_1, \ldots, x_m\}$ of common variables, let us denote by $\mathit{int}(F, G)$ the size of the smallest Boolean circuit $C$ computing an interpolant for the formulas (that is, $C$ has as input variables the set $x$ of common variables for $F$ and $G$, and $C$ computes an interpolation function for $F \to G$). The function $\delta$ measuring the largest size of a circuit computing an interpolant for formulas of a certain length is defined as

$$\delta(n) = \max\{\mathit{int}(F, G) \mid |F| = |G| = n\}.$$

It is not hard to see that if every problem in NP has polynomial size circuits (NP $\subseteq$ P/poly), then there is a polynomial $p$ bounding the size of the function $\delta$. Superpolynomial (and even moderate polynomial) lower bounds for this function are therefore very hard to prove. A different approach in order to give evidence that $\delta$ is not polynomially bounded was taken by Mundici [14]. He considered the following hypothesis:

**Hypothesis 1 (H1)** *There is a polynomial $p$ such that for all $n$, $\delta(n) \leq p(n)$.*

Mundici proceeded to show that H1 implies unexpected results. He proved that if H1 holds, then the class NP $\cap$ co-NP has polynomial size circuits.

We obtain in this note further consequences of the assumption of this hypothesis, namely that the class UP of problems in NP accepted by nondeterministic Turing machines with unique accepting paths [18] has polynomial size circuits, and that every single valued NP function has a total extension in FP/poly.

Observe that H1 assumes the existence of a single polynomial that bounds the size of the interpolant for every possible pair of formulas. In order to deal with uniform complexity classes and relate the size of the interpolants with other complexity theoretical assumptions, we consider uniform families of (pairs of) Boolean formulas and assume the existence of a polynomial bounding the size of the interpolant for each family (instead of a single polynomial bounding all the interpolants, as it is done in H1). We consider the second hypothesis:

**Hypothesis 2 (H2)** *For every polynomial time uniform family of pairs of formulas $\mathcal{F} = \{F_n, G_n\}$ such that for every $n$, $F_n$ and $G_n$ have $n$ common variables and $F_n \to G_n$ is a tautology, there is a polynomial $p$ and a family $\mathcal{H} = \{H_n\}$ of circuits such that for every $n$ $H_n$ is an interpolation circuit for $F_n \to G_n$, with the set of common variables of $F_n$ and $G_n$ as input gates, and with size $|H_n| \leq p(n)$.*

Observe that it is possible to define different "uniform versions" of Mundici's hypothesis H1. The version considered [7] not only assumes that the interpolants have polynomial size, but also that they are polynomial time computable. An alternative hypothesis whose strength would lie between H1 and H2 can be defined by avoiding the requirement in H2 that forces the families of formulas to be polynomial time computable. We have choosen H2 as the most adequate hypothesis since it the weakest hypothesis and relates better to

other complexity assumptions. The corresponding results to the ones presented here can be adapted to the other possible assumptions.

Clearly H1, implies H2, and the known consequences that follow from H1, follow already from H2. In fact, as we show in Section 3, H2 is equivalent to the following two statements:

- for every pair of disjoint NP sets, there exist a family of polynomial size circuits separating them,
- every single valued NP function can be extended to a total function in FP/poly.

Dahlhaus, Israeli and Makowski considered in [7] the interpolation problem for quantified Boolean formulas. Every pair of quantified Boolean formulas $F, G$ satisfying $F \rightarrow G$ has a quantifier free formula as interpolant, and one can also consider the size of its smallest circuit description. It is not hard to see that for the case of *existential* Boolean formulas $F$ and $\neg G$, satisfying $F \rightarrow G$, one can easily find a pair of quantifier-free formulas $F', G'$ (namely the same formulas without the quantifiers) satisfying $F' \rightarrow G'$ and whose interpolants are also interpolants for $F \rightarrow G$. Because of this, H2 remains the same if we consider existential Boolean formulas instead of quantifier-free formulas.

