The computational power of bounded arithmetic from the predicative viewpoint

Samuel R. Buss*
Department of Mathematics
University of California, San Diego
La Jolla, CA 92093-0112
sbuss@math.ucsd.edu

December 4, 2006

Abstract

This paper considers theories of bounded arithmetic which are predicative in the sense of Nelson, that is, theories which are interpretable in Robinson's Q. We give a nearly exact characterization of functions which can be total in predicative bounded theories. As an upper bound, any such function has polynomial growth rate and its bit-graph is in nondeterministic exponential time and in co-nondeterministic exponential time. In fact, any function uniquely defined in a bounded theory of arithmetic lies in this class. Conversely, any function which is in this class (provably in $\mathrm{I}\Delta_0 + \exp$) can be uniquely defined and total in a (predicative) bounded theory of arithmetic.

1 Introduction

Theories of bounded arithmetic and their associated provable total functions have been extensively studied for over two decades. Bounded arithmetic arose originally from the definition of $\mathrm{I}\Delta_0$ by Parikh. A subsequent development by Nelson of his "predicative" theories gave an alternate route to bounded arithmetic. The present author's thesis [2] introduced the fragments S^i_2 and T^i_2 of bounded arithmetic (with Nelson's smash function present), and these have been extended over the years to a proliferation of theories of bounded arithmetic that have good characterizations of their

*Supported in part by NSF grant DMS-0400848.

provably total functions in terms of computational complexity. To mention only a few such characterizations, the Δ_0-definable predicates of $I\Delta_0$ are precisely the functions in the linear time hierarchy [8, 7], the provably total functions of S^1_2 are precisely the polynomial time computable functions [2], the provably total functions of T^1_2 are precisely the projections of polynomial local search (PLS) functions [4], Clote and Takeuti [5] gave formal theories that capture log space functions and alternating log time, and Arai [1] defined a system AID that better captures alternating log time. A large number of further bounded theories of arithmetic have been formulated by others, including Zambella, Cook and several students of Cook, see Cook-Nguyen [6] for a partial survey.

This paper returns to (one of) the historical motivations for bounded arithmetic by considering the computational complexity of functions which are definable in predicative theories of arithmetic. By "predicative" is meant in the sense of Nelson, namely, interpretable in Robinson's theory Q. Nelson introduced this notion of predicative because of his finitistic formalist philosophy of mathematics. Remarkably, another group of researchers were independently investigating mathematically equivalent notions of interpretability in Q, not for philosophical reasons, but for investigations into independence results for arithmetic and into computational complexity. This latter line of research included the foundational results of Solovay [11], Paris-Dimitracopoulos [9], Wilkie-Paris [13], Wilkie [12], and Pudlák [10].

The goal of the present paper is to study functions f which are defined in a predicative theory of arithmetic, and to characterize such functions in terms of computational complexity. We shall consider only bounded predicative theories and only theories that are interpretable with cut-interpretations. These restrictions are quite natural, since only cut-interpretations have been used for predicative theories to date and, at the present state of the art, we have essentially no ideas for what kinds of non-cut-interpretations could be defined, much less be useful.

The general outline of the paper is as follows. Although we presume the reader is familiar with both bounded arithmetic and predicative arithmetic, the next section give some technical preliminaries necessary for our exposition. Then section 3 proves an upper bound on the complexity of predicative functions. This upper bound is actually a upper bound on the complexity of any function that is uniquely determined by a definition over a bounded theory of arithmetic (not necessarily a predicative theory). Section 4 gives lower bounds. These lower bounds state that any function within a certain complexity class can be defined by a function symbol in some bounded predicative theory. For non-predicative theories, the upper and lower bounds match; namely, it is the class of functions whose bit-graph is in both nondeterministic exponential time and co-nondeterministic exponential time (see below for the exact definitions). For predicative theories, the lower bound further requires provability in the theory $I\Delta_0 + \exp$ of membership in this class.

2 Definitions

2.1 Cut-interpretability

When considering interpretability of a theory T in Q, we shall restrict our attention to theories T that (a) are Δ_0-axiomatizable, and (b) are interpreted in an inductive cut. The first condition, (a), means that T is axiomatized with a set of (universal closures of) bounded formulas over some language that extends the language of Q. The second condition, (b), means that the interpretation is relative to some *inductive* formula $\theta(x)$, for which Q proves the closure properties:

$$\theta(0) \ \wedge \ (\forall x)(\forall y)(\theta(x) \wedge y < x \rightarrow \theta(y))$$

and

$$(\forall x)(\theta(x) \rightarrow \theta(Sx)).$$

By compactness considerations, we may always assume without loss of generality that T has a finite language and is finitely axiomatized. Thus we do not need to worry about the distinction between local interpretability versus global interpretability.

