Bounded arithmetic for deep inference and monotone systems

Anupam Das

April 30, 2015

This abstract is based on work that has been submitted to *MFCS '15* [12].

1 Introduction

Bounded arithmetic has been a fruitful way to relate complexity classes with logical theories and propositional proof systems. For example Paris and Wilkie showed that proofs of Π_1-sentences in the theory $I\Delta_0$, Robinson arithmetic with induction on Δ_0-formulae, translate to classes of polynomial-size Hilbert-Frege proofs of bounded depth and, conversely, that the soundness of bounded-depth Hilbert-Frege systems can be proved in $I\Delta_0$ [15].

In this work we introduce theories of bounded arithmetic for various monotone and deep inference proof systems. Along the way we are required to draw on a blend of tools at both the high and low level, including deep inference normalisation via graph rewriting, least fixed point operators, and intuitionistic bounded arithmetic.

Deep inference proof complexity has received much attention in recent years, and the complexity of the minimal system, KS, is considered as yet unresolved [5] [14] [10]. While an extension of it, KS^+, is known to quasipolynomially simulate Hilbert-Frege systems[1] there is neither such a simulation known for KS nor some nontrivial lower bound separating the two systems.

So-called 'analytic' deep inference systems ([4]) for propositional logic can be viewed as subclasses of *monotone proofs*, first introduced as sequent calculus proofs free of negation steps (*MLK*) and studied in e.g. [1] and [2], and we exploit this correspondence in the present work.

The ideas we present, in fact, were at the heart of the design of quasipolynomial-size proofs of the propositional pigeonhole principle in KS [11],[2] and so we believe that our theories might be useful for future research.

2 Preliminaries

Propositional logic consists of connectives $\bot, \vee, \wedge, \supset$. A *positive* or *monotone* formula is one that does not contain $\supset$. First-order logic extends propositional logic in the usual way.

2.1 Bounded arithmetic

We work in the language of Buss' theories S^i_2, T^i_2 [7], i.e. the language of arithmetic together with symbols $|\cdot|$ (length of binary representation), $\lfloor \frac{\cdot}{2} \rfloor$ (halve the argument and round down) and # (where $x\#y = 2^{|x|\cdot|y|}$).[3] We also assume the presence of further nonlogical predicate symbols R^k_i, which we call *nonarithmetic* symbols.

[1] Indeed, it is widely believed that in fact a polynomial simulation holds.

[2] Although in that work the arguments were presented internally to propositional logic.

[3] Due to the presence of #, we adopt quasipolynomial time as our model of feasible computation.

We work in a standard two-sided sequent calculus for first-order logic, denoted LK, extended by an appropriate set negation-free initial rules (i.e. axioms) which can essentially be found in [7].[4] Unless otherwise mentioned, we always assume *BASIC* to be contained in any defined theories.

For a class of formulae X the *induction* and *polynomial induction* rules for X-formulae ϕ are defined as follows:

$$X\text{-}IND\ \frac{\Gamma, \phi(a) \longrightarrow \phi(a+1), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(t), \Delta} \qquad X\text{-}PIND\ \frac{\Gamma, \phi(\lfloor \frac{a}{2} \rfloor) \longrightarrow \phi(a), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(t), \Delta}$$

The theories S_2^i and T_2^i are defined as Δ_i^b-*PIND* and Δ_i^b-*IND* respectively. The theories S_2 and T_2 are defined as $\bigcup_i S_2^i$ and $\bigcup_i T_2^i$ respectively. It is known that $S_2^i \subseteq T_2^i \subseteq S_2^{i+1}$, and so $S_2 = T_2$ [7].[5]

Paris and Wilkie in [15] defined a translation $\langle \cdot \rangle$ of closed Δ_0-formulae to (quasi)polynomial-size propositional formulae, where nonarithmetic symbols are associated with propositional variables. They essentially proved the following:[6]

Theorem 1 (Paris-Wilkie) *A T_2-proof of a Π_1-sentence $\forall x_1, \ldots, x_d.\phi(\vec{x})$ induces a quasipolynomial-time construction of bounded-depth sequent proofs of $\langle \phi(\vec{n}) \rangle_{\vec{n} \in \mathbb{N}^d}$.*