On the other hand, we show in Section 3 that H2 for *universal* formulas, that is, the question of whether interpolants for such formulas have polynomial size, becomes equivalent (in the non-uniform case) to the recently considered assumption $Q'$ [8]. $Q'$ is the question of whether a single bit of an inverse of a polynomial time computable and onto function can be computed in polynomial time. This question has been characterized in a variety of ways [8], and is closely related to the Tautology Search Problem [10]. Assumption $Q'$ is therefore, in a sense, dual to H2 since $Q'$ is equivalent to the statement that every pair of disjoint co-NP sets is separable, while, as shown in Section 2, H2 is equivalent to every pair of disjoint NP sets being separable.

In this note we use several well known complexity theory notions, like polynomial size circuits, uniformity, or different complexity classes. For formal definitions we refer the reader to the standard text books in the area.

# 2 Polynomial size interpolants imply UP $\subseteq$ P/poly

Mundici proved in [14] that if H1 holds then the class NP $\cap$ co-NP has polynomial size circuits. We show here that the same holds for UP, the class of NP problems accepted by nondeterministic Turing machines with at most one computation path on every input. For the proof we make use of the Cook formulas [5] for the languages in NP, in a similar way as it was done in Mundici's result.

**Theorem 2.1** *If H1 holds then UP $\subseteq$ P/poly.*

**Proof.** Let $p$ be a polynomial and let $A$ be a problem in UP computed by a nondeterministic machine $M$ in time $p$. For strings $x$ of length $n$ consider the Cook formula $F_{n,0}(x, y)$ expressing that $y$ is an accepting path for $x$ and its first bit is a 0. Define $F_{n,1}(x, y)$ annalogously. Since for every input there is at most one accepting path, we have that

$$F_{n,1}(x, y) \rightarrow \neg F_{n,0}(x, z)$$

is a tautology. By the hypothesis there is a small circuit description $C_n$ for an interpolant of the formulas. In case $x \in A$, $C_n(x)$ outputs the first bit of the unique accepting path of $x$.

In the same way, we can define a pair of Boolean formulas $F^i_{n,1}(x,y)$, $F^i_{n,0}(x,y)$, for every position $i$ in an accepting path for $x$, $1 \leq i \leq p(n)$. By H1 there are circuit descriptions $C^i_n$ of polynomial size computing the $i-th$ bit of the unique accepting path. The circuits for all the positions $i$ can be combined into a single circuit of polynomial size in $n$. If $x \in A$ this circuit computes the accepting path of $x$.

The final circuit accepting $A$ needs one more modification. It works as follows: on input $x$, it uses the interpolation circuits to compute a string $y$ and outputs a 1 if and only if $F_n(x,y)$ holds ($y$ is an accepting path for $x$). This last check is necessary in order to detect strings that do not belong to $A$, since in this case we do not have any control over the values produced by the interpolants. □

It is an open problem whether it also follows from H1 that the (probably) larger complexity class FewP [1, 12] has polynomial size circuits.

Later in this section, we show that H1 implies that every pair of disjoint NP sets can be separated by polynomial size circuits. This, together with the result of [9] showing that if disjoint NP sets have separator circuits, then UP $\subseteq$ P/poly, constitutes an alternative proof for the previous result.

In order to be able to compare the question of whether interpolants of propositional Boolean formulas have always short circuit description, with other standard (uniform) complexity theory assumptions, we condidered in the Introduction a uniform version of Mundici's hypothesis, H2. We show next that H2 can be exactly characterized in terms of separator circuits for NP problems, and single valued NP functions. Let us define these concepts formally.

**Definition 2.2** *Let $A, B \subseteq \Sigma^*$ be two sets. We say that $A$ and $B$ have polynomial size separator circuits if there is a polynomial $p$ and a family of Boolean circuits $\mathcal{C}$ such that for every $n \in \mathbb{N}$, and for every $x \in \Sigma^n$, if $x \in A$ then $C_n(x) = 1$ and if $x \in B$ then $C_n(x) = 0$.*

Obviously, only disjoint sets $A$ and $B$ can have separator circuits.

**Definition 2.3** *The class of single valued NP functions, NPSV, is formed by the functions $f : \Sigma^* \longrightarrow \Sigma^*$ (not necessarily total), for which there is a nondeterministic Turing machine with output, $M$, such that for every $x$, $M(x)$ has accepting computations (with output) iff $f(x)$ is defined, and in case that $f(x)$ is defined, all the accepting computations produce the same output.*

The assumption H2 can now be characterized in different ways.