We also may assume without loss of generality that any theory T contains S_2^1 (for instance) as a subtheory, and that any subset of the polynomial time functions and bounded axioms defining these functions and their properties are included in the theory T.

We frequently need the theory T to include induction for all bounded formulas in the language of T. In particular, if the theory T has non-logical language L, we write $I\Delta_0(L)$ to denote the set of induction axioms for all Δ_0-formulas over the language L. It is known (see Pudlák [10]) that if T is interpretable in Q, then so is $T + I\Delta_0(L)$. Thus, we may assume w.l.o.g. that any bounded theory T interpretable in Q includes the $I\Delta_0(L)$ axioms. For similar reasons, the theory T may be assumed to include the smash function, $\#$, and its defining axioms; in this case, the $I\Delta_0(L)$ axioms imply all of $S_2(L)$.

An important fact about interpretability in Q is that the growth rate of functions can be exactly characterized. We define $|x|$ as usual to equal the length of the binary representation of x. Then define

$$\Omega_0(x) \ = \ 2^{2|x|}$$

so that $\Omega_0(x) \approx x^2$, and further define

$$\Omega_{i+1}(x) \ = \ 2^{\Omega_i(|x|)}.$$

We have $\Omega_1(x) = 2^{2^{2||x||}} \approx 2^{|x|^2} = x\#_2 x$, and $\Omega_2(x) = 2^{2^{2^{2|||x|||}}} \approx x\#_3 x$, etc.

Solovay [11] proved that the functions $\Omega_i(x)$ are interpretable with inductive cuts in Q. Conversely, Wilkie [12] proved that any function f that is interpretable in an inductive cut in Q is bounded by some Ω_i, i.e., that for some i, $f(x) < \Omega_i(x)$ for all x. More generally, for a multivariable function symbol $f(x_1, \ldots, x_n)$, we say that f *is eventually dominated by* Ω_i provided that

$$(\exists \ell)(\forall x_1)(\forall x_2) \cdots (\forall x_n)(\ell \le x_1 + x_2 + \cdots + x_n \\ \rightarrow f(x_1, x_2, \ldots, x_n) < \Omega_i(x_1 + x_2 + \cdots + x_n)).$$

Then, if f is interpretable in an inductive cut in Q, f is eventually dominated by Ω_i for some i. Furthermore, the fact that f is dominated by Ω_i is provable in a suitable theory (which is interpreted in Q). This is expressed by the following theorem.

Theorem 1 *Suppose T is a finite Δ_0-axiomatized theory, interpretable in Q with an inductive cut. Then there is a finite extension T' of T which is also Δ_0-axiomatized and interpretable in Q with an inductive cut such that T' proves that every function symbol is eventually dominated by some Ω_i.*

Theorem 1 is a strengthening of the theorem of Wilkie [12]: its proof is beyond the scope of the present paper, but the crucial point of the proof is that Lemmas 8 and 9 and Corollary 10 of [12] can be formalized in $\mathrm{I}\Delta_0 + \exp$.

In view of the restriction on the growth rate of functions in inductive cuts in Q, we define the computational complexity class Ω_i-TIME to be the class

$$\Omega_i\text{-TIME} \ = \ \Omega_{i-1}(n^{O(1)})\text{-TIME}.$$

Here n indicates the length of an input, and it is easy to check that this means that the runtime of an Ω_i-TIME function on an input x of length $n = |x|$ is bounded by $|t(x)|$ for some term $t = \Omega_i^{(k)}(x)$ where $k \in \mathbb{N}$ and $\Omega_i^{(k)}$ indicates the k-fold composition of Ω_i.

Likewise we define analogues of exponential time by

$$\text{EXP}^i\text{-TIME} \;=\; 2^{\Omega_{i-1}(n^{O(1)})}\text{-TIME}.$$

The nondeterministic and co-nondeterministic time classes NEXP^i-TIME and coNEXP^i-TIME are defined similarly.