2.2 Monotone and normal proofs

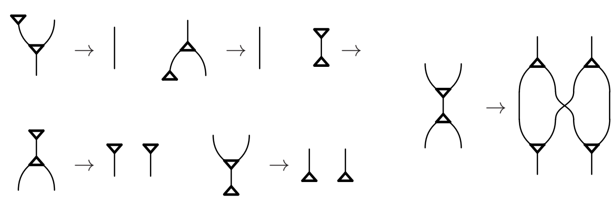
The setting we use is essentially due to Jeřábek in [14]: monotone proofs can be represented as term rewriting derivations in the following system,

$$\begin{array}{lll} w_1 : A \wedge B \to A & c_1 : A \to A \wedge A & \\ w_2 : A \to A \vee B & c_2 : A \vee A \to A & s : A \wedge (B \vee C) \to (A \wedge B) \vee C \end{array} \qquad (1)$$

modulo associativity and commutativity of $\wedge$ and $\vee$ (denoted AC). A *normal* monotone proof is one where all 1-steps occur before all 2-steps. We denote by Mon the rewriting system above and by Nor the set of all normal monotone proofs.[7]

The *(atomic) flow* of a proof is the graph obtained by tracing the paths of all atoms, designating nodes when atoms are created, destroyed or duplicated [13].[8]

Using flows, normalisation of monotone proofs can be conducted in an entirely 'syntax-free' way. We denote the following graph-rewriting system norm:



(2)

These steps can be *lifted* to proofs: for a Mon-proof π, manipulating its flow via norm results in the flow of a Mon-proof with same premiss and conclusion as π.[9]

norm is *terminating* and *confluent*, and its normal forms are precisely the flows of Nor-proofs [13] [10]. In particular we have the following result from [10]:

Theorem 2 norm *is weakly normalising in time polynomial in the number of paths in the input flow, and so induces a normalisation procedure for Mon of this complexity.*

[4]While the formulation referred to does include some occurrences of negation, these can be easily eliminated by simple arithmetic identities, e.g. as done in [6] and [9].

[5] Whether these inclusions are strict is an open problem.

[6]Paris and Wilkie in fact worked with slightly different theories, $I\Delta_0$ and $I\Delta_0 + \Omega_1$, but these only differ by the choice of base language.

[7]For the sake of reducing prerequisites, we deal with Mon and Nor in this abstract rather than explicitly defining the associated deep inference proof systems, KS^+ and KS respectively.

[8]These can be thought of as specialised versions of Buss' flow graphs [8].

[9]The proof of this relies somewhat crucially on an extension of Mon by the *medial* rule [3], as well as basic manipulations with constants.

3 Main results

Let Δ_0^+ be the class of *positive* Δ_0-formulae. We define the theories MS_2 and MT_2 as Δ_0^+-*PIND* and Δ_0^+-*IND* respectively.[10]

We define positive *least fixed point* operators LFP^f, in such a way that their closure functions[11] are bounded by $f(\vec{x})$.[12]

3.1 From arithmetic to propositional proofs

Lemma 3 *The translation $\langle \cdot \rangle$ for closed Δ_0-formulae can be extended to LFP^f, inducing monotone circuits with quasipolynomial-size $\bigwedge, \bigvee$ gates and depth $O(f)$.*

From here the translation from $MT_2 + LFP^{\text{polylog}}$ to Mon is a simple observation:

Proposition 4 (Extended Paris-Wilkie) *A $MT_2 + LFP^{\text{polylog}}$ proof of a positive sequent $\phi(\vec{x}) \longrightarrow \psi(\vec{x})$ induces quasipolynomial-size monotone proofs $\langle \phi(\vec{n}) \rangle \xrightarrow[Mon]{*} \langle \psi(\vec{n}) \rangle$.*

The case for MS_2 and Nor, however, relies on analysing the atomic flows of $\langle \cdot \rangle$.

Lemma 5 *The $\langle \cdot \rangle$-translation of $MS_2 + LFP^{\text{polylog}}$ proofs have polylogarithmic length flows.*[13]

We can now apply our normalisation result, Thm. 2, to obtain the following:

Theorem 6 *A $MS_2 + LFP^{\text{polylog}}$ proof of a positive sequent $\phi(\vec{x}) \longrightarrow \psi(\vec{x})$ induces quasipolynomial-size normal proofs $\langle \phi(\vec{n}) \rangle \xrightarrow[Nor]{*} \langle \psi(\vec{n}) \rangle$.*

We also point out that using LFP^{poly} instead would allow us access to monotone circuits of polynomial depth, and so corresponds to monotone proofs with extension.