**Theorem 2.4** *The following statements are equivalent:*

1. *H2.*
2. *Every disjoint pair of NP sets has polynomial size separator circuits.*

*3. Every function in NPSV has a total extension in FP/poly.*

**Proof.** $1 \Rightarrow 2$: This proof is again based on Mundici's idea of considering the Cook's formulas for NP problems. Let $A_0$ and $A_1$ be two disjoint NP sets accepted by the nondeterministic polynomial time machines $M_0$ and $M_1$. For every $n \in \mathbf{N}$, let $F_{n,a}(x,y)$ (for $a \in \{0,1\}$) be the propositional formula expressing that $y$ is an accepting path for the machine $M_a$ on input $x$, ($|x| = n$). Since $M_0$ and $M_1$ accept disjoint sets, $F_{n,1}(x,y) \rightarrow \neg F_{n,0}(x,z)$ holds. Consider the families of pairs of formulas $\{\{F_{n,1}, F_{n,0}\}$. By the assumption, these formulas have interpolation circuits $C_n$ of polynomial size. The circuits are separators for $A_0$ and $A_1$ since for every $n \in \mathbf{N}$, and for every $x \in \Sigma^n$, if $x \in A_1$ then $C_n(x) = 1$ and if $x \in A_0$ then $C_n(x) = 0$.

$2 \Rightarrow 3$: Let $f$ be a function in NPSV, computed by a nondeterministic machine $M$ being time bounded by polynomial $p$. In order to have the same length for the value of $f$ for every $x$ of a given length, define $f'(x)$ as the string $f'(x) = 0^{p(|x|)-|f(x)|}1f(x)$ in case $f(x)$ is defined. In case $f(x)$ is not defined, then neither $f'(x)$ is defined. Clearly a polynomial size circuit can extract from $f'$ the value of $f$. Define for $a \in \{0,1\}$ the two disjoint NP sets

$$A_a = \{\langle x, i\rangle \mid 1 \leq i \leq p(|x|) + 1, \text{ and the } i-\text{th bit of } f'(x) \text{ is } a\}.$$

In case $f(x)$ is defined, a separator circuit $C_n$ for $A_1$ and $A_0$, on input $\langle x, i\rangle$ computes the $i$-th bit of $f'(x)$. We can use this separator circuit to construct another one that on input $x$ computes in parallel the values $C_n(\langle x, i\rangle)$ for all the possible values of $i$, obtaining $f'(x)$, and then extracts from this the correct value for $f(x)$. The new circuit still has polynomial size. In case $f(x)$ is not defined, the circuit computes some value (depending on the separator) for the total extension of $f$.

$3 \Rightarrow 1$: Let $\mathcal{F} = \{F_n, G_n\}$ be a polynomial size families of formulas satisfying that for every $n$, $F_n(x,y) \rightarrow G_n(x,z)$ is a tautology and $F_n$ and $G_n$ have the common set of $n$ variables $x$. Define the function $f$ as:

$$f(x) = \begin{cases} 1 & \text{if for some } y,\ F(x,y) \text{ holds} \\ 0 & \text{if for some } z,\ \neg G(x,z) \text{ holds} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Observe that $f$ is well defined by the logical relation between $F$ and $G$, it can only take one value. Also, since the families of formulas are polynomial time uniform, strings $y$ and $z$ have polynomial length with respect to the length of $x$. Because of this, $f \in$ NPSV. A total extension for $f$ (taking only values 0 and 1) is an interpolant for the families of formulas; because if $F_n(x,y)$ holds, then $f(x) = 1$, and if $f(x) = 1$, then $G_n(x,z)$ holds. □

Notice that the known conclusion, NP ∩ co-NP ⊆ P/poly and UP ⊆ P/poly, following from Mundici's assumption that there is a polynomial bounding the size of the interpolants of all the propositional formulas, can already be derived from H2.