The bit-graph of a function f is the binary relation $BG_f(x, b, j)$ which is true exactly when the b-th bit of the binary representation of f is equal to j. Letting C be any of the time classes defined above, we define f to be in the complexity class C provided its bit-graph is in C. Note that, assuming f is a single-valued function (rather than a multifunction), f is in NEXP^i-TIME iff f is in coNEXP^i-TIME. In this case, we can say that f is in $\text{NEXP}^i\text{-TIME} \cap \text{coNEXP}^i\text{-TIME}$.

2.2 Definition of Δ_0-interpretable function

This section presents the crucial definition of what is meant by a function being interpretable in Q by a bounded theory. The intuition is that there should be a theory T with language $L \cup \{f\}$ which is Δ_0-axiomatized and which is interpretable in Q and which uniquely specifies the function f. The only tricky part of the definition is what it means to uniquely specify f: for instance, it would be cheating to have a function symbol $g \in L$ and an axiom $(\forall x)(f(x) = g(x))$, since this would merely beg the question of whether g is uniquely specified.

In order to formalize this properly, we let L^* be the language that obtained by making a "copy" of L: for each symbol $g \in L$, there is a symbol $g^* \in L^*$ (g may be a function symbol, a constant, or a predicate symbol). The function symbol f^* is defined similarly. The theory T^* is obtained from T be replacing all the symbols in the language $L \cup \{f\}$ with the corresponding symbol from $L^* \cup \{f^*\}$.

Definition A Δ_0*-interpretation in* Q of a function f consists of a theory T as above which is Δ_0-axiomatizable, is interpretable in Q with a cut interpretation, and for which

$$\mathrm{I}\Delta_0(f, f^*, L, L^*) + T + T^* \vdash (\forall x)(f(x) = f^*(x)). \qquad (1)$$

It is obvious that any Δ_0-interpretable function defines a function $f : \mathbb{N} \to \mathbb{N}$; that is to say, it defines an "actual" function on the integers. At the risk of confusing syntax and semantics, we define that any actual function defined by a symbol f of a theory T satisfying the conditions of the definition is Δ_0-interpretable in Q.

3 Upper bound

This section gives an upper bound on the computational complexity of functions which are Δ_0-interpretable in Q. The upper bound will not use the interpretability at all, but rather, will depend only on the fact that the function is uniquely defined in a bounded theory with the right growth rate functions.

Theorem 2 *Let T be Δ_0 axiomatized, with language $L \cup \{f\}$, f a unary function symbol. Suppose $T \supset \mathrm{I}\Delta_0(f, L)$ and that equation (1) holds so that T uniquely defines f. Further suppose that there is a $i > 0$ such that $\Omega_i \in L$ and the defining axioms of Ω_i are in T and such that for each function symbol g in the language of T, T proves that g is dominated by Ω_i. Then, f is in $\mathrm{NEXP}^i\text{-TIME} \cap \mathrm{coNEXP}^i\text{-TIME}$.*

Proof We may assume w.l.o.g. that T is finitely axiomatized and that T contains as many $\mathrm{I}\Delta_0$ axioms as is helpful. In fact, we may suppose T is axiomatized by a single $\forall\Delta_0$-sentence $(\forall x)\Theta(x)$. Also without loss of generality, we may assume that the axiom contains only terms of depth 1; that is, that no function symbols are nested. (This is easily accomplished at the expense of making the formula Θ more complicated with additional bounded quantifiers.) In addition, we may assume that every bounded quantifier in Θ is of the form $(\forall y \leq x)$ or $(\exists y \leq x)$; i.e., the universal quantified variable x effectively bounds all variables in the axiom. We let Θ^* denote the formula obtained from Θ by replacing each nonlogicial symbol g with g^*.

Let c be a new constant symbol. By (1), we have

$$\mathrm{I}\Delta_0(f, L, f^*, L^*) + f(c) \neq f^*(c) \vdash (\exists x)\left[\neg\Theta(x) \vee \neg\Theta^*(x)\right].$$

By Parikh's thoerem, there is a term $t(c)$ such that the quantifier $(\exists x)$ may be replaced by $(\exists x \leq t(c))$. It follows that there is a $k \in \mathbb{N}$ such that $t(x)$ is eventually dominated by $\Omega_i^{(k)}(x)$, provably in $\mathrm{I}\Delta_0(f, L, f^*, L^*)$, where the superscript "(k)" indicates k-fold iterated function composition. It follows that

$$\mathrm{I}\Delta_0(f, L, f^*, L^*) \ \vdash \ \text{"}c \text{ is sufficiently large"} \wedge f(c) \neq f^*(c) \\ \rightarrow (\exists x \leq \Omega_i^{(k)}(c))\, [\neg\Theta(x) \vee \neg\Theta^*(x)]\,.$$