3.2 From propositional proofs to arithmetic

Unfortunately, the monotone setting in arithmetic does not allow us to readily conduct metamathematical reasoning, and so it seems difficult (perhaps impossible) to prove soundness results within MS_2 and MT_2 ($+LFP^{\text{polylog}}$).

Therefore we introduce an intuitionistic hierarchy of theories in which to conduct metamathematical reasoning. The idea here is to view intuitionistic logic as a "logic of proofs". In this way we can extend the PW-translation to deal with implication without breaking monotonicity.

Proof systems for *intuitionistic* theories coincide with usual ones, except with the proviso that at most one formula occurs on the right of a sequent.[14]

Define $\Delta_0^{I_j}$ as the class of Δ_0 formulae whose left implication depth[15] is at most j. We define the systems $I_j S_2$ and $I_j T_2$ as $\Delta_0^{I_j}$-*PIND* and $\Delta_0^{I_j}$-*IND* respectively.

It is not difficult to see, at the level of theories, that $I_0 S_2 = MS_2$ and $I_0 T_2 = MT_2$. More interestingly, $I_1 S_2$ also translates to small Mon-proofs:

Theorem 7 *An $I_1 S_2 + LFP^{\text{polylog}}$ proof of a positive sequent $\phi(\vec{x}) \longrightarrow \psi(\vec{x})$ induces quasipolynomial-size monotone proofs $\langle \phi(\vec{n}) \rangle \xrightarrow[Mon]{*} \langle \psi(\vec{n}) \rangle$.*

[10] Notice that, in the absence of negation for nonarithmetic symbols, it is no longer clear that $MS_2 = MT_2$.

[11] This is the number of times the inductive definition needs to be applied to reach a fixed point.

[12] Notice that these can even be designed by simply internalising a 'clock' in the inductive definitions.

[13] This is due to the fact that each *PIND*-step only multiplies the length of a flow by a logarithm.

[14] We point out that previous work on intuitionistic bounded arithmetic has included only positive induction [6] [9], in order to conduct realisability arguments. These, in fact, simulate the full power of non-positive induction in the absence of nonarithmetic symbols.

[15] This is the largest number of times a path chooses the left branch of a $\supset$ symbol in the formula tree.

The idea here is that a proof π of $I_1 S_2$ of a $\Delta_0^{I_j}$ sequent,

$$A_1(\vec{x}) \supset B_1(\vec{x}), \ldots, A_k(\vec{x}) \supset B_k(\vec{x}) \longrightarrow A(\vec{x}) \supset B(\vec{x})$$

is translated to a quasipolynomial-time transformation,

$$\langle \pi \rangle \quad : \quad \begin{pmatrix} A_1(\vec{n}) & & A_k(\vec{n}) \\ \Phi_1 \| & , \ldots , & \Phi_k \| \\ B_1(\vec{n}) & & B_k(\vec{n}) \end{pmatrix} \mapsto \begin{matrix} A(\vec{n}) \\ \Phi \| \\ B(\vec{n}) \end{matrix}$$

for arbitrary monotone derivations Φ_i of the given format.[16] Complexity is generated by left-contraction steps, which correspond to dag-like behaviour in an *MLK*-proof, e.g.

$$\frac{\begin{matrix} \pi \\ \vdots \\ A \supset B, A \supset B \longrightarrow C \supset D \end{matrix}}{A \supset B \longrightarrow C \supset D} \qquad \rightsquigarrow \qquad \langle \pi' \rangle \text{ where } \langle \pi' \rangle(\Phi) := \langle \pi \rangle(\Phi, \Phi)$$

However *PIND*, again, ensures the graphs of these proofs have only polylogarithmic length, and so can be transformed in quasipolynomial-time to tree-like *MLK*, i.e. Mon proofs.

It turns out that this theory contains just enough negation to obtain a converse result:

Theorem 8 (Reflection) $I_1 S_2 + LFP^{\text{polylog}}$ *proves the soundness of* Mon.

As one would expect, we also have that $I_1 T_2$ corresponds to dag-like *MLK*, in the same way.