# 3 Interpolants and the hypothesis $Q'$

Dahlhaus et al. consider in [7] the interpolation problem for quantified Boolean formulas. Following their notation, let us define $\Sigma(0) = \Pi(0)$ as the class of quantifier free Boolean

formulas. For $k \geq 1$ let $\Sigma(k)$ be the set of all the formulas of the form $\exists x_1 \exists x_2 \ldots \exists x_m F$ with $F \in \Pi(k-1)$, and let $\Pi(k)$ be the class of formulas of the form $\forall x_1 \forall x_2 \ldots \forall x_m F$ with $F \in \Sigma(k-1)$. For every pair of quantified Boolean formulas $F, G$ satisfying $F \rightarrow G$ with set $x$ of common free variables, there is a quantifier free formula $H$ that is an interpolant for $F$ and $G$ and whose variables are those in $x$.

Dahlhaus et al. consider bounds on the size of the interpolants for certain classes of quantified formulas.

**Hypothesis 3 ($\mathbf{H}_{\Sigma(k)}$)** *For every polynomial size family $\mathcal{F}$ of pairs of formulas $\{F_n, G_n\}$ such that for every $n$, $F_n, \neg G_n \in \Sigma(k)$, $F_n$ and $G_n$ have $n$ common variables and $F_n \rightarrow G_n$ is a tautology, there is a polynomial $p$ and family $\mathcal{H}$ of circuits such that for every $n$ $H_n$ is an interpolation circuit for $F_n \rightarrow G_n$, with the set of common variables of $F_n$ and $G_n$ as input gates, and with size $|H_n| \leq p(n)$.*

From the results in [7] follows that for $k \geq 2$ $\mathrm{H}_{\Sigma(k)}$ is equivalent to $\mathrm{H}_{\Pi(k)}$ and to NP having polynomial size circuits. It follows also that the hypothesis for quantifier free formulas, H2, is equivalent to the hypothesis for existential formulas, $\mathrm{H}_{\Sigma(1)}$. We consider here the only missing case, the interpolation of universal formulas, and show that the hypothesis $\mathrm{H}_{\Pi(1)}$ is equivalent (in the non-uniform case) to the following complexity assumption, denoted $Q'$ in [8]:

> $Q'$: For all polynomial time computable onto functions $f$, there exists a polynomial time computable function $g$ that computes the first bit of $f^{-1}$.

Observe that $Q'$ is a uniform assumption while $\mathrm{H}_{\Pi(1)}$ is non-uniform. In order to compare both hypothesis, we consider the non-uniform version of $Q'$, that is, we allow function $g$ to be computable by a family of polynomial size circuits $\mathcal{G}$. We will denote the non-uniform version of $Q'$ by $nQ'$.

**Theorem 3.1** $\mathrm{H}_{\Pi(1)}$ *is equivalent to $nQ'$.*

**Proof.** In [8] it is proved that $nQ'$ is true if and only if for every polynomial time nondeterministic Turing machine $M$ accepting $\Sigma^*$, there is a family $\mathcal{G}$ of polynomial size circuits, that for every $x$, $G_{|x|}(x)$ computes the first bit of an accepting computation of $M(x)$. We show that $\mathrm{H}_{\Pi(1)}$ holds iff this statement is true.

From left to right, let $M$ be a nondeterministic polynomial time Turing machine accepting $\Sigma^*$. By Cook's Theorem, we can construct two families of formulas $\mathcal{F}_0$, $\mathcal{F}_1$ such that for a given length of $x$, $n$,($a \in \{0,1\}$), $F_{n,a}(x,y)$ is true iff

$$y \text{ is a rejecting path for } x \text{ or } y \text{ starts with bit } a.$$

Consider the universal formulas $\forall y F_{n,a}(x,y)$ (for $a \in \{0,1\}$). Since $M$ accepts $\Sigma^*$, the expression $\forall y F_{n,1}(x,y) \rightarrow \neg \forall z F_{n,0}(x,z)$ is a tautology (this expression means that there is some accepting path for $x$ starting by 1 or there is an accepting path starting by 0). By the hypothesis there is a circuit description $C_n$ of an interpolant for these formulas, with polynomial size with respect to $n$. It is easy to check that if $C_n(x)$ outputs bit $a$ then $x$ has an accepting path starting with $a$.