The algorithm to compute the bit-graph of f can now be described: On input an integer c and integers b and j, the algorithm non-deterministically guesses and saves the following values:

1. For each n-ary function symbol $g(x_1, \ldots, x_n)$ of L and all values of $x_1, \ldots, x_n \leq \Omega_i^{(k)}(c)$, a value of $g(x_1, \ldots, x_n)$ which is $\leq \Omega_i(\Omega_i^{(k)}(c)) = \Omega_i^{(k+1)}(c)$,[1] and
2. For each n-ary predicate P and all values of $x_1, \ldots, x_n \leq \Omega_i^{(k+1)}(c)$, a truth value of $P(x_1, \ldots, x_n)$.

After non-deterministically guessing these values, the algorithm verifies that the axiom $\Theta(x)$ holds for all $x < \Omega_i(c)$. If they all hold, the algorithm accepts if the b-th bit of the guessed value of $f(c)$ is equal to j. Otherwise, the algorithm rejects.

It is now straightforward to check that the non-deterministic algorithm correctly recognizes the bit-graph of f. Furthermore, the run time of f is clearly bounded by $\Omega_i^{(s)}(c)$ for some $s \in \mathbb{N}$. Thus, the the algorithm is in NEXP^i-TIME. Since the function f is single-valued, the bit-graph is also in coNEXP^i-TIME. □

Corollary 3 *Suppose the $f : \mathbb{N} \rightarrow \mathbb{N}$ is Δ_0-interpretable in Q. Then f is in* NEXP^i-TIME ∩ coNEXP^i-TIME *for some $i \geq 0$.*

The corollary is an immediate consequence of Theorem 2 because of the results discussed in Section 2.

4 Lower bound

This section gives lower bounds for the definability of functions in bounded theories that match the upper bounds of the earlier section. Theorem 4 applies to arbitrary bounded theories and Theorem 5 applies to predicative bounded theories.

[1] This case covers constant symbols, since they may be viewed as 0-ary function symbols. In addition, the symbol f is of course one of the functions g, so the value of $f(c)$ is guessed as part of this process.

Theorem 4 *Suppose $f(x)$ is dominated by $\Omega_i(x)$ for some $i \geq 0$ and that the bit-graph of $f(x)$ is in* $\mathrm{NEXP}^i\text{-TIME} \cap \mathrm{coNEXP}^i\text{-TIME}$. *Then there is a bounded theory T in a language $L \cup \{f\}$ such that $T \vdash \mathrm{I}\Delta_0(f, L)$ and such that T proves f is total and uniquely defines $f(x)$.*

Theorem 5 *Let $f : \mathbb{N} \to \mathbb{N}$ be in* $\mathrm{NEXP}^i\text{-TIME} \cap \mathrm{coNEXP}^i\text{-TIME}$ *and be dominated by Ω_i for some $i \geq 0$. Suppose that $\mathrm{I}\Delta_0 + \exp$ can prove those facts; namely, there are predicates $A(x, b, j)$ and $B(x, b, j)$ such that $\mathrm{I}\Delta_0 + \exp$ can prove that,*

(a) *$A(x, b, j)$ and $B(x, b, j)$ are equivalent for all x, b, j,*

(b) *A is computable by a* NEXP^i*-TIME Turing machine, and*

(c) *B is computable by a* coNEXP^i*-TIME machine,*

and such that further the predicates A and B each define the bit-graph of f.

Then f is Δ_0-interpretable in Q.

We already gave a sketch of a proof of a weakened form of these two theorems in the appendix to [3]. That proof was based on the equivalence of alternating polynomial space and exponential time. Our proofs below, however, are based on directly representing non-deterministic exponential time computation with function values.

Proof (of Theorem 4). Consider two NEXP^i-TIME Turing machines, M and N, such that the language accepted by M is the complement of the language of N. Without loss of generality, the machines accept a single integer as input (in binary notation, say), use a single half-infinite work tape, and halt after exactly $\Omega_i(n)$ steps on an input z of length $n = |z|$.