4 Further work and conclusions

We gave uniform versions of monotone and analytic deep inference proof systems, in the setting of bounded arithmetic. This constituted an application of least fixed points, deep inference proof normalisation and intuitionistic bounded arithmetic to propositional proof complexity. In the case of monotone proofs we were able to also prove a converse result.

We believe that the set of provably total functions in all our theories is the *polynomial hierarchy*, due to prior results on 'intuitionistic' bounded arithmetic [6] [9] where only positive induction is used. However this is perhaps not as meaningful in the presence of nonarithmetic symbols in our settings.

We do not yet have a full correspondence for Nor, and this reflects the difficulty in deep inference proof complexity of conducting any metamathematical reasoning at all in KS. One approach might be to incorporate structural restrictions from *linear logic*, e.g. that induction formulae must be exponential-free. While this might not be desirable from the point of view of reasoning, it might allow us to conduct one-off proofs of soundness of various systems.

A more general question is that of the relative strength of our theories. In fact, we suspect that the nonuniform monotone simulation of Hilbert-Frege proofs given in [2] can be made uniform in $I_1 S_2 + LFP^{\text{polylog}}$, so that the hierarchy collapses to this level. Thence, two natural questions arise:

1. Are the inclusions $MS_2 + LFP^{\text{polylog}} \subseteq MT_2 + LFP^{\text{polylog}} \subseteq I_1 S_2 + LFP^{\text{polylog}}$ strict?
2. Can a simpler purely logical proof be given of the collapse of the intuitionistic hierarchy, perhaps to some $I_k S_2$ for some large, but finite k?

[16] Positive boolean combinations of implications are interpreted in the natural way.

A positive answer to the second question would give a simple proof of the quasipolynomial-simulation of Hilbert-Frege over monotone sequents by some 'super-dag-like' monotone system, or equivalently propositional intuitionistic Hilbert-Frege systems of bounded implication depth.

This work further brings research in deep inference in line with the standards of mainstream proof complexity. By studying restrictions of monotone systems we also somewhat contribute to 'bridging the gap' between weak systems and Hilbert-Frege systems, for which no nontrivial lower bounds are known.

References

[1] A. Atserias, N. Galesi, and R. Gavalda. Monotone proofs of the pigeon hole principle. 2000.

[2] A. Atserias, N. Galesi, and P. Pudlák. Monotone simulations of non-monotone proofs. *Journal of Computer and System Sciences*, 65(4):626–638, 2002.

[3] K. Brünnler and A. F. Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001. http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf.

[4] P. Bruscoli and A. Guglielmi. On analyticity in deep inference. http://cs.bath.ac.uk/ag/p/ADI.pdf, 2009.

[5] P. Bruscoli and A. Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009. Article 14. http://cs.bath.ac.uk/ag/p/PrComplDI.pdf.

[6] S. R. Buss. The polynomial hierarchy and intuitionistic bounded arithmetic. 1985.

[7] S. R. Buss. *Bounded arithmetic*, volume 3. Bibliopolis Naples, 1986.

[8] S. R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.

[9] S. Cook and A. Urquhart. Functional interpretations of feasibly constructive arithmetic. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 107–112. ACM, 1989.

[10] A. Das. Complexity of deep inference via atomic flows. In S. B. Cooper, A. Dawar, and B. Löwe, editors, *Computability in Europe*, volume 7318 of *Lecture Notes in Computer Science*, pages 139–150. Springer-Verlag, 2012. http://www.anupamdas.com/items/RelComp/RelComp.pdf.

[11] A. Das. On the pigeonhole and related principles in deep inference and monotone systems. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, pages 36:1–36:10, New York, NY, USA, 2014. ACM.

[12] A. Das. Bounded arithmetic for deep inference and monotone systems, 2015. http://www.anupamdas.com/bdd-arith-di-full.pdf.

[13] A. Guglielmi and T. Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36, 2008. http://www.lmcs-online.org/ojs/viewarticle.php?id=341.

[14] E. Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009. http://www.math.cas.cz/~jerabek/papers/cos.pdf.

[15] J. Paris and A. Wilkie. δ_0 sets and induction. *Open Days in Model Theory and Set Theory, W. Guzicki, W. Marek, A. Pelc, and C. Rauszer, eds*, pages 237–248, 1981.