For the other direction, let $\mathcal{F} = \{F_n(x,y), G_n(x,z)\}$ be a family of formulas such that for every $n$ the expression $\forall y F_n(x,y) \rightarrow \exists z G_n(x,z)$ is a tautology. Consider the following nondeterministic Turing machine $M$:

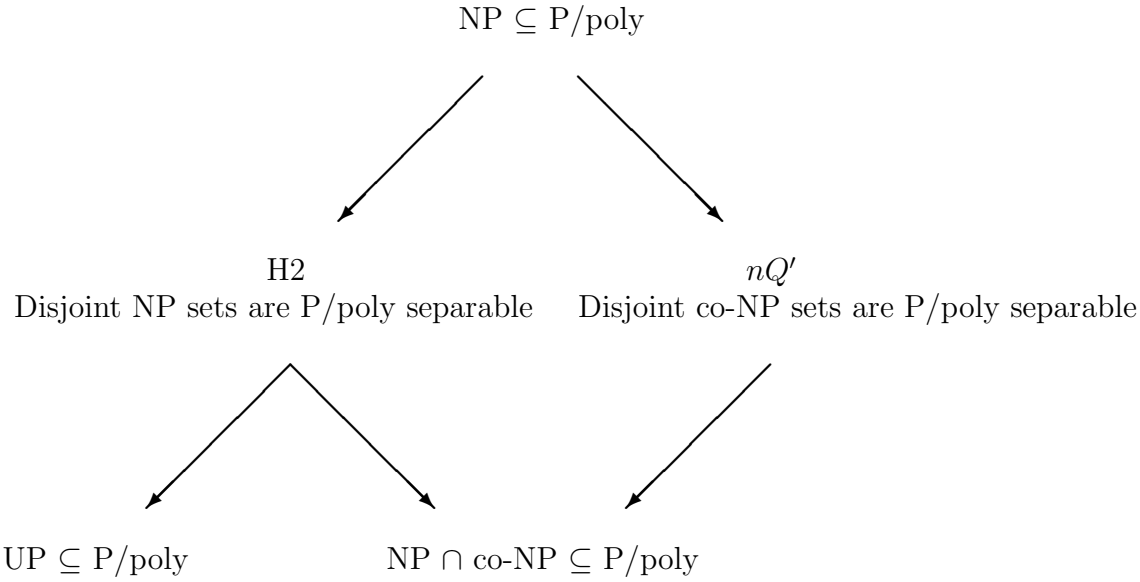
**input** $x$;
**guess** $a \in \{0,1\}$;
**if** $a = 0$ **then guess** $y$
  **if** $\neg F(x,y)$ **then accept**;
**else guess** $z$
  **if** $G_n(x,z)$ **then accept**
**end.**

$M$ accepts $\Sigma^*$ since for every $x$ there are strings $y, z$ such that $\neg F_n(x,y)$ or $G_n(x,z)$ holds. A circuit computing the first bit of an accepting path for $M$ on an input $x$ is an interpolant for the formulas. Observe that if $\forall y F_n(x,y)$ holds, then there are only accepting paths starting by a 1, and some accepting path starts by a 1, then $\exists z G_n(x,z)$ holds.

□

Fenner et al. give in [8] several characterizations of $Q'$. Particularly interesting is that $Q'$ is equivalent to every pair of disjoint co-NP sets being polynomial time separable. Recall from Theorem 2.4 that H2 is equivalent to every pair of disjoint NP sets having separator circuits of polynomial size. This shows that H2 and $Q'$ are dual assumptions.

The diagram summarizes the situation.




# 4 Discussion and open problems

As observed by Mundici, the question of whether Craig Interpolats have polynomial size is closely related to the existence of polynomia size circuits for NP. As we have mentioned