We describe an execution of the machine M with trio of functions $T_M(z, i, j)$, $H_M(z, i)$, and $S_M(z, i)$. The intended meaning of $T_M(z, i, j) = c$ is that in the execution of M on input z, after i steps, the jth-tape square of M contains the symbol c. The intended meaning of $S_M(z, i) = q$ is that, M on input z after i steps, is in state q. The intended meaning of $H_M(z, i) = j$ is that M's tape head is positioned over tape square j after i steps. Now, since M is nondeterministic, there is more than one possible execution of M on input z, and this means the above "intended meanings" are under-specified. To clarify, the real intention is that if there is an execution of M on input z which leads to an accepting state, then we choose an arbitrary accepting computation and let $T_M(z, i, j)$, $H_M(z, i)$ and $S_M(z, i)$ be defined according to that accepting computation. On the other hand, if there is no accepting computation, we just choose any computation, and set the values of $T_M(z,i,j)$, $H_M(z,i)$ and $S_M(z,i)$ accordingly.

These conditions can be represented by Δ_0-axioms that express the following conditions stating that T_M, H_M, and S_M correctly define an execution of M:

(1) For all j, $T_M(z,0,j)$ has the correct value for initial state of M with z written on its input tape.

(2) $S_M(z,0)$ is equal to the initial state of M.

(3) $H_M(z,0)=0$, where w.l.o.g., M starts at tape square zero.

(4) For all $i \geq 0$, if $H_M(z,i) \neq j$ then $T_M(z,i,j) = T_M(z,i+1,j)$.

(5) For all $i \geq 0$, the transition rules for the machine M include a rule that allows M when reading symbol $T_M(z,i,H_M(z,i))$ in state $S_M(z,i)$ to write symbol $T_M(z,i+1,H_M(z,i))$, enter state $S_M(z,i+1)$, and either (i) move right one tape square, or (ii) move left one tape square. In case (i), it is required that $H_M(z,i+1) = H_M(z,i)+1$, and in case (ii), it is required that $H_M(z,i+1) = H_M(z,i)-1$.

Clearly, the conditions (1)-(5) are satisfied only if T_M, H_M, and S_M describe a correct execution of M (which may be either accepting or rejecting). Furthermore, there are natural exponential bounds on i and j given that M is an exponential time machine. It is clear that the conditions (1)-(5) can be expressed by $\forall\Delta_0$ statements, Γ_M.

Similar conditions Γ_N can be defined for the machine N using symbols T_N, H_N, and S_N.

The theory T defining f can now be defined. Let M be the NEXP^i-TIME machine that accepts an input $z = \langle x,b\rangle$ precisely when b-th bit of $f(x)$ is equal to 1. Likewise, let N be the NEXP^i machine that accepts the complement of the set accepted by M. The language of T includes the function symbol f, (a sufficiently large subset of) the language of PV, and the symbols T_M, H_M, S_M, T_N, H_N, and S_N. The axioms of T include induction for all Δ_0-formulas of T, plus the axioms Γ_M and Γ_N and axioms expressing the following two conditions:

(a) For each z, either $S_M(z,\Omega_i(z))$ is an accepting state, or $S_N(z,\Omega_i(z))$ is an accepting state, but not both, and

(b) The b-th bit of $f(x)$ is equal to 1 if and only $S_M(\langle x,b\rangle,\Omega_i(\langle x,b\rangle))$ is an accepting state.

By the fact that M accepts the complements of the set accepted by N, we see that (a) is a true condition, and the condition (b) is a correct definition of a function f. It is clear from the construction that the theory T correctly defines f. $\Box$

Proof (of Theorem 5, sketch). The idea of the proof is to formalize the proof of Theorem 4 in $\mathrm{I}\Delta_0 + \exp$. For this, recall from [13] that if $\mathrm{I}\Delta_0 + \exp$ can prove a Δ_0-formula $\theta(x)$, then there is some $k > 0$ such that $\mathrm{I}\Delta_0$ can prove "if the k-fold exponential of x exists, then $\theta(x)$ holds." Thus if the hypotheses of Theorem 5 hold, there is some $k > i$ such that $\mathrm{I}\Delta_0 +$ 'k-fold exponential of x exists" can prove that the predicates $A(x, b, j)$ and $B(x, b, j)$ accept the same set and are in NEXP^i-TIME and coNEXP^i-TIME (respectively).