above, a superpolynomial lower bound on the interpolant size implies that NP does not have polynomial size circuits. A further relationship between the size of Boolean circuits for NP problems, and the size of interpolation circuits arises when we restrict ourselves to monotone circuits computing interpolants. Based on the results on exponential lower bounds for monotone circuits for the Clique function [17, 2] it can be observed that there are pairs of propositional formulas with monotone interpolants for which the smallest monotone circuit description must have exponential size. In order to see this, consider pairs of formulas $F_{n,k}(x,y)$ and $G_{n,k}(x,z)$ expressing that a graph $G_x$ (encoded as adjacency matrix in the $x$ variables) is a $k$ clique (respectively a $(k-1)$-coloring), and the variables $y$ and $z$ encode a solution to the problem. Clearly we have that $F_{n,k}(x,y) \rightarrow \neg G_{n,k}(x,z)$ is a tautology and therefore interpolants for $F$ and $\neg G$ depending only on the variables $x$ exist. The formulas have also monotone interpolants (for example the function $Clique_{n,k}$). By the lower bound results in [2], for $k = n^{2/3}$, any monotone circuit computing an interpolant for $F_{n,k} \rightarrow G_{n,k}$ has exponential size. The existence of formulas $F_{n,k}$ and $G_{n,k}$ follow already from Cook's Theorem and it is not hard to construct explicitely such formulas of size $O(n^4)$ for any value of $k$. (Similar families of formulas are given in [3]). From this fact and the monotone lower bound result from Alon and Boppana, it follows that any monotone circuit computing an interpolant for the formulas $F_{n,k}(x,y)$ and, $G_{n,k}(x,z)$ for $k = n^{2/3}$ must have size $2^{\Omega(m^\epsilon)}$ for $\epsilon < 1/12$, where $m$ is the sum of the sizes of $F_{n,k}$ and $G_{n,k}$.

Superpolynomial lower bounds on the size of circuits computing NP functions are only known for the monotone case. On the other hand, it is well known that the existence of polynomial size circuits for NP imply the collapse of the Polynomial Time Hierarchy [11]. It is an open question whether from the assumption H2 follows also a collapse of PH. In [8] the same open question for the dual hypothesis $Q'$ is stated. It is also open whether the simulataneous assumption of both hypothesis H2 and $Q'$ imply a collapse of PH.

# References


[1] E. Allender *Invertible Functions.* PhD Thesis, Georgia Tech, 1985.

[2] N. Alon, R. Boppana, The monotone circuit complexity of Boolean functions. *Combinatorica* **7** (1987) 1–22.

[3] M. Bonet, T. Pitassi and R. Raz, Lower bounds for cutting planes proofs with small coefficients. Proc. 27th ACM STOC (1995) pp. 575–584.

[4] S.R. Buss, *Lecture on Proof Theory.* Tech. Report SOCS-96.1, School of Computer Science, McGill University, 1996.

[5] S. Cook, The complexity of theorem-proving procedures. *Proc. 3rd ACM Symp. on the Theory of Computing* (1971) 36–50.

[6] W. Craig, Three uses of the Herbrand-Gentzen Theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, **44** (1957) 36–50.

[7] E. Dahlhaus, A. Israeli, J.A. Makowski, On the existence of polynomial time algorithms for interpolation problems in propositional logic. *Notre Dame Journal of Formal Logic* **29** (1988), 497–509.

[8] S. Fenner, L. Fortnow, A. Naik and J. Rogers, Inverting onto functions. *Proc. 11th IEEE Computational Complexity Conference* (1996), pp. 213–222.

[9] J. Grollman and A. Selman, Complexity measures for public-key cryptosystems. *SIAM Journal on Computing* **17** (1988) pp. 309–355.

[10] R. Imlagiazzo and M. Naor, Decision trees and downward closures. *Proc. 3rd Structure in Complexity Conference* (1988) pp. 29–38.

[11] R. Karp and R. Lipton Some connection between nonuniform and uniform complexity classes. *Proc. 12th ACM Symp. on the Theory of Computing* (1980) 302–309.

[12] J. Köbler, U. Schöning, S. Toda and J. Torán, Turing machines with few accepting computations and low sets for PP. *Journal of Comp. and System Sciences* **44** (1992) 272–286.

[13] J. Krajíček, Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. Preprint, (1995).

[14] D. Mundici, Tautologies with a unique Craig interplant, uniform vs. nonuniform complexity, *Annals of Pure and Applied Logic* **27** (1984), 265–273.

[15] D. Mundici, A lower bound for the complexity of Craig's interpolants in sentential logic. *Archiv. Math. Logik* **23** (1983) 27–36.

[16] P. Pudlák, Lower bounds for resolution and cutting plane proofs and monotone computations. Preprint, (1995).

[17] A. Razborov, Lower bounds for the monotone complexity of some Boolean functions. *Sov. Math. Dokl.* **31**, (1985) 354–357.

[18] L. Valiant, Relative complexity of checking and evaluating. *Information processing letters* **5** (1976) 20–23.