It is well-known [11, 13] that Q can define inductive cuts $I(a)$ and $J(a)$ such that $J \subseteq I$ and $I \vDash \mathrm{I}\Delta_0$, and such that for all $x \in J$, the k-fold iterated exponential of x exists in I. Working in the cut J, Q can formalize the construction of the proof of Theorem 5 (with the aid of the k-fold exponentials of elements of J that exist in I). Then, introducing the function symbol f and the function symbols T_M, H_M, S_M, T_N, H_N and S_N for the NEXP^i-TIME Turing machines M and N which accept the set $\{z = \langle x, b\rangle : A(x, b, 1)\}$ and its complement (respectively), and restricting to the cut J, we have interpreted the definition of f into Q. $\Box$

The above theorems give an essentially exact characterization of the computational complexity of the functions which are Δ_0-interpretable in Q. It is clear that Theorems 2 and 4 give matching upper and lower bounds on the computational complexity of functions which are uniquely definable in Δ_0-theories. For Δ_0-interpretability in Q, Corollary 3 and Theorem 5 differ in the bounds on the function since the latter mentions provability in $I\Delta_0 + \exp$ whereas Corollary 3 does not mention provability explicitly. However, already the definition of Δ_0-interpretability, especially the provability in Q of the uniqueness condition (1), essentially implies the provability in $I\Delta_0 + \exp$ of the fact that the bit-graph of f is in NEXP^i-time (via the construction of the proof of Theorem 2).

We conclude with a few open problems. The first problem is the question of what multifunctions are interpretable in Q. A multifunction is a multiple-valued function (i.e., a relation); that is to say, there may be several values y such that $f(x) = y$ for a fixed x. The question is, if a multifunction is interpretable in Q, what is the minimum computational complexity of a total multifunction which satisfies the axioms of the theory?

A second line of research is to answer some questions left open from the work of Wilkie [12]. One such question is whether (now working over the base theory $I\Delta_0$ rather than Q) it is possible for a Σ_{n+1}-formula to define an inductive cut closed under Ω_n. [12] shows that it is not possible for a Σ_n or Π_n formula to define such a cut in $I\Delta_0$ and is possible for a Π_{n+1} to define one, but leaves the Σ_{n+1} case open. Likewise, it appears no one has studied the corresponding questions over the base theory Q. Another question is whether Theorem 1 can be proved by a direct proof-theoretic argument. The game theoretic argument in [12] is combined with a model theoretic proof. The model-theoretic part can be avoided, but it would nice to give a more direct proof-theoretic proof.

References

[1] T. Arai, *Frege system, ALOGTIME, and bounded arithmetic.* Manuscript, 1992.

[2] S. R. Buss, *Bounded Arithmetic*, Bibliopolis, 1986. Revision of 1985 Princeton University Ph.D. thesis.

[3] ——, *Nelson's Work on Logic and Foundations and Other Reflections on Foundations of Mathematics*, Princeton University Press, 2006, pp. 183–208. edited by W. Faris.

[4] S. R. Buss and J. Krajíček, *An application of Boolean complexity to separation problems in bounded arithmetic*, Proc. London Math. Society, 69 (1994), pp. 1–21.

[5] P. Clote and G. Takeuti, *Bounded arithmetics for NC, ALOGTIME, L and NL*, Annals of Pure and Applied Logic, 56 (1992), pp. 73–117.

[6] S. Cook and P. Nguyen, *Foundations of Proof Complexity: Bounded Arithmetic and Propositional Translations.* Book in preparation. Draft manuscipt available on web.

[7] R. J. Lipton, *Model theoretic aspects of computational complexity*, in Proceedings of the 19th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1978, pp. 193–200.

[8] R. J. Parikh, *Existence and feasibility in arithmetic*, Journal of Symbolic Logic, 36 (1971), pp. 494–508.

[9] J. B. Paris and C. Dimitracopoulos, *A note on the undefinability of cuts*, Journal of Symbolic Logic, 48 (1983), pp. 564–569.

[10] P. Pudlák, *Cuts, consistency statements and interpretation*, Journal of Symbolic Logic, 50 (1985), pp. 423–441.

[11] R. M. Solovay. Letter to P. Hájek, August 1976.

[12] A. J. Wilkie, *On sentences interpretable in systems of arithmetic*, in Logic Colloquium '84, J. B. Paris, A. J. Wilkie, and G. M. Wilmers, eds., North-Holland, 1986, pp. 329–342.

[13] A. J. Wilkie and J. B. Paris, *On the scheme of induction for bounded arithmetic formulas*, Annals of Pure and Applied Logic, 35 (1987), pp. 261–